퍼지 전처리기를 가진 신경회로망 모델의 개발

Developement of a Neural Network with Fuzzy Preprosessor

조 성 원*, 황 인 호**
Seongwon Cho*, In Ho Hwang**

* 본 논문은 1993년도 학술진흥재단의 공모과제 연구비에 의하여 연구되었음

요 약

본 논문에서는 신경회로망의 분류 정확도를 향상시키고 퍼지 데이타도 분류 할 수 있도록 하기 위하여 퍼지 전처리기를 갖는 신경회로망을 제안한다. 제안된 방법을 사용하면 수치값으로 주어지는 입력뿐 아니라 언어의 형태로 주어지는 부정확한 입력들도 분류할 수 있다. 퍼지 신호 변환 방법은 전처리로 수행되어 진다. 이 과정을 통하여 언어 형태의 부정확한 입력과 수치형태의 입력을 다차원 수치 값으로 변환한다. 변환된 입력은 후처리 모듈인 신경회로망의 입력으로 사용된다. 제안된 방법을 사용하여 실험한 결과 퍼지 입력 신호표현 방법이 이진 입력표현이나 십진 입력표현 방법에 비해 우수한 분류 성능을 나타냄을 확인 할 수 있었다.

ABSTRACT

In this paper, we propose a neural network with fuzzy preprocessor not only for improving the classification accuracy but also for being able to classify objects whose attribute values do not have clear boundaries. The fuzzy input signal representation scheme is included as a preprocessing module. It transforms imprecise input in linguistic form and precisely stated numerical input into multidimensional numerical values. The transformed input is processed in the postprocessing module. The experimental results indicate the superiority of fuzzy input signal representation scheme in comparison to binary input signal representation scheme and decimal input signal representation scheme.

I. 서 론

본 논문에서는, 신경회로망의 분류 정확도를 향상시키고 퍼지 데이타도 분류할 수 있도록 하기위하여 입력 신호를 퍼지 변환하는 전처리 모듈을 갖는 신경회로망을 제안한다.

퍼지 신호 변환을 통하여 퍼지집합으로 주어지는 입력이나 수치값으로 주어지는 입력을 다차원 데이타로 변환하고, 이 변환된 입력은 신경회로망의 학습 데이타로 사용된다.

* 홍익대학교 전자·전기·제어공학과 조교수
** 홍익대학교 대학원생

본 논문은 다섯 장으로 이루어져있다. 2장은 학습모드에서의 퍼지 신호 표현에관해 설명하고 3장은 테스팅 모드에서의 입력 신호 표현에 대해서 설명한다. 4장에서는 퍼지 전처리기를 갖는 신경회로망과 원래의 신경회로망과의 분류성능을 비교하고, 5장은 결론부이다.

Ⅱ. 신경회로망의 학습모드에서 퍼지 신호 표현

입력정보를 어떻게 수치값으로 표현할 것인지가 신경회로망에서는 중요한 문제이다. 입력 신호를 십진수나 이진수로 표현하는 것이 반드시 최적인것은 아니다. 본 논문에서는 학습 데이타로부터 생성된 퍼지집합의 멤버십함수를 이용한 새로운 표현법을 제안한다. 신경회로망 학습의 전처리 단계로써 주어진 입력신호를 퍼지 입력신호로 변환하게 되는데 그 절차는 다음과 같다.

단계1 : 학습데이타로부터 퍼지집합들의 멤버십함수를 유도한다. 이 단계를 수행하는 방법은 2.1에서 설명한다.
단계2 : 각 퍼지집합을 두개의 새로운 퍼지집합으로 나눈다. 자세한 것은 2.2에서 설명한다.
단계3 : 퍼지집합들의 클래스 분리정도에 근거해 K개의 퍼지집합을 선택한다. 이 단계를 수행하는 방법은 2.3에서 설명한다.
단계4 : 단계 3에서 선택된 퍼지지합들과 유사정도 계산법을 사용하여 학습벡터들을 다차원 벡터로 변환한다. 자세한 것은 2.4에서 설명한다.

2.1 멤버십함수의 자동적인 생성

비퍼지 시스템과는 다르게, 퍼지시스템에서는 사용되는 퍼지집합들의 멤버십함수를 정의해 주어야 한다. 일반적으로 퍼지시스템에서는 전문가의 지식이나 경험을 이용해 퍼지집합의 멤버십 함수를 주관적으로 정의한다. 이것을 할 수 없을 때에는 실험 데이타로부터 직접 멤버십함수를 유도하게 되는데, 지금까지 여러가지 방법들이 제안되었다[1][3][4][6]. 본 논문에서는 객관적인 실험 데이타의 분포에 근거하여 퍼지집합의 멤버십함수를 자동으로 생성하는 새로운 방법을 제안한다. 실험 데이타는 샘플의 집합으로 이루어져 있고, 각 샘플은 관찰되는 입출력 관계를 표현하며 어떤 대상의 몇가지 속성값들과 그 대상이 속하는 클래스로 이루어져 있다. 각 속성에 대하여 제한된 수의 퍼지집합과 그것들의 멤버십함수들이 유도된다. 샘플들로부터 퍼지집합의 멤버십함수를 유도하기 위하여 각 속성에대해 수행해야할 절차는 다음과같다.

초기화 : i = 1

단계1 : 클래스 i에 속하는 샘플들에 대하여, 각 속성의 히스토그램을 만든다. 변수 n은 가능한 클래스의 갯수이다. 히스토그램의 X축과 Y축은 각각 속성값의 전체 영역과 각 구간에 속하는 샘플들의 갯수이다.
단계2 : j번째 구간, x_j, 퍼지집합 A_i의 특성들을 만족하는 정도를 계산한다.

$$m_{Ai}(x_j) = \frac{1}{N_{ci}} \sum_{k=1}^{I} \min(N_j, N_k) \tag{1}$$

여기서 $m_{Ai}(x_j)$는 A_i에대한 x_j의 멤버십값을 나타내며, N_j와 N_k는 각각 j와 k번째 구간에 속하는 샘플들의 갯수이고, I는 구간의 갯수이며, N_{ci}는 클래스 i에 속하는 전체 샘플의 갯수이다.

단계3 : i = n이면 단계 4로 가고, 그렇지 않으면 i를 하나 증가 시키고 단계 1로 간다.
단계4 : 단계 2에서 계산된 멤버십함수들을 사용해 n개의 클래스에 대응되는 퍼지집합들 사이의 유사정도를 계산한다. 유사정도의 계산에는 Euclidian 거리법이나 Hamming 거리법을 사용한다.
단계5 : 만약 유사정도가 임계값보다 작으면, 유사한 퍼지집합들을 합쳐 하나의 새로운 퍼지집합을 만든다. 새

로운 퍼지집합의 멤버십함수는 식(1)에의해 다시 계산한다.

한 속성에 대하여 가능한 퍼지집합의 최대 갯수는 n이다. 단계 1에서 단계 5의 과정이 각 속성에 대하여 반복적으로 수행된다. 그러므로, 만약 m개의 속성이 있다면 퍼지집합의 전체 최대 갯수는 $m \times n$이다.

2.2 퍼지집합의 변형

일반적으로 학습 데이타로부터 생성된 퍼지집합의 멤버십함수는 비감소 함수(nondecreasing function)가 아니다. 그림 1 에서 예를 보여준다.

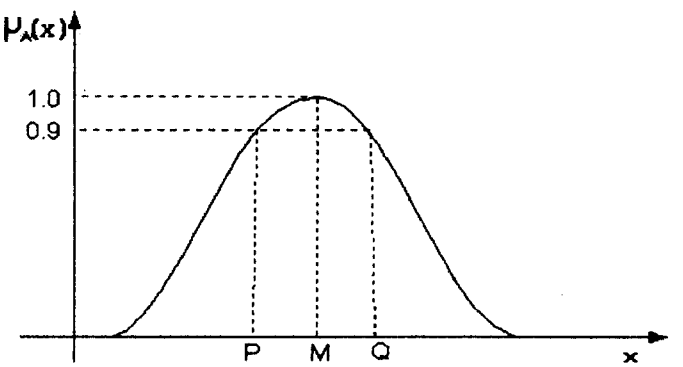

그림 1 원래의 퍼지집합 A

P와 Q점은 X축에서는 서로 다른 값이지만 퍼지집합 A에서의 멤버십값은 서로 같다. 그러므로, 만약 이러한 멤버십함수를 신경회로망의 퍼지 입력신호에 직접 사용한다면 P점에 위치한 어떤 대상물이 Q점에 위치한 다른 대상물과 동일한 것으로 간주될 가능성이 크다. 이 문제를 해결하기 위해서는 멤버십함수를 변형해 주어야 하는데, 다음에서 이를위한 하나의 방법을 기술한다.

단계1 : 퍼지집합, A,의 멤버십값들중 가장 큰 값을 갖는 X축상의 값, M,을 찾는다.

$$m_A(M) = 1 \tag{2}$$

단계2 : 원래의 퍼지집합을 두개로 나눈다.

1) X축상의 값, x, 이 M보다 크거나 같으면, 첫번째 새로운 퍼지집합, A_1,의 멤버십값들은 1이고 두번째 퍼지집합, A_2,의 멤버십값들은 원래 퍼지집합의 멤버십값들과 같다.

$$m_{A1}(x) = 1 \quad \text{and} \quad m_{A2} = m_A(x) \quad \text{for} \quad x \geq M \tag{3}$$

2) 만약 X축상의, x, 이 M보다 작으면, 첫번째 새로운 퍼지집합, A_1,의 멤버십값은 원래 퍼지집합의 멤버십값과 같고 두번째 퍼지집합, A_2,의 멤버십값은 1이다.

$$m_{A1}(x) = m_A \quad \text{and} \quad m_{A2} = 1 \quad \text{for} \quad x < M \tag{4}$$

그림 2는 그림 1의 원래의 퍼지집합으로부터 위의 방법에 따라 변형된 두개의 새로운 퍼지집합을 보여주고 있다.

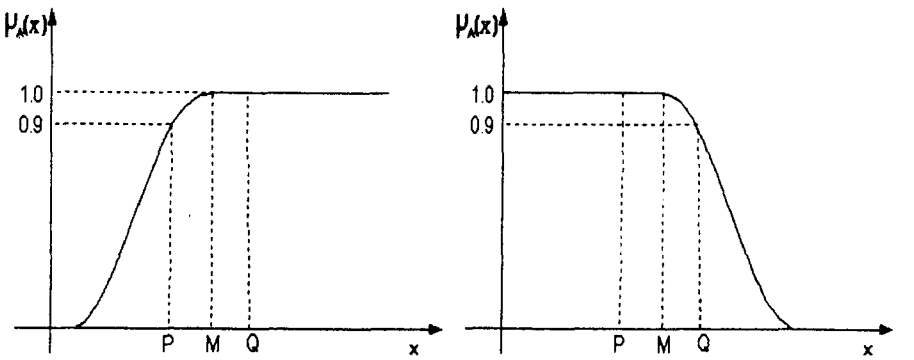

그림 2 A로부터 나누어진 퍼지집합 A_1과 A_2

2.3 퍼지집합의 선택

신경회로망에서 퍼지 입력신호 표현법을 사용할때, 입력층의 노드의 갯수는 퍼지집합의 전체 갯수와 같다. 모든 각각의 퍼지집합이 두개의 새로운 퍼지집합으로 나누어진 후에는, 전체 퍼지집합의 최대 갯수는 $2\times m\times n$이다. 여기서 m은 속성의 갯수(원래 입력 벡터의 차원)이고 n은 클래스의 갯수이다. 그러나, 실제 하드웨어적인 구현에 있어서는, 입력층에 있는 노드의 갯수를 줄일 필요가있다.

본 절에서는, 클래스 분리정도에 근거해서 퍼지집합들을 선택하는 방법론을 기술한다. 클래스 분리정도가 작은 퍼지집합들은 사용하지 않은 때 보다 오히려 학습능력을 저하 시킬 우려가 있으므로 제외시키고 분리정도가 큰 퍼지집합들만을 선택하여 입력신호 변환에 사용한다. 퍼지집합의 클래스 분리정도는 within-class scatter와 between-class scatter를 사용하여 계산하는데, 그 절차는 다음과 같다.

단계1 : 각 클래스 i와 클래스 i에 속하는 입력 벡터들의 j번째 속성값들의 멤버십값들을 갖는 퍼지집합 A_{jk}에 대하여 샘플들의 평균값, m_i,을 계산한다.

$$m_i=\frac{1}{N_i}\sum_{m=1}^{N_i} m_{Ajk}(x_{mj}) \tag{5}$$

단계2 : 각 클래스 i와 클래스 i에 속하는 입력 벡터들의 j번째 속성값의 멤버십을 갖는 퍼지집합 A_{jk}에대하여 샘플의 분산, s_i^2,을 계산한다.

$$s_i^2=\frac{1}{N_i}\sum_{n=1}^{N_i}[\,m_{Ajk}(x_{nj})-m_i\,]^2 \tag{6}$$

단계3 : 퍼지집합 A_{jk}에대한 within-class scatter, S_w,를 계산한다. Within-class scatter는 클래스내에 있는 샘플들의 평균분산을 나타내며 다음과 같이 표현된다.

$$S_w=\sum_{i=1}^{n} P_i s_i^2 \tag{7}$$

여기서 n은 클래스의 전체 갯수이며, P_i는 전체 데이타 중에서 각 속성의 데이타들이 차지하는 비율을 나타내는

확률이다.

$$P_i = \frac{N_i}{\sum_{k=1}^{n} N_k} \tag{8}$$

단계 4 : 퍼지집합 A_{jk}에 대하여 m_i의 평균값 m_o를 계산한다.

$$m_o = \sum_{i=1}^{n} P_i m_i \tag{9}$$

단계 5 : Between-class scatter, S_b,를 계산한다.

$$S_b = \sum_{i=1}^{n} P_i [m_i - m_o]^2 \tag{10}$$

단계 6 : 마지막으로, within-class scatter와 between-class scatter를 이용하여 클래스 분리 정도, S,를 계산한다.

$$S = \frac{S_b}{S_w} \tag{11}$$

단계 7 : 모든 퍼지집합에 대하여 단계 1에서 단계 6까지를 반복한다.

2.4 학습입력의 멤버십함수 벡터로의 변환

일단 입력 표현에 사용될 퍼지집합들을 선택한 후에는, 원래의 학습입력들이 수치값으로 주어진 경우에는 다음 식을 이용하여 멤버십함수 벡터로 변환된다[6].

$$M_{jk} = \mu_{A_{jk}}(x_{mj}) \tag{12}$$

여기서 x_{mj}는 입력벡터 x_m의 j번째 속성값을 의미하며, A_{jk}는 j번째 속성의 k번째 퍼지집합이고, $\mu_{A_{jk}}(x_{mj})$는 x_{mj}의 A_{jk}에서의 멤버십값이다. M_{jk}는 j번째 속성의 k번째 퍼지집합으로부터 얻어지는 멤버십값이다. j번째 속성값에 대하여, 선택된 퍼지집합의 수가 K_j라면, 원래 학습벡터의 대응되는 속성 값은 K_j차원의 멤버십값으로 변환된다.

만약 원래의 학습 벡터들이 수치값이 아닌 애매모호함을 나타내는 퍼지집합으로 주어지면, 멤버십 벡터들은 다음과 같은 식들을 사용해 얻어진다. 이를 위해 먼저 두개의 파라미터들이 정의된다[8].

$$CERT_+ = \max\{\min(\mu_A, \mu_{A*})\}$$
$$CERT_- = \max\{\min(1-\mu_A, \mu_{A*})\}$$

여기서 μ_A는 선택된 퍼지집합의 멤버십함수이고 μ_{A*}는 학습 입력을 나타내는 퍼지집합의 멤버십함수이다.

$CERT_+(x_{mj}, A_{jk}) \geq 0.5$ 이고 $CERT_-(x_{mj}, A_{jk}) \leq 0.5$ 인 경우에는,
$$M_{jk} = CERT_+(x_{mj}, A_{jk}) \tag{13}$$
$CERT_+(x_{mj}, A_{jk}) \geq 0.5$ 이고 $CERT_-(x_{mj}, A_{jk}) \geq 0.5$ 인 경우에는,
$$M_{jk} = \{CERT_+(x_{mj}, A_{jk}) - CERT_-(x_{mj}, A_{jk})\} + 0.5 \tag{14}$$
$CERT_+(x_{mj}, A_{jk}) \leq 0.5$ 이고 $CERT_-(x_{mj}, A_{jk}) \geq 0.5$ 인 경우에는,
$$M_{jk} = 1 - CERT_-(x_{mj}, A_{jk}) \tag{15}$$

멤버십벡터는 입력벡터의 모든 M_{jk}값들을 일정한 순서로 나열함으로써 얻어진다. 선택된 퍼지집합의 멤버십 함수들과 M_{jk}값들의 멤버십벡터에서의 순서는 테스팅벡터의 퍼지 입력신호 표현에 사용할 수 있도록 저장해둔다. 멤버십벡터의 크기는 2.3절의 절차를 사용해 선택한 전체 퍼지집합의 갯수와 같다.

$$\text{멤버십벡터의 차원} = \sum_{j=1}^{m} K_j \tag{16}$$

여기서 K_j는 j번째 속성에대하여 선택된 퍼지집합의 갯수이고 m은 속성의 전체 갯수이다.

Ⅲ. 테스팅 모드에서의 퍼지 신호 표현

테스팅 모드에서의 퍼지 신호 표현은 간단하다. 학습모드에서 필요했던, 멤버십함수를 유도하고, 퍼지집합을 변형하고, 퍼지집합을 선택하는 단계들이 테스팅 모드에서는 필요없다. 테스팅 데이타들은 학습 데이타와 같은 방법에의해 멤버십벡터로 변환된다.

정확한 속성값의 경우에는, 식(12)에의해 멤버십벡터로 변환된다. 부정확한 속성값인 경우에는 식(13)~(15)를 사용하여 멤버십값들을 계산한다.

Ⅳ. 실험 결과

퍼지 입력신호 표현을 십진이나 이진 데이타 입력 표현과 비교하기위해 가장 널리 사용되는 신경회로망인 Backpropagaton Network(BPN)을 사용한 실험들이 수행되었다. 실험에 사용된 데이타는 4개의 클래스를 갖는 데이타와 8클래스를 갖는 두가지의 원거리 측정 데이타, FLC1(Flight Line C1)가 사용되었다. 퍼지 전처리기의 분류성능을 고찰하기 위해, 퍼지 신호 변환을 갖는 BPN과 원래의 BPN의 분류성능들을 비교하였다.

4.1 데이타 셋(Data Set)

실험 데이타로는 미국 중부 지방의 평원 지역을 항공 촬영한 원격 탐사(remote sensing) 데이타를 사용하였다[9]. 이 데이타는 8-밴드(band)를 가진다. 즉 입력벡터들은 8차원의 0에서 255사이의 정수값을 갖는다. 또한 〈표 1〉과 〈표 2〉와 같이 4개의 농작물과 8개의 농작물을 나타내는 두종류의 데이타로 이루어져 있다.

각 클래스당 학습 데이타는 200개의 샘플로 구성되며 테스팅 데이타는 375개로 구성되어 있다.

〈표 1〉4클래스 실험 데이타의 구성

클래스	영역 설명
1	풀밭
2	옥수수
3	귀리
4	붉은 클로버

〈표 2〉8클래스 실험 데이타의 구성

클래스	영역 설명
1	풀밭
2	옥수수
3	귀리
4	붉은 클로버
5	콩
6	밀
7	공터
8	호밀

4.2 시뮬레이션

퍼지집합에 대한 멤버십함수가 유도된후, 유사정도 계산에는 Hamming 거리법이 사용되었다. 4 클래스 데이

타를 사용한 실험에서는 퍼지집합들이 유사한가 아닌가를 판단하기위한 임계값으로는 Hamming 거리 4.0이 사용되었다. 각 퍼지집합을 2개로 나눈후, 퍼지집합의 전체 갯수는 44개였다. 이중에서 클래스 분리정도가 큰 26개의 퍼지집합만을 선택했다. 그러므로 퍼지 입력표현을 갖는 Backpropagation Network(BPN)인 경우, 입력노드의 수는 1개의 bias를 포함하여 27개이고, 은닉층의 노드수도 1개의 bias를 포함하여 27개로 하였고 출력노드는 4개로 잡았다. 이진 데이타를 사용한 실험에서는 각 속성을 8 비트로 이진화하여 사용하였다. 이 경우 입력노드의 갯수는 bias를 포함하여 65개이고, 은닉층의 노드수도 1개의 bias를 포함하여 65개로 하였고 출력노드는 4개로 잡았다. 십진 데이타를 사용한 실험에서는 모든 데이타를 각 속성에 대하여 0에서 1의 값으로 정규화(normalization)하여 사용하였다. 입력노드는 bias를 포함하여 9개이고, 은닉층의 노드수도 1개의 bias를 포함하여 9개로 하였고 출력노드는 4개로 잡았다. 학습률은 모든 실험에서 똑같이 0.01로 잡았고, 가중치(weight)는 −0.5에서 0.5사이의 값들로 초기화하여 실험을 하였다.

〈표 3〉 이진 데이타에 대한 BPN의 분류성능(4 클래스)

Classification Accuracy		
Iteration	Training data	Testing Data
200	91.88%	84.60%
300	94.00%	87.53%
400	96.25%	88.73%
500	97.25%	89.60%
600	98.25%	90.07%
700	99.00%	90.13%
800	99.13%	90.53%
900	99.63%	90.53%
1000	99.88%	90.73%
1100	99.88%	90.67%

〈표 4〉 십진 데이타에 대한 BPN의 분류성능(4 클래스)

Classification Accuracy		
Iteration	Training data	Testing Data
200	91.63%	87.13%
300	92.75%	89.67%
400	93.13%	90.20%
500	93.75%	90.20%
600	94.00%	90.33%
700	94.38%	90.40%
800	94.50%	90.87%
900	94.75%	91.20%
1000	95.00%	91.33%
1100	95.13%	91.13%

8 클래스의 데이타를 사용한 실험에서는 퍼지집합이 유사한가 아닌가를 판단하기위한 임계값으로는 Hamming 거리 3.5가 사용되었다. 각 퍼지집합을 2개로 나눈후, 퍼지집합의 전체 갯수는 44개였다. 이중에서 클래스 분리정도가 큰 23개의 퍼지집합만을 선택했다. 그러므로 퍼지 입력표현을 갖는 Backpropagation Network (BPN)인 경우, 입력노드의 수는 1개의 bias를 포함하여 24개이고, 은닉층의 노드수도 1개의 bias를 포함하여 24개로 하였고 출력노드는 8개로 잡았다. 이진 데이타를 사용한 실험에서는 각 속성을 8 비트로 이진화하여 사용하였다. 이 경우 입력노드의 갯수는 bias를 포함하여 65개이고, 은닉층의 노드수도 1개의 bias를 포함하여 65개로 하였고 출력노드는 8개로 잡았다. 십진 데이타를 사용한 실험에서는 모든 데이타를 4클래스의 경우와 마찬가지로 각 속성을 정규화 하여 신경회로망의 입력으로 사용하였다. bias를 포함하여 9개이고, 은닉층의 노드수는 1개의 bias를 포함하여 16개로 하였고 출력노드는 8개로 잡았다. 학습률은 모든 실험에서 똑같이 0.002로 잡았고, 가중치(weight)는 −0.5에서 0.5사이의 값들로 초기화하여 실험을 하였다.

위의 실험 결과들에서 알 수 있듯이 이진 데이타를 BPN의 입력으로 사용했을 경우의 분류 성능은 학습 데이타에 대해서는 거의 완벽하지만 테스팅 데이타에 대해서는 퍼지 입력 신호 표현을 사용한 경우보다 낮게 나타남을 알 수 있다. 십진 데이타를 BPN의 입력으로 사용했을 경우의 분류 성능은 4 클래스 데이타를 사용한 실험에서는 학습 데이타에 대해서는 세가지 입력 표현중에서 가장 낮지만 테스팅 데이타에 대해서는 이진 데이타의 경우 보다 높게 나온다. 십진 데이타를 BPN의 입력으로 사용했을 경우의 분류 성능은 8 클래스 데이타를 사용한 실험에서는 세가지 입력 표현중에서 가장 낮지만 노드의 수가 가장 작으므로 계산적인 면에서는 효율적이라고 할 수 있을 것이다. 퍼지 입력신호 표현을 사용한 경우에는 학습 데이타의 분류 성능은 이진 데이타를 입력으로 사용한 경

우보다 조금 떨어지지만 테스팅 데이타에 대해서는 가장 높은 결과가 나왔다. 즉 제안된 퍼지 입력표현 방법이 신경회로망의 generalization에 도움이 됨을 알 수 있다.

〈표 5〉 퍼지 입력표현에 대한 BPN의 분류 성능(4 클래스)

Classification Accuracy		
Iteration	Training data	Testing Data
200	94.88%	92.13%
300	95.50%	92.47%
400	95.88%	92.93%
500	96.00%	93.20%
600	96.25%	93.53%
700	96.75%	93.60%
800	96.88%	93.47%
900	97.13%	93.73%
1000	97.13%	93.87%
1100	97.63%	93.80%

〈표 6〉 이진 데이타에 대한 BPN의 분류 성능(8클래스)

Classification Accuracy		
Iteration	Training Data	Testing Data
100	86.63%	82.00%
200	91.44%	85.63%
300	93.57%	87.10%
400	94.66%	88.00%
500	95.06%	88.93%
600	95.56%	89.30%
700	96.25%	89.47%
800	96.69%	89.63%
900	96.75%	89.83%
1000	97.19%	90.07%

〈표 7〉 십진 데이타에 대한 BPN의 분류 성능(8클래스)

Classification Accuracy		
Iteration	Training Data	Testing Data
100	72.06%	70.53%
200	77.50%	77.10%
300	82.19%	82.63%
400	82.69%	83.77%
500	86.63%	87.03%
600	88.75%	88.10%
700	89.81%	88.53%
800	90.31%	89.00%
900	90.81%	89.30%
1000	91.25%	89.40%

〈표 8〉 퍼지 입력 표현에 대한 BPN의 분류 성능(8 클래스)

Classification Accuracy		
Iteration	Training Data	Testing Data
100	89.06%	87.00%
200	90.81%	89.13%
300	90.63%	89.33%
400	90.81%	89.47%
500	91.13%	89.77%
600	91.25%	89.80%
700	91.38%	89.87%
800	91.75%	90.20%
900	91.81%	90.50%
1000	91.88%	90.50%

V. 결 론

본 논문에서는, 퍼지 입력신호 표현을 갖는 신경회로망을 제안했다. 실험 데이타로부터 멤버십함수를 자동으로 유도하는 방법이 제안되었고, 데이타들이 잘못 분류될 수 있는 가능성을 줄이기 위하여 유도된 퍼지집합들을 변형했다. 그리고 나서 클래스 분리정도가 큰 퍼지집합들을 선택한 후 유사정도 계산법을 사용하여 원래의 데이타를 다차원 벡터값으로 변환했다. 오류역전파 신경회로망(BPN)을 사용한 실험에서 퍼지 입력신호 표현을 사용한 신경회로망의 분류성능이 우수하다는 것을 확인할 수 있었다. 여기서 제안한 퍼지 입력신호 표현은 다른 신경회로망들과도 결합될 수 있을 것이다.

참 고 문 헌

1. B. Kosko and Seong-Gon Kong , Neural Networks and Fuzzy Systems : A Dynamical Approach to Machine Intelligence, Eglewood Cliffs, NJ : Prentice Hall, 1992.

2. C. H. Lee, "Fuzzy Logic in Control Systems : Fuzzy Logic Controller-Part I and II", IEEE Transactions on System, Man, and Cybernetics, Vol. 20, No. 2, pp. 404-435, March/April 1990.
3. C. T. Lin and C. S. G. Lee, "Neural-Network -Based Fuzzy Logic Control and Decision System", IEEE Transaction on Computers, Vol. 40, No. 12, pp. 1320-1336, December 1991.
4. H. Kang, H. G. Lee, Y. H. Kim and H. T. Jeon, "Automatic Tunning of Fuzzy Optimal Control Systems", International Fuzzy Systems Association World Congress '93, Vol. II, pp. 1195-1198.
5. J. M. Zurada, Introduction to Artificial Neural Systems, West Publishing Company, St. Paul, 1992
6. R. Krishnapuram and F. C-H Rhee, "Compact Fuzzy rulebase generation methods for computer vision", Proc. Second IEEE Int. Conf. Fuzzy Systems, March 1993, San Francisco, CA, Vol. II, pp. 809-814.
7. S. Cho and M. R. Lehto, "An Algorithm to compute the Degree of Match in Fuzzy Systems", Fuzzy Sets and Systems, Vol. 49, No. 3, August 1991.
8. S. Cho and M. R. Lehto, "A Fuzzy Scheme with Application to Expert Classification Systems for Computing the Degree of Match between a rule and an Assertion", Cybernetics and Systems, Vol. 50, August 1992.
9. O. K. Erosy and D. Hong, "Parallel, Self-Organizing, Hierachical Neural Networks", IEEE Transaction on Neural Networks, Vol. 1, No. 2, June 1990.