

# 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 줄어들어서 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 쌍들로부

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

터 퍼지 규칙을 생성하는 방법을 제안하였는데, 수집된 많은 데이터 쌍과 언어학적인 정보들로부터 퍼지 규칙을 생성하여 하나의 공통적인 퍼지 연합 메모리 뱅크(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 쌓이게 되어 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 Chi와 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 적응 임계값을 이용한 퍼지 규칙 감소 방법을 제안하였다.

제안한 방법에서는 적응임계값을 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 픽셀의 이진식 데이터 약 10개씩 100개를 초기 참조 패턴으로 사용하여 학습률 0.01로 1000번 학습하였다. 초기 참조 패턴 학습을 통해 입력 데이터의 부류 분포에 균질하도록 픽셀별 명암도에 대한 가중값을 부여하였다. 생성된 픽셀별 가중값을 정렬하여 인접하는 이웃 퍼지 지역에 초기값 0.0005, 8단계로 점점 증가하는 적응 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 적응 임계값에 의한 퍼지 규칙 생성방법에 관한 제안을 설명하고, 제4장에서 실험 결과 및 분석, 제5장에서는 결론을 기술하였다.

## 2. 퍼지 규칙(Fuzzy Rule) 생성

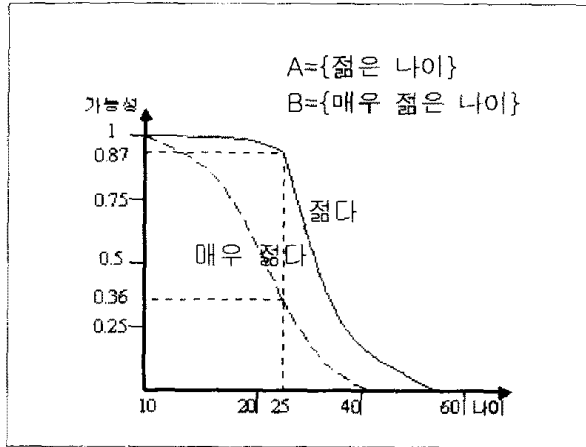
숫자 인식을 위한 역전파(Back Propagation) 모델이나 다층 퍼셉트론(Multi Layer Perceptron) 등에서 입력층, 중간층, 출력층 가운데 중간층 노드(Node)가 많으면 많을수록 그 인식률은 높으나 어느 정도 이상이 되면 인식률에 포화하는 경향이 있는데 이는 입력층과 중간층, 중간층과 출력층이 완전 연결되어 수행되기 때문에 메모리만 과다 요구하고 수행시간도 길어진다. 따라서 중간층을 적당하게 줄이고 어떤 기본형(Prototype)을 만들어 사용한다면 메모리 문제와 수행시간 문제를 해결할 수 있을 것이다. 이러한 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 값이런 '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.

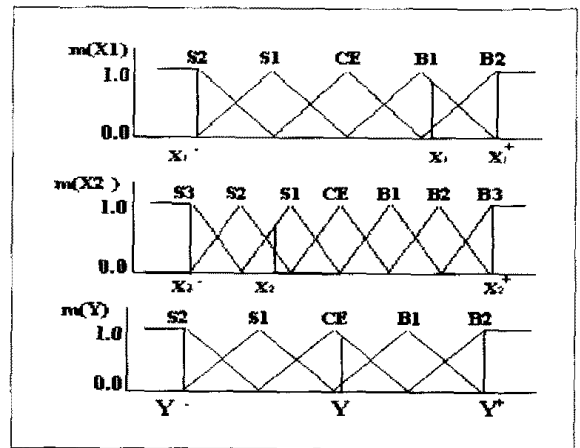
퍼지 규칙은 조건부와 결론부로 구성되며 일반적인 퍼지 규칙의 형태는:

IF  $x_1$  is R1 and  $x_2$  is R2 and ...  $x_n$  is Rn 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,  $X_1(0.8$  in  $B_1)$ ,  $X_2(0.7$  in  $S_1)$ ;  $Y(0.9$  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.



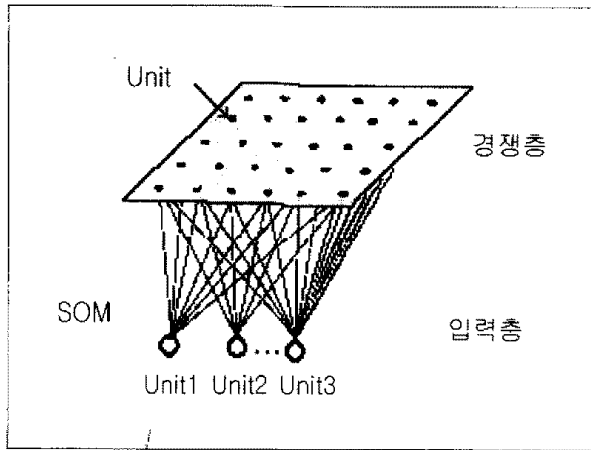
(그림 2) 퍼지 규칙 생성의 일반적인 예  
(Fig. 2) General example of generation of Fuzzy rules

### 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 없어도 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 반영은 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

$$c = \underset{i}{\text{MIN}} \|x - m_i\| \tag{1}$$

에 의해서 결정되는데, 이렇게 결정된 참조 패턴을  $m_c$ 로 표기하면, 최 근접 이웃 분류에서 오분류율을 최소화시키는 참조패턴  $m_i$ 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

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

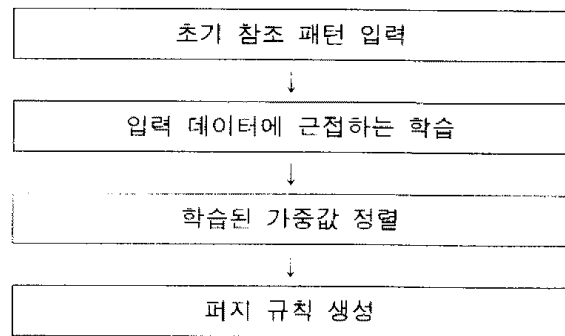
: x와  $m_c$ 가 같은 부류에 속할 경우 (2)

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

: x와  $m_c$ 가 서로 다른 부류에 속할 경우 (3)

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. 최종의 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].

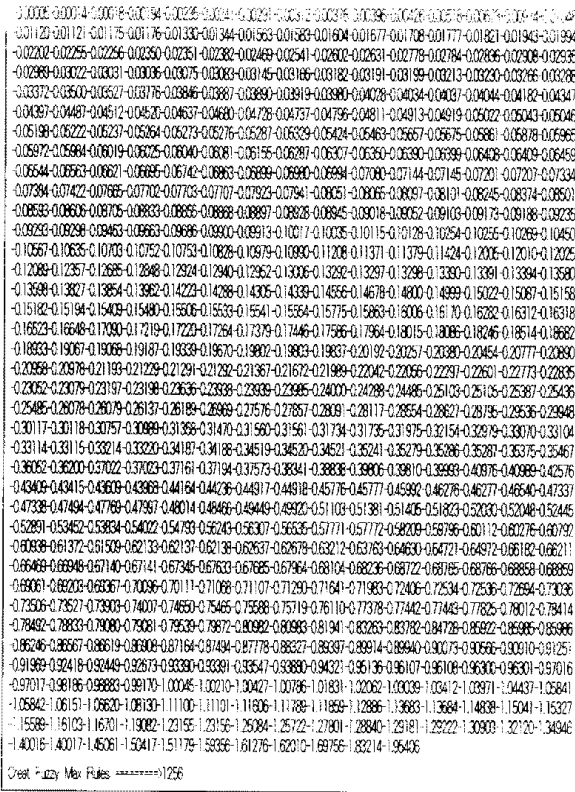


(그림 4) 퍼지 규칙 생성 과정  
(Fig. 4) Course of generation of Fuzzy rules

### 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

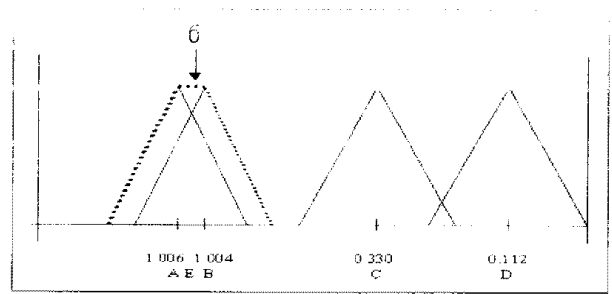
특히, 본 논문에서와 같이 단순한 요소 즉, 명암도 같은 처리에 아주 효율적이다. (그림 8)은 학습된 가중값을 숫자 0~9까지 내림차순으로 정렬한 것이다.

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( $\delta$ )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].

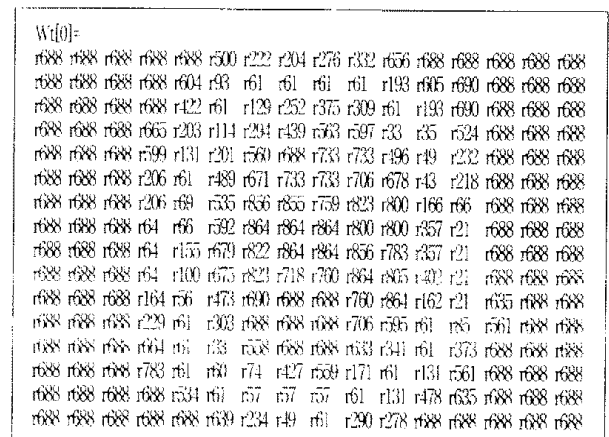
이를 도식화하면 (그림 9)와 같고, 임계값( $\delta$ )을 0.005로 정하여 점선 부분이 하나의 규칙으로 생성됨을 알 수 있다.



(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  $1.006 - 1.004 = 0.002$ 이므로 이 것은 임계값( $\delta$ ) 0.005보다 작기 때문에 묶어서 같은 규칙을 부여하였다. 또, A, B의 중심 E와 C의 차,  $|E - C|$ 의 값이  $1.005 - 0.330 = 0.675$ 이므로 이것은 임계값( $\delta$ ) 0.005보다 크기 때문에 새로운 규칙을 생성하였다. C와 D의 차이도 역시 임계값( $\delta$ ) 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 퍼지 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.

W(0)	r1	r2	r3	r4	r5	r6	r7	r8	r9	r10	r11	r12	r13	r14	r15	r16	r17	r18	r19	r20	r21	r22	r23	r24	r25	r26	r27	r28	r29	r30	r31	r32	r33	r34	r35	r36	r37	r38	r39	r40	r41	r42	r43	r44	r45	r46	r47	r48	r49	r50	r51	r52	r53	r54	r55	r56	r57	r58	r59	r60	r61	r62	r63	r64	r65	r66	r67	r68	r69	r70	r71	r72	r73	r74	r75	r76	r77	r78	r79	r80	r81	r82	r83	r84	r85	r86	r87	r88	r89	r90	r91	r92	r93	r94	r95	r96	r97	r98	r99	r100
------	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------

(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 r1 and	F2 is r2 and	F3 is r3 and	F4 is r4 and	F5 is r5 and	F6 is r6 and	F7 is r7 and	F8 is r8 and	F9 is r9 and	F10 is r10 and
F11 is r11 and	F12 is r12 and	F13 is r13 and	F14 is r14 and	F15 is r15 and	F16 is r16 and	F17 is r17 and	F18 is r18 and	F19 is r19 and	F20 is r20 and
F21 is r21 and	F22 is r22 and	F23 is r23 and	F24 is r24 and	F25 is r25 and	F26 is r26 and	F27 is r27 and	F28 is r28 and	F29 is r29 and	F30 is r30 and
F31 is r31 and	F32 is r32 and	F33 is r33 and	F34 is r34 and	F35 is r35 and	F36 is r36 and	F37 is r37 and	F38 is r38 and	F39 is r39 and	F40 is r40 and
F41 is r41 and	F42 is r42 and	F43 is r43 and	F44 is r44 and	F45 is r45 and	F46 is r46 and	F47 is r47 and	F48 is r48 and	F49 is r49 and	F50 is r50 and
F51 is r51 and	F52 is r52 and	F53 is r53 and	F54 is r54 and	F55 is r55 and	F56 is r56 and	F57 is r57 and	F58 is r58 and	F59 is r59 and	F60 is r60 and
F61 is r61 and	F62 is r62 and	F63 is r63 and	F64 is r64 and	F65 is r65 and	F66 is r66 and	F67 is r67 and	F68 is r68 and	F69 is r69 and	F70 is r70 and
F71 is r71 and	F72 is r72 and	F73 is r73 and	F74 is r74 and	F75 is r75 and	F76 is r76 and	F77 is r77 and	F78 is r78 and	F79 is r79 and	F80 is r80 and
F81 is r81 and	F82 is r82 and	F83 is r83 and	F84 is r84 and	F85 is r85 and	F86 is r86 and	F87 is r87 and	F88 is r88 and	F89 is r89 and	F90 is r90 and
F91 is r91 and	F92 is r92 and	F93 is r93 and	F94 is r94 and	F95 is r95 and	F96 is r96 and	F97 is r97 and	F98 is r98 and	F99 is r99 and	F100 is r100 and
THEN									
it is digit 0									

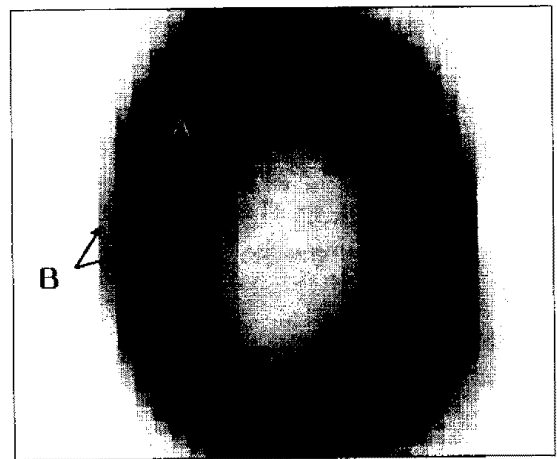
(그림 12) 'IF~THEN~'으로 표현한 숫자 0의 퍼지 규칙  
(Fig. 12) Fuzzy rules of digit 0, illustrated in a sentence. 'IF~THEN~'

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

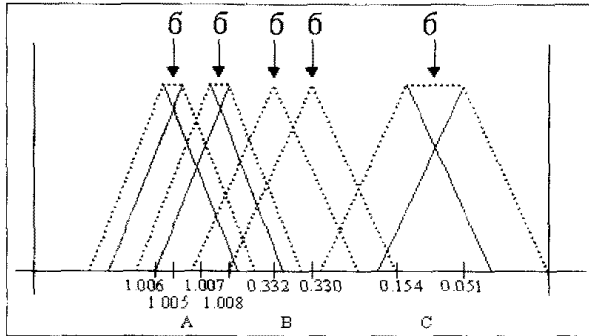
M개의 벡터값을 t개의 단계로 임계값을 적용한다면 임계값은  $\delta(t+1) = \delta(t) + \alpha$ 로 결정된다. 여기서  $\alpha$ 는 초기 임계값을 나타내는 상수로서  $0 < \alpha$ 인 일정한 값을 갖는다.

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

값이 낮아 자가 임계값(6)보다 작으면 비슷한 영역을 합하여 규칙을 생성하고, 그렇지 않으면 새로운 규칙을 부여한다. 이 처리를 연속된 이웃영역에 처음부터 끝까지 적응 임계값으로 비교 처리하여 규칙을 만든다. 따라서 많은 퍼지 영역이 하나의 삼각형 대신 사다리꼴 모양을 가진다.

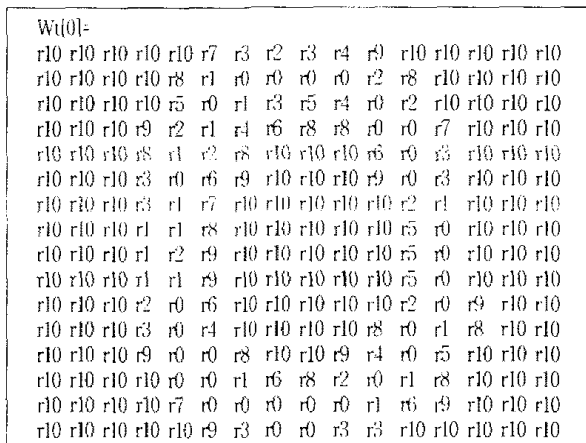


(그림 14) 적응 임계값에 의한 퍼지 규칙 생성의 예  
(Fig. 14) The generation of Fuzzy rules with adaptive threshold value

이를 도식화 한 것이 (그림 14)로서, A 영역은 임계값을 0.001로, B 영역은 임계값을 0.002, C영역은 0.003으로 적용하여 규칙을 생성한 것이다.

숫자 0에 임계값을 0.005로 일정하게 적용한 결과 퍼지 규칙이 30개이다. 숫자 가운데 부분은 명암도의 가중값이 낮아 분류에 비교적 영향을 끼치지 않는 부분들이지만 규칙을 같게 하지 않고 따로 부여함으로써 규칙 수를 많게 하였다. 결과는 (그림 11)과 같다.

숫자 0에 대한 임계값을 일정하게 적용하지 않고



(그림 15) 적응 임계값을 적용했을 경우의 퍼지 규칙  
(Fig. 15) Fuzzy rules with adaptive threshold value

자기 임계값을 0.0005에서 8단계로 단계 증가하는 적응 임계값을 적용한 결과 퍼지 규칙은 11개다. 명암도의 가중값이 높은 부분, 즉 분류에 비교적 많은 영향을 끼치는 부분은 일정한 임계값을 적용했을 때와 마찬가지로 규칙이 세분화되었고, 명암도의 가중값이 낮은 가운데 부분과 바깥 부분은 분류에 비교적 많은 영향을 끼치지 않아 규칙을 통합하여 규칙의 수를 줄였다. 결과는 (그림 15)와 같다.

#### 4. 결과 및 분석

본 논문의 실험은 펜티엄프로 200MHz 윈도우95환경에서 Delphi 4.0 언어로 프로그래밍 하였으며, 숫자 0에서부터 9까지 16x16 양질의 이미지 데이터 각각 10개씩 100개를 초기 참조 패턴으로 사용하여 학습률 0.01로 1000번 학습하였다. 퍼지 규칙 생성을 하기 위해 LVQ 알고리즘을 이용하여 초기 참조 패턴 학습을 통해 입력 데이터의 부류 분포에 근접하도록 픽셀별 명암도에 가중값을 부여하였다. 갱신된 픽셀별 가중값을 정렬하여 근접하는 퍼지 지역에 단계적으로 단계 증가하는 적응 임계값을 적용해 이웃하는 퍼지 지역과 융통성 있게 규칙을 생성함으로써 퍼지 규칙의 수를 줄여나갔다.

결과는 <표 1>과 같이 임계값을 주지 않았을 때 생성된 퍼지 규칙수는 숫자 0~9까지 16x16 데이터의 픽셀별 가중값 2560개중 중복되는 것을 제외한 1256개이고, 임계값을 0.005로 일정하게 적용했을 때 생성된 규칙이 108개로, 임계값을 적용하지 않았을 경우에 비해 1083개의 퍼지 규칙이 줄어들었다.

<표 1> 임계값에 의해 변한 퍼지 규칙 수  
(Table 1) The numbers of Fuzzy rules, with threshold value

원류 방법	임계값을 적용안한것	일정한 임계값 (0.005) 적용(A)	적용 임계값 적용(B)		규칙 수 차이(A-B)	줄어든 륜(%)	
			단계	초기임계값 규칙수			
퍼지 규칙 수	1256	108	8	0.0005	48	60	56%
				0.001	53	55	51%
				0.002	18	90	84%
			16	0.0005	40	68	63%
				0.001	19	89	83%
				0.002	11	97	90%



본 논문에서 제안한 적응 임계값을 적용한 방법에 의한 퍼지 규칙은 단계를 8단계로 하고, 초기임계값을 0.001로 하였을 때 퍼지 규칙이 53개, 0.002일 때 18개, 0.0005일 때 48개이며, 단계를 16단계로 하고, 초기임계값을 0.001일 때 19개, 0.002일 때 11개, 0.0005일 때 40개의 퍼지 규칙이 생성되었다.

적응 임계값을 적용하여 퍼지 규칙을 생성시킬 때 <표 1>에서와 같이 단계와 초기 임계값에 따라서 규칙 수가 다르게 변하는 것을 알 수 있다. 이때 퍼지 규칙을 많이 줄이는 것만이 좋은 것만은 아니다. 퍼지 규칙이 약 30개 이내로 줄어들면 규칙이 올바르게 나오지 않아 성능하락을 가져오는 단점이 있다. 따라서 상태가 가장 좋은 조건은 단계 8, 초기 임계값 0.0005를 적용하여 규칙 수가 48개 생성 될 때이다. 또 일정한 임계값을 적용할 때도 마찬가지로 임계값을 작게 하면 규칙 수가 너무 많아지고, 반대로 임계값을 크게 하면 규칙 수는 줄어들지만 숫자를 분류할 수 없을 정도로 성능이 하락된다.

그래서 규칙 수를 줄이면서 적응임계값을 적용할 때와 일정한 임계값을 적용할 때를 비교해 보기 위해 규칙 수를 45~50개 정도로 비슷하게 만들어 보았다. 먼저 단계 8, 초기 임계값 0.005로 정하여 적응임계값을 적용한 결과 (그림 16)과 같이 48개의 규칙이 생성되었으며, 이때가 최적인 상태였다. 반면 임계값을 0.01로 정하여 일정한 임계값을 적용한 결과 (그림 17)과 같이 45개의 규칙이 생성되었다. 두 방법에 의한 규칙 수는 비슷하였으나, 일정한 임계값을 적용했을 때는 이미 숫자 8을 분류하지 못할 정도로 규칙이 단일화되어 성능이 하락되었다. 따라서 규칙 수를 줄게 만들면 일정한 임계값을 적용할 때가 먼저 성능 하락을 가져오는 결과를 얻었다. 그래서 일정한 임계값을 0.005로 낮추어 적용한 결과 규칙 수가 108개 생성되었으며, 상태가 가장 양호하였다.

(그림 16) 적응 임계값에 의한 숫자 0~9까지 퍼지 규칙

(Fig. 16) Fuzzy rules of digits, 0~9, with adaptive threshold value

(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 적응 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.

제한한 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 교차학습 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 성렬하여 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 입계값을 적용했을 경우는 43개의 작은 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] Siv, 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 시뮬레이티드아닐라의 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)

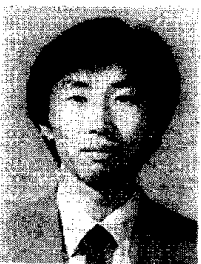
1983년~1984년 현대중공업 사원  
 1989년~현재 안동대학교 전자정보산업학부 부교수  
 관심분야 : 뉴럴 네트워크, 패턴 인식, 컴퓨터 그래픽스



### 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