

# LVQ를 이용한 퍼지 규칙 생성

이 남 일<sup>†</sup> · 장 광 규<sup>\*\*</sup> · 임 한 규<sup>†</sup>

## 요 약

본 논문에서는 LVQ(Learning Vector Quantization)을 이용하여 퍼지 규칙의 수를 줄이는 방안을 제안하였다. 훈련 패턴이 많아지면 퍼지 규칙 수가 증가하게 되어 많은 기억용량과 많은 분류시간이 요구된다. 따라서 이러한 문제를 해결하기 위해서는 퍼지규칙의 수를 줄일 수 있는 방법이 강구되어야한다. 그러나, 퍼지 규칙의 수가 줄어들으로써 발생하는 성능의 하락을 최소화하기 위하여 양질의 초기 참조 패턴으로 훈련 한 후에, 퍼지 규칙을 생성한다. 시뮬레이션을 통해서 제안된 방법이 매우 효과적임을 알 수 있었다.

## Fuzzy Rules Generation Using the LVQ

Nam-Il Lee<sup>†</sup> · Gwang-Gyu Jang<sup>\*\*</sup> · HanKyu Lim<sup>†</sup>

## ABSTRACT

This paper is to investigate the method of reducing the number of fuzzy rules with the help of LVQ. A large number of training patterns usually leads to a large set of fuzzy rules that require a large computer memory and take a long time to perform classification. So, in order to solve these problems, it is necessary to study to minimize the number of fuzzy rules. However, so as to minimize the performance degradation resulting from the reduction of fuzzy rules, fuzzy rules are generated after training the high-quality initial reference pattern. Through the simulation, we confirm that the proposed method is very effective.

### 1. 서 론

퍼지는 우리가 일상생활에서 정확하게 표현하기 어려운 애매한 것에 대하여 수치적 표현이 가능하게 하는 것으로 1965년 Zadeh[1] 교수에 의해 처음으로 제안되었다.

그 이론은 자동차의 자동운행 분야와 패턴인식 분야 등에 이용되고 있다. 특히, Siy와 Chen[2]은 손으로 그린 선, 원의 일부를 퍼지 집합으로 고려하여 특징의 선택에 퍼지 로직을 사용했으며, 인간의 인식 학습으

로부터 경험상에 그려진 분류단계에 응용하였다. 그래서 퍼지 규칙들은 제어와 의사결정, 패턴인식을 포함하는 많은 분야에서 응용되고 있다.

그러나 아직도 해결되지 못한 문제점은 퍼지 영역과 멤버십 함수의 선택이 어렵고, 많은 학습 패턴을 입력하게 되면 그에 따른 퍼지 규칙 집합 수의 증가로 메모리의 과다요구와 분류수행에 긴 시간을 필요로 하는 것이다. 뿐만아니라 소량의 데이터라 할지라도 종류별, 픽셀별로 퍼지 규칙을 생성한다면 이것 역시 퍼지 규칙의 수가 많아져서 문제점으로 대두되고 있다.

이런 문제점을 해결하기 위해 많은 연구자들은 퍼지 규칙을 생성하는 다양한 방법들을 제안하였다. 이들 가운데 Wang과 Mendel[3]은 숫자데이터 쌍들로부터

† 종신회원 : 안동대학교 전자정보산업학부 교수  
\*\* 정 회 원 : 의성남부초등학교 교사  
논문접수 : 1998년 10월 7일, 심사완료 : 1999년 2월 19일

터 퍼지 규칙을 생성하는 방법을 제안하였는데, 측정된 많은 데이터 쌍과 언어학적인 정보들로부터 퍼지 규칙을 생성하여 하나의 공통적인 퍼지 연합 메모리뱅크(Fuzzy Associate-Memory Bank)로 결합하였으므로 몇몇 모순된 규칙들이 있을 수 있고 규칙의 수도 많아진다. 따라서 데이터 쌍들로부터 만들어지는 각각의 규칙에 등급을 배정하고 최대 등급을 갖는 규칙만 인정함으로써 규칙의 수를 줄여 나갔다. 그러나 이 방법도 점점 많은 학습 데이터를 이용함에 있어서 규칙이 늘어나므로 보다 높은 등급을 가진 새로운 규칙이 겹쳐지지 않는 한 규칙이 쌓이게되어 메모리 문제가 생겨 오래된 규칙들을 잘라내는 것으로 극복하였다.

Lee Kil-jae와 Bien Zeung-nam[4]은 물체 인식에 퍼지 로직을 사용하여 코너 매칭 알고리즘을 개발하였는데, 물체의 특징점 즉, 각 부분(Segment)의 길이, 각도, 모서리 등을 퍼지 규칙으로 만들어 모델 물체와 실제 물체간의 매칭된 부분을 최소화하기 위해 중첩된 확률 계산(Overlapping cost)으로 퍼지 규칙 수를 줄여 나갔다. 퍼지 규칙의 수가 많으면 모델을 묘사하는데 사용되는 모든 정보들과 매칭된 세그먼트 리스트를 짧은 시간에 추출하기에 너무 어렵고, 실제 세그먼트가 다중 모델 세그먼트로 묘사되는 경우가 자주 일어난다.

Zheru Chi과 Hong Yan[5]은 적은 수의 퍼지 규칙을 이용하여 필기체 숫자를 인식하는 방법을 제시하였는데 퍼지 규칙의 생성에서 숫자 각 픽셀별 가중값에 임계값을 일정하게 적용하여 이웃하는 퍼지 지역끼리 비교해서 임계값보다 작으면 비슷한 이웃 지역을 하나의 규칙으로 통합함으로써 퍼지 규칙 수를 줄여 나갔다. 이것은 비교적 분류에 영향을 끼치지 않는 부분까지 규칙을 세분화하여 생성할 뿐만 아니라 때에 따라서는 규칙이 하나로 될 가능성도 있다.

본 논문에서는 적용 임계값을 이용한 퍼지 규칙 감소 방법을 제안하였다.

제안한 방법에서는 적용임계값을 이웃하는 퍼지 지역과 단계적으로 크게 통합할 수 있도록 증가 함수로 만들었다. 가중값이 높아 상대적으로 분류에 영향을 많이 끼치는 부분은 규칙이 세분화 되도록 하였으며, 반대로 가중값이 낮은 부분은 크게 통합되어 퍼지 규칙수가 줄어들도록 하였다.

퍼지 규칙 생성을 위해 교사학습 알고리즘인 Learning Vector Quantization(LVQ)[6]를 이용했으며, 숫자

0~9까지 16x16 양질의 이미지 데이터 각각 10개씩 100개를 초기 참조 패턴으로 사용하여 학습률 0.01로 1000번 학습하였다. 초기 참조 패턴 학습을 통해 입력 데이터의 부류 분포에 근접하도록 픽셀별 명암도에 대한 가중값을 부여하였다. 갱신된 픽셀별 가중값을 정렬하여 근접하는 이웃 퍼지 지역에 초기값 0.0005, 8단계로 점점 증가하는 적용 임계값을 적용하였다.

실험 결과 픽셀별 명암도에 대한 가중값이 높은 부분은 분류에 비교적 영향을 많이 끼치므로 규칙 수가 세분화되었고, 반대로 가중값이 낮은 부분은 통합되어 전체적으로 퍼지 규칙수가 많이 감소되었음을 확인하였다.

본 논문의 제2장에서는 퍼지 이론과 규칙, 퍼지 규칙 생성, 제3장에서는 적용 임계값에 의한 퍼지 규칙 생성방법에 관한 제안을 설명하고, 제4장에서 실험 결과 및 분석, 제5장에서는 결론을 기술하였다.

## 2. 퍼지 규칙(Fuzzy Rule) 생성

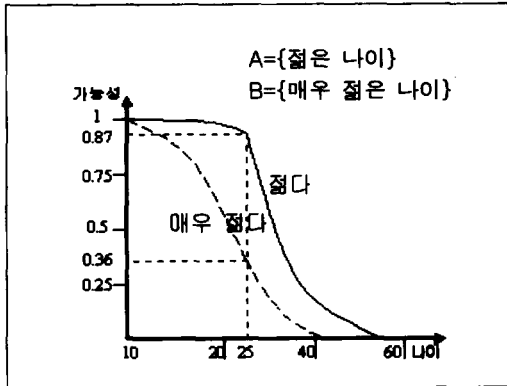
숫자 인식을 위한 역전파(Back Propagation) 모델이나 다층 퍼셉트론(Multi Layer Perceptron) 등에서 입력층, 중간층, 출력층 가운데 중간층 노드(Node)가 많으면 많을수록 그 인식률은 높으나 어느 정도 이상이 되면 인식률에 포화하는 경향이 있는데 이는 입력층과 중간층, 중간층과 출력층이 완전 연결되어 수행되기 때문에 메모리만 과다 요구하고 수행시간도 길어진다. 따라서 중간층을 적당하게 줄이고 어떤 기본형(Prototype)을 만들어 사용한다면 메모리 문제와 수행시간 문제를 해결할 수 있을 것이다. 이러한 관점에서 간편하게 규칙을 만든 기본형을 사용한다는 것은 너무나 중요한 사실이다. 이를 위해 Self-Organizing Map(SOM)[7] 알고리즘 변형인 LVQ 알고리즘을 이용하여 퍼지 규칙을 생성하고자 한다.

### 2.1 퍼지 이론과 규칙

일반적으로 인간이 복잡한 현상의 문제를 해결한다는 것은 대부분 복잡한 문제를 단순화시켜 우리가 이해할 수 있는 정도의 간단한 문제로 만든 다음 그 문제를 해결하는 것을 말하나 이렇게 복잡한 문제를 단순화시키면 정보의 손실이 많아지므로 정보의 손실이 적도록 하는 것을 연구하고 있다[8].

(그림 1)에서 횡축은 나이, 종축은 소속함수 값을

나타내며, 곡선은 퍼지 집합에 포함될 가능성을 나타내는데, 이때 소속함수 값이란 '젊은 나이'의 퍼지 집합에 포함될 수 있는 정도를 말한다.



(그림 1) '젊다'와 '매우 젊다'의 소속함수  
(Fig. 1) Functions of 'young' and 'very young'

0세인 사람은 당연히 '젊다'이며, 포함될 가능성은 1이고, 25세인 사람은 젊은 나이에 포함될 가능성이 0.87, 60세는 0이다. 이제 퍼지 집합을 왼쪽으로 이동하여 '매우 젊다'로 정의하면 40세 미만 되는 사람만이 '매우 젊다'란 퍼지 집합에 포함될 가능성이 있으므로 25세는 0.36의 가능성으로 바뀌게 된다.

이처럼 퍼지 이론의 응용은 여러 분야에서 이용되고 있는데 특히, 시스템의 특성이 복잡하여 기존의 정량적인 방법으로 해석할 수 없거나, 얻어지는 정보가 정성적이고 부정확하며, 불확실한 경우가 기존 제어기보다 더 우수한 제어 결과를 나타내므로 시스템의 복잡성이 어느 정도이상 되면 정량적인 방식으로는 시스템 가동이 부정확하며, 의미가 있는 기술이 불가능하게 된다는 것이다.

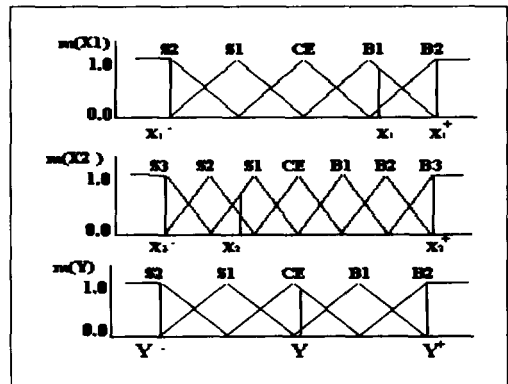
퍼지 규칙은 조건부와 결론부로 구성되며 일반적인 퍼지 규칙의 형태는

IF  $x_1$  is  $R_1$  and  $x_2$  is  $R_2$  and ...  $x_n$  is  $R_n$  THEN  $Y$  is  $A$

이다.

즉, 다수개의 퍼지 집합으로 구성되어 있는 조건부와 하나의 결론부를 갖는다. 조건부와 결론부는 퍼지 변수(Fuzzy variable)와 퍼지 값(Fuzzy value)들로 구성되며, 결론부는 숫자 가능성으로 표현된다[3].

(그림 2)는 퍼지 규칙 생성의 일반적인 예를 나타낸 것이다.  $X_1$ 과  $X_2$ 는 입력,  $Y$ 가 출력인 데이터 쌍들이 있다고 가정하면, 이를 퍼지 지역으로 분할하여 삼각형 구성요소를 갖게 한다. 삼각형 중심에 있는 정점은 구성요소와 일치하며, 이웃에 있는 정점은 0의 구성요소 값을 가진다. 따라서, 구성요소 값의 최대 등급을 갖는 지역에 할당한다. 즉,  $X_1(0.8 \text{ in } B_1)$ ,  $X_2(0.7 \text{ in } S_1); Y(0.9 \text{ in } CE)$ 으로 표현되는데 이를 퍼지 규칙으로 나타내면 IF  $X_1$  is  $B_1$  and  $X_2$  is  $S_1$ , THEN  $Y$  is  $CE$  이 된다.



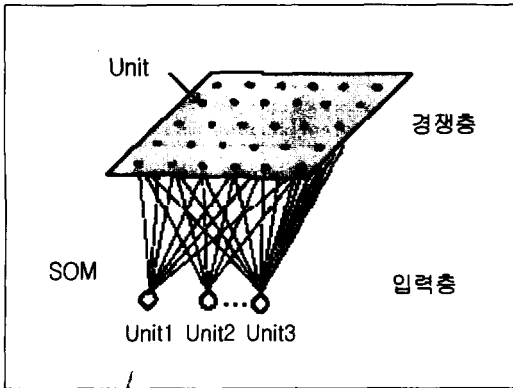
(그림 2) 퍼지 규칙 생성의 일반적인 예  
(Fig. 2) General example of generation of Fuzzy rules

## 2.2 LVQ(Learning Vector Quantization)를 이용한 퍼지 규칙의 최소화

인간은 태어나면서 부모에게 배우기도 하지만, 대부분의 아이들은 부모들이 가르치지도 않는 많은 것을 자연스럽게 터득하게 된다. 이러한 것에 대한 설명 중 하나가 1962년 Teuvo Kohonen[7]에 의해 제안된 SOM이다. 자기조직화(Self-Organizing)란 주어진 입력 패턴에 대하여 정확한 해답을 미리 주지 않고 스스로 학습할 수 있는 능력을 말한다. 즉, 입력 패턴에 대하여 해답 없이도 스스로 학습하는 능력으로, 자신이 선택되기 위하여 각 뉴런들은 경쟁한다는 뜻이다.

SOM의 구조를 살펴보면 (그림 3)과 같이 두 개의 층으로 되었는데, 데이터를 입력받는 입력층과 입력받은 값에 완전 연결되어 각 뉴런들간에 경쟁하는 경쟁층이 있다. 경쟁층의 각 뉴런들은 상호간의 연결강도(Weight) 벡터와 입력벡터가 얼마나 가까운가를 계산한다. 여기서 입력 벡터와의 거리가 가장 가까운 뉴런

이 승자 뉴런이 되는데, 이 승자 뉴런만이 출력신호를 보낼 수 있는 유일한 뉴런이다. 승자 뉴런에 이웃한 뉴런들은 또한 학습이 가능하고, 학습이 반복될수록 이웃 뉴런들의 학습 반경은 줄어들게 되어 마침내 승자 뉴런만이 남게 된다. 이런 과정을 거치면서 연결강도를 갱신한다[7].



(그림 3) SOM의 구조  
(Fig. 3) The structure of SOM

SOM은 비 교사학습 알고리즘으로 새로운 입력데이터를 학습하여 종류별, 계급별, 계층별 등 그룹으로 분류하는데 편리한 반면, SOM 알고리즘의 변형인 LVQ는 교사학습 알고리즘으로서, 정밀하게 부류에 근접시킬 수 있어 분류 수행에 시간이 적게 들고, LVQ 자체적으로 인식할 수 있는 기능까지 있어서 LVQ 알고리즘을 사용했다.

LVQ 알고리즘은 초기 참조 패턴을 적절하게 선택한 후, 학습을 통하여 입력 데이터의 부류 분포에 근접하도록 갱신시킨다. 따라서 각 참조 패턴의 초기 값을 어떻게 설정하느냐가 대단히 중요하다. 현재의 LVQ 알고리즘은 LVQ2, LVQ3 등 개선된 알고리즘이 있으나 여기서는 LVQ1을 기준으로 설명하였다.

M개의 영역으로 구성된 표본 공간상에서 각 부류의 참조 패턴이 해당부류의 표본 패턴들이 이루는 군집의 중심에 위치한다고 가정한다면 일반적으로 한 부류에 여러 개의 참조패턴을 배정할 수 있다. 미지의 입력 패턴 x는 참조 패턴 내에서 최 근접 거리에 위치한 참조 패턴의 소속 부류와 같은 부류에 해당하는 것으로 결정될 수 있다. 입력 패턴 x와 최 근접 거리에 위치한 참조패턴은 식

$$c = \underset{i}{\text{MIN}} \{ \|x - m_i\| \} \quad (1)$$

에 의해서 결정되는데, 이렇게 결정된 참조 패턴을  $m_c$ 로 표기하면, 최 근접 이웃 분류에서 오분류율을 최소화시키는 참조패턴  $m_c$ 는 학습 데이터 집합을 사용한 학습에 의해 점진적으로 결정될 수 있다.  $x(t)$ 를 시각 t에서의 참조 패턴이라고 하고  $m_c(t)$ 를 시각 t에서의 참조 패턴이라고 했을 때, 초기 참조 패턴을 적절하게 선택한 후의 LVQ1 알고리즘은

$$m_c(t+1) = m_c(t) + \alpha(t)[x(t) - m_c(t)] \quad (2)$$

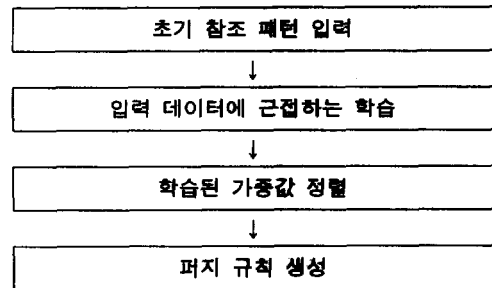
: x와  $m_c$ 가 같은 부류에 속할 경우

$$m_c(t+1) = m_c(t) - \alpha(t)[x(t) - m_c(t)] \quad (3)$$

: x와  $m_c$ 가 서로 다른 부류에 속할 경우

와 같다. 식(2),(3)에서  $\alpha(t)$ 는 학습율을 나타내는 상수로  $0 < \alpha(t) < 1$ 의 범위 내에서 일정한 값을 갖거나 시간이 경과함에 따라 단조 감소한다. 모든 참조 패턴에 각각의 학습율을 배정함으로써 알고리즘이 빠르게 수렴한다[6].

본 논문에서 퍼지 규칙 생성은 LVQ 알고리즘의 학습과정과 인식과정 중에서 학습과정이 끝난 후에 이루어지는데, 16×16 명암도 숫자 자료를 사용하였다. 최종 퍼지 규칙에 나타나는 값은 각각 숫자가 가져야 될 명암 값에 대한 가중값을 나타내게 되며, 퍼지 규칙 생성 과정은 (그림 4)와 같이 LVQ의 초기참조 패턴의 입력단계와 학습단계, 퍼지 규칙 생성 단계 4단계로 구성하였다[9].

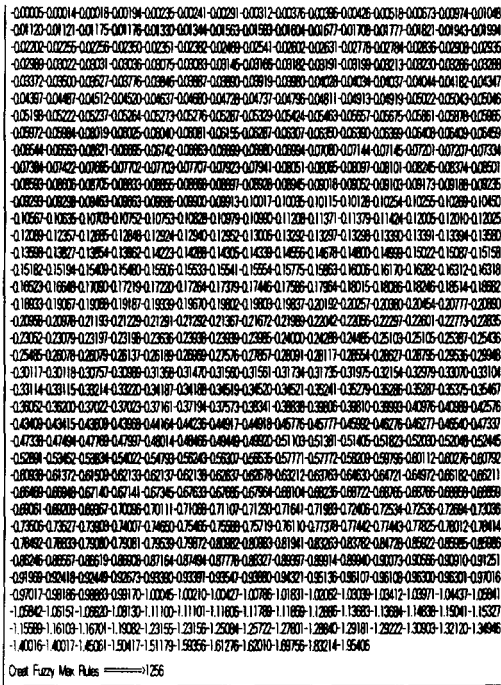


(그림 4) 퍼지 규칙 생성 과정  
(Fig. 4) Course of generation of Fuzzy rules

### 2.2.1 LVQ에 초기 패턴 입력

LVQ에 입력패턴들의 특징을 충분히 반영할 수 있는 양질의 패턴을 선정하여 입력으로 준다. 입력에 필





(그림 8) 가중값이 높은 순서부터 정렬  
(Fig. 8) Weights, arranged according to decrease-sorting

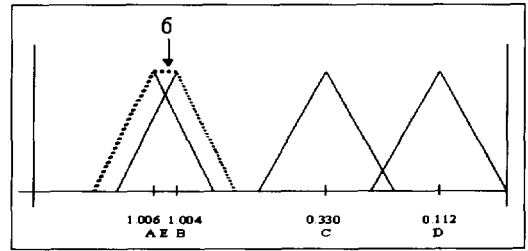
특히, 본 논문에서와 같이 단순한 요소 즉, 명암도 같은 처리에 아주 효율적이다. (그림 8)은 학습된 가중값을 숫자 0~9까지 내림차순으로 정렬한 것이다.

2.2.4 퍼지 규칙 생성  
정렬된 가중값은 수식

$$\left. \begin{aligned} |w(i) - w(i+1)| > \delta : \text{Then Assign A New Rule} \\ |w(i) - w(i+1)| < \delta : \text{Then Retain The Rule} \end{aligned} \right\} (4)$$

에 따라 규칙을 부여하는데, 이웃하는 영역이 퍼지 구성요소 값의 차가 임계값(δ)보다 작으면 비슷한 영역을 합하여 규칙을 생성하고, 그렇지 않으면 새로운 규칙을 부여한다. 이와 같이 연속된 이웃영역에 처음부터 끝까지 일정한 임계값으로 비교하여 처리한다. 따라서 많은 퍼지 영역은 하나의 삼각형 대신 사다리꼴 모양을 가진다[5].

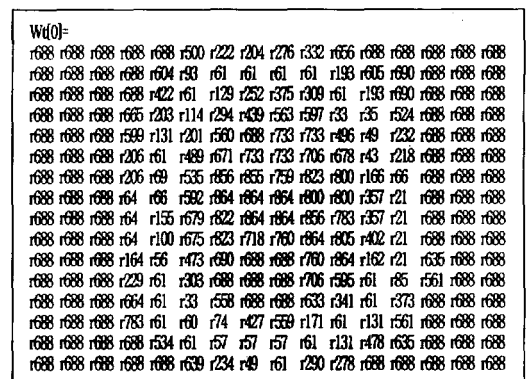
이를 도식화하면 (그림 9)와 같고, 임계값(δ)을 0.005로 정하여 점선 부분이 하나의 규칙으로 생성됨을 알 수 있다.



(그림 9) 퍼지 규칙 생성의 예  
(Fig. 9) An illustration of generation of Fuzzy rules

연속된 퍼지 영역 A, B, C점의 숫자 명암도에 대한 가중값이 각각 1.006, 1.004, 0.030일 때, |A-B|의 값이 1.006 - 1.004 = 0.002이므로 이 것은 임계값(δ) 0.005보다 작기 때문에 묶어서 같은 규칙을 부여하였다. 또, A, B의 중심 E와 C의 차, |E-C|의 값이 1.005 - 0.330 = 0.675이므로 이것은 임계값(δ) 0.005보다 크기 때문에 새로운 규칙을 생성하였다. C와 D의 차이도 역시 임계값(δ) 0.005보다 크기 때문에 또 새로운 규칙을 생성하였다. 이로써, 퍼지 규칙이 3개 생성되었음을 알 수 있다.

숫자 0에 대한 임계값을 적용하지 않은 상태에서 퍼지 규칙을 생성한 결과 많은 규칙이 생겼으며, 이는 숫자 0의 명암도의 가중값에 따라 각각 다르게 규칙을 부여하였다. 숫자 0에 대한 임계값을 적용하지 않은 상태에서 규칙 생성 결과는 (그림 10)과 같다.



(그림 10) 임계값이 주어지지 않은 경우의 퍼지 규칙  
(Fig. 10) Fuzzy rules, without threshold value

숫자 0에 대한 임계값을 0.005로 일정하게 주었을 때 퍼지 규칙은 (그림 11)과 같다.

Wd(0)=															
57	57	57	57	58	r4	r3	r3	r3	r3	r3	r3	r3	r3	r3	r3
57	57	57	57	r4	r2	r2	r2	r2	r3	r4	57	57	57	57	57
57	57	57	57	r3	r0	r5	r3	r3	r2	r1	57	57	57	57	57
57	57	57	r0	r0	r5	r6	r4	r4	r2	r2	r2	57	57	57	57
57	57	57	r7	r2	r3	r7	57	57	r2	r3	57	57	57	57	57
57	57	57	r3	r8	r0	57	57	r9	r6	r1	r3	57	57	57	57
57	57	57	r3	r2	r3	r3	r3	r3	r3	r1	r2	57	57	57	57
57	57	57	r2	r2	r4	r9	r9	r9	r9	r8	r18	57	57	57	57
57	57	57	r2	r0	r0	57	57	r9	r9	r9	r2	57	57	57	57
57	57	57	r2	r2	r5	57	57	r9	r9	r2	r2	57	57	57	57
57	57	57	r0	r18	r0	57	57	57	r3	r1	r2	r2	57	57	57
57	57	57	r3	r18	r3	57	57	57	r4	r2	r2	r7	r9	57	57
57	57	57	r0	r21	r21	r7	57	57	r0	r5	r2	r9	r8	57	57
57	57	57	r1	r21	r21	r2	r4	r4	r3	r2	r2	r4	r9	57	57
57	57	57	r3	r4	r2	r2	r2	r2	r2	r2	r2	r2	r2	57	57
57	57	57	r1	r5	r3	r2	r2	r3	r4	r8	57	57	57	57	57

(그림 11) 일정한 임계값을 적용했을 경우의 퍼지 규칙

(Fig. 11) Fuzzy rules with fixed threshold value

(그림 11)에서 나타난 숫자 "0"에 대한 결과를 퍼지 규칙 IF~THEN~ 문장으로 표현한 것이 (그림 12)와 같으며, 퍼지 규칙이 30개임을 알 수 있다.

IF																			
F1	is	r57	and	F2	is	r57	and	F3	is	r57	and	F4	is	r57	and	F5	is	r48	and
F6	is	r44	and	F7	is	r34	and	F8	is	r31	and	F9	is	r34	and	F10	is	r36	and
F11	is	r50	and	F12	is	r57	and	F13	is	r57	and	F14	is	r57	and	F15	is	r57	and
F16	is	r49	and	F17	is	r27	and	F18	is	r22	and	F19	is	r22	and	F20	is	r22	and
F21	is	r22	and	F22	is	r31	and	F23	is	r47	and	F24	is	r57	and	F25	is	r57	and
F26	is	r57	and	F27	is	r57	and	F28	is	r57	and	F29	is	r57	and	F30	is	r57	and
F31	is	r57	and	F32	is	r38	and	F33	is	r20	and	F34	is	r25	and	F35	is	r34	and
F36	is	r38	and	F37	is	r37	and	F38	is	r22	and	F39	is	r31	and	F40	is	r57	and
...																			
F221	is	r49	and	F222	is	r57	and	F223	is	r57	and	F224	is	r57	and	F225	is	r57	and
F226	is	r57	and	F227	is	r57	and	F228	is	r38	and	F229	is	r44	and	F230	is	r22	and
F231	is	r22	and	F232	is	r22	and	F233	is	r22	and	F234	is	r22	and	F235	is	r27	and
F236	is	r42	and	F237	is	r52	and	F238	is	r57	and	F239	is	r57	and	F240	is	r57	and
F241	is	r57	and	F242	is	r57	and	F243	is	r57	and	F244	is	r57	and	F245	is	r57	and
F246	is	r55	and	F247	is	r34	and	F248	is	r22	and	F249	is	r22	and	F250	is	r34	and
F251	is	r34	and	F252	is	r38	and	F253	is	r57	and	F254	is	r57	and	F255	is	r57	and
F256	is	r57																	
THEN																			
it is digit 0																			

(그림 12) 'IF~THEN~'으로 표현한 숫자 0의 퍼지 규칙

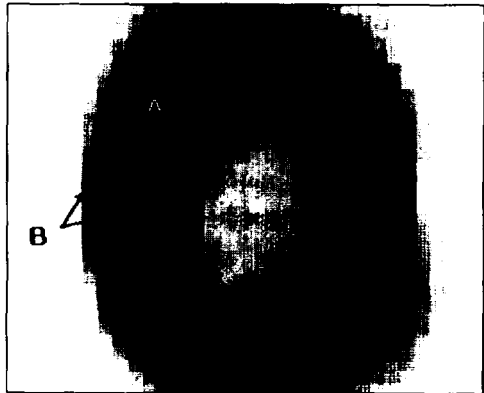
(Fig. 12) Fuzzy rules of digit 0, illustrated in a sentence, 'IF~THEN~'

퍼지 규칙의 생성에서 임계값을 일정하게 적용함으로써 이웃하는 퍼지 지역과 규칙적인 배분에 의해 임계값보다 작으면 하나의 규칙으로 통합하여 퍼지 규칙의 수를 줄여 나갔다. 이것은 (그림 11)의 중앙부분과 같이 실제로 분류 수행에 비교적 영향을 끼치지 않는 부분까지 규칙을 세분화할 뿐만 아니라 때에 따라서는 규칙이 하나로 될 수도 있는 문제점이 있다.

### 3. 적응 임계치(Adaptive Threshold)에 의한 퍼지 규칙 생성 방법 제안

퍼지 규칙 수가 증가하면 많은 기억용량의 요구와 분류 수행에도 많은 시간이 요구된다. 그러나 규칙의 수가 적은 것만이 좋은 것은 아니다. 규칙의 수가 줄어들어서 발생하는 인식을 하락을 가져올 수 있다. 이를 최소화하기 위해 Zheru Chi와 Hong Yan이 제안한 방법에서 숫자 분류에 비교적 의미가 없는 부분, 즉 명암도에 대한 가중값이 낮은 부분은 임계값을 크게 하여 규칙을 만들어나가고, 명암도에 대한 가중값이 큰 부분은 임계값을 좀더 작게 하여 규칙을 세분화한다.

(그림 13)은 10개의 중첩된 숫자0에 대한 명암도를 나타낸 것으로서, A부분은 명암도에 대한 가중값이 높아 분류에 비교적 많은 영향을 끼치는 부분으로 규칙을 세분화하고, B부분은 명암도에 대한 가중값이 낮아 분류에 영향을 비교적 적게 끼치는 부분으로 규칙을 통합하여 퍼지 규칙을 생성한다.



(그림 13) 10개 중첩된 숫자0에 대한 명암도 (Fig. 13) Brightness of digit 0, when overlapped ten times

M개의 벡터값을 t개의 단계로 임계값을 적용한다면 임계값은  $\delta(t+1) = \delta(t) + \alpha$ 로 결정된다. 여기서  $\alpha$ 는 초기 임계값을 나타내는 상수로서  $0 < \alpha$ 인 일정한 값을 갖는다.

정렬된 가중값을 수식

$$\left. \begin{aligned} |w(i) - w(i+1)| > \delta(t) : \text{Then Assign A New Rule} \\ |w(i) - w(i+1)| < \delta(t) : \text{Then Retain The Rule} \end{aligned} \right\} (5)$$

에 따라 규칙을 부여하는데 이웃하는 영역과 퍼지 구








(그림 17) 일정한 임계값에 의한 숫자 0-9까지 퍼지 규칙

(Fig. 17) Fuzzy rules of digits, 0~9, with fixed threshold value

일정한 임계값을 적용하였을 때 상태가 가장 양호한 것과 적용 임계값을 적용했을 때 상태가 가장 양호한 것을 비교해 보았다. 객관적인 비교는 될 수 없지만 같은 환경에서 비교했을 때 숫자 0에 임계값을 일정하게 적용한 것은 (그림 18)의 왼쪽과 같이 30개의 많은 퍼지 규칙이 생성되었다.


(그림 18) 숫자 0에 대해 일정한 임계값과 적응 임계값을 적용했을 경우

(Fig. 18) Fuzzy rules of digit, 0, with fixed or adaptive threshold value

이는 명암도에 대한 가중값이 낮은 부분, 즉 분류에

비교적 영향을 끼치지 않는 부분까지 세분화되어 규칙 수가 늘어났다. 반면 초기 임계값에서 단계적으로 증가하는 적응 임계값을 적용한 것은 (그림 18)의 오른쪽과 같이 11개의 적은 퍼지 규칙이 생성되었다. 이는 명암도의 가중값이 높은 부분은 일정한 임계값을 적용했을 때와 마찬가지로 규칙이 세분화되었고, 명암도의 가중값이 낮은 가운데 부분과 바깥 부분은 분류에 비교적 영향을 많이 끼치지 않아 규칙을 통합하여 수를 줄였다.

본 실험에서 양질의 데이터로 적은 수의 퍼지 규칙을 생성하였으므로 이것을 기본형(prototype)으로 사용하여 모든 숫자 인식에 직접 적용, 참조할 수 있으며, 새로운 인식모델로 활용할 수 있다. 이것은 분류와 인식에 퍼지 규칙을 생성하는 단계를 생략하고, 곧바로 인식하는 단계로 들어갈 수 있어, 많은 학습 패턴 입력에 따른 퍼지 규칙 집합수가 증가하게 되고 기억용량도 더욱 많이 요구하게 되며, 분류 수행에도 긴 시간을 필요로 하는 인식분야에서 제기되는 문제점을 해결할 수 있다.

### 5. 결 론

본 논문에서는 적응 임계값을 이용한 퍼지 규칙 감소 방법을 제안하였다.

제안한 방법에서는 이웃하는 퍼지 지역과 단계적으로 크게 통합할 수 있도록 적응임계값을 증가 함수로 만들었다. 가중값이 높아 상대적으로 분류에 영향을 많이 끼치는 부분은 규칙이 세분화 되도록 하였으며, 반대로 가중값이 낮은 부분은 크게 통합되어 퍼지 규칙수가 줄어들도록 하였다.

퍼지 규칙 생성을 위해 교사학습 알고리즘인 LVQ를 이용했으며, 숫자 0~9까지 16×16 양질의 이미지 데이터 각각 10개씩 100개를 초기 참조 패턴으로 사용하여 학습률 0.01로 1000번 학습하였다. 초기 참조 패턴 학습을 통해 입력 데이터의 부류 분포에 근접하도록 픽셀별 명암도에 대한 가중값을 부여하였다. 갱신된 픽셀별 가중값을 정렬하여 근접하는 값들에 초기값 0.0005, 8단계로 점점 증가하는 적응 임계값을 적용하였다.

실험결과 숫자 0에서 9까지 16×16 픽셀별 명암도에 대한 가중값은 2560개중 중복되는 것을 제외하면 1256개였다. 일정한 임계값을 적용하였을 경우는 전체

가중값 중 108개의 퍼지 규칙이 만들어졌으나, 적용 임계값을 적용했을 경우는 43개의 적은 퍼지 규칙이 만들어졌다. 픽셀별 명암도에 대한 가중값이 높은 부분은 분류에 상대적으로 영향을 많이 끼치므로 규칙수가 세분화되었고, 반대로 가중값이 낮은 부분은 통합되어 전체적으로 퍼지 규칙수가 많이 감소되었음을 확인하였다.

**참 고 문 헌**

[1] Zadeh, L. A., "Fuzzy sets," *Information Control*, pp. 338~353, 1965. 8.  
 [2] Siy, P., & Chen, C. S. "Fuzzy logic for hand-written numeral character recognition." *IEEE Transactions on Systems, Man and Cybernetics*, SMC-4, pp.570-574, 1974.  
 [3] Wang, L., & Mendel, J. M. "Generating fuzzy rules by learning from examples," *Proceedings of the 1991 IEEE International Symposium on Intelligent Control*, pp.263-268, Virginia, USA. 1991.  
 [4] Lee, K. j., & Bien, Z. N., "A Corner Matching Algorithm Using Fuzzy Logic."  
 [5] Zheru, C. & Hong, Y., "Handwritten Numeral Recognition Using a Small Number of Fuzzy Rules With Optimized Defuzzification Parameters," The Hong Kong Polytechnic University and The University of Sydney, Nov., 1994.  
 [6] 송시현, 이성환, "개선된 LVQ3와 시플레이티드아닐리의 결합을 통한 대용량 필기체 문자 인식을 위한 최적의 참조 모델 설계", 「정보과학회논문지」, 제21권 제6호, pp.59-1607, 1994. 6.  
 [7] Kohonen, T., "The self-organizing map," *Proceedings of IEEE*, 78(9), pp.1464-1480.  
 [8] 이광형, 오길록, 「퍼지이론 및 응용 I, II」, 홍릉과학출판사, I 권 pp.1.4-2.18, II 권, pp.5.3-7.28, 1992. 10.



**이 남 일**

e-mail : nilee@andong.ac.kr

1981년 경북대학교 전자공학과(학사)  
 1986년 경북대학교 대학원 전자공학과 전산전공(석사)  
 1993년 경북대학교 대학원 전자공학과 전산전공(박사)

1983년~1984년 현대중공업 사원

1989년~현재 안동대학교 전자정보산업학부 부교수

관심분야 : 뉴럴 네트워크, 패턴 인식, 컴퓨터 그래픽스



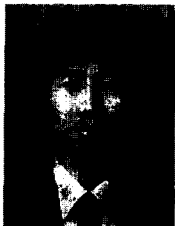
**장 광 규**

e-mail : ggjang@andong.ac.kr

1986년 대구대학교 무역학과(학사)  
 1999년 안동대학교 대학원 컴퓨터공학과(석사)  
 1996년 3월~1998년 2월 경상북도 교육과학연구원

1999년 3월~현재 의성 남부초등학교 교사

관심분야 : 가상 현실



**임 한 규**

e-mail : hklim@andong.ac.kr

1981년 경북대학교 전자계산기공학 전공(학사)  
 1984년 연세대학교 산업대학원 전산 전공(석사)  
 1997년 성균관대학교 대학원 전자계산공학 전공(박사)

1981년~1982년 대한주택공사 전산실

1982년~1986년 한국전자통신연구소 위성통신연구실 연구원

1986년~1994년 한국IBM 소프트웨어연구소 선임연구원

1994년~1998년 한서대학교 전산정보학과 조교수

1998년~현재 안동대학교 전자정보산업학부 조교수

관심분야 : 자연언어처리, 영상처리, 멀티미디어