

# 퍼지 멤버십 함수로 최적화된 LVQ를 이용한 패턴 분류 모델

(Pattern Classification Model using LVQ Optimized by  
Fuzzy Membership Function)

김도현<sup>†</sup> 강민경<sup>\*\*</sup> 차의영<sup>\*\*\*</sup>

(Dotlyeon Kim) (Min-Kyeong Kang) (Eui-Young Cha)

**요약** 패턴인식은 전처리 과정에서 패턴들의 특징을 추출하고 이를 학습을 통하여 유사한 패턴들끼리 클러스터링을 한 다음 식별 과정을 거쳐 인식하게 된다. 본 연구에서는 OCR 시스템에서의 패턴 인식을 위한 패턴 분류 모델로서 퍼지 멤버십 함수를 도입하여 LVQ 학습 알고리즘을 최적화한 F-LVQ(Fuzzy Learning Vector Quantization)를 제안한다. 본 논문의 효율성을 검증하기 위하여 한글 및 영어 22종의 글꼴에 대한 숫자 데이터 220개 패턴을 학습한 후 이를 다양한 형태로 변형시킨 4840개의 테스트 패턴에 대하여, 기존의 여러 가지 패턴 분류 모델과의 비교 분석을 통해 그 유효성과 강인성을 증명하였다.

**키워드** : 퍼지 멤버십, 클러스터링, LVQ

**Abstract** Pattern recognition process is made up of the feature extraction in the pre-processing, the pattern clustering by training and the recognition process. This paper presents the F-LVQ(Fuzzy Learning Vector Quantization) pattern classification model which is optimized by the fuzzy membership function for the OCR(Optical Character Recognition) system. We trained 220 numeric patterns of 22 Hangul and English fonts and tested 4840 patterns whose forms are changed variously. As a result of this experiment, it is proved that the proposed model is more effective and robust than other typical LVQ models.

**Key words** : Fuzzy Membership, Clustering, LVQ

## 1. 서론

패턴 인식은 전처리 및 특징 추출 과정, 학습 과정, 식별 과정으로 구성된다. 전처리 및 특징 추출 과정에서는 입력패턴에 대한 잡음 제거 및 효과적인 특징들을 추출[1-3]하여 입력벡터를 구성함으로써 패턴 학습기의 입력 벡터를 만들어주고, 학습 과정을 통해서 입력 벡터들을 유사한 특징에 따라 클러스터링 함으로써 참조모델(reference model)을 생성하며, 식별 과정에서는 입력

패턴을 참조모델내의 각 참조패턴과 비교하여 입력패턴과 가장 유사한 참조패턴에 속하는 부류로 입력패턴을 분류함으로써 패턴 인식이 이루어지게 된다.

유사한 특징을 가지는 입력패턴을 클러스터링 하는 방법에는 통계적인 방법으로 Hierarchical Clustering[4], K-means Clustering 등이 있으며, 신경망을 이용하는 방법으로 코넨 자기조직화 지도 네트워크(Kohonen SOFM : Self-Organizing Feature Map Network)[5-6], LVQ(Learning Vector Quantization)[6] 등의 방법이 사용되고 있다. 특히 LVQ는 지도학습(supervised learning)에 근거한 패턴인식기의 일종으로서 Bayes 결정 경계면을 근사화하여 높은 분류능력을 가지며, 오류 역전과 알고리즘에 비해 학습속도가 빨라 학습패턴 수가

<sup>†</sup> 학생회원 : 부산대학교 전자계산학과  
unlimit@hanmail.net

<sup>\*\*</sup> 학생회원 : 부산디지털정보기술대학  
dragon@harmony.cs.pusan.ac.kr

<sup>\*\*\*</sup> 종신회원 : 부산대학교 전자계산학과 교수  
eycha@harmony.cs.pusan.ac.kr

논문접수 : 2001년 10월 12일

심사완료 : 2002년 5월 28일

1) 여기서 참조패턴이란 입력패턴 집합이 클러스터링되었을 때 생성된 각 클러스터의 중심 패턴을 말하며, 따라서 참조패턴 벡터는 클러스터의 중심 벡터를 말한다.

많은 경우에 적합한 패턴 인식기로 널리 사용되고 있다 [7-14]. 하지만 클러스터의 초기 중심 패턴이 입력패턴과 멀리 떨어진 값으로 설정되거나(outside of convex hull)나 초기 학습을 및 참조패턴의 설정, 참조패턴 벡터의 갱신 방법, 갱신할 패턴 선정에 있어 개선할 여지가 다분하다. 이에 본 논문에서는 이러한 점들을 개선하는 방법으로 입력패턴 집합을 대표하는 패턴들에 대해 베즈텍의 퍼지 멤버십 등식[15-16]을 적용한 Fuzzy-LVQ를 이용하여 입력패턴의 변형에 보다 강인하고 정확한 학습을 할 수 있도록 함으로써 안정적인고 신뢰성있는 패턴 분류 모델을 제안한다.

2장에서는 관련 연구로 초기 참조패턴의 설정 방법 및 몇 가지 클러스터링 방법에 대하여 설명하고, 3장에서는 제안하는 패턴인식 모델로서 F-LVQ(Fuzzy Learning Vector Quantization)를 설명한다. 그리고 4장에서 다양한 변형을 가지는 입력패턴을 통한 실험 및 결과를 분석하고, 향후 연구 방향 및 결론을 5장에서 맺는다.

## 2. 관련 연구

### 2.1 초기 참조패턴 벡터의 설정

초기 참조패턴 벡터를 설정하는 방법은 여러 가지가 있는데 일반적으로 임의의 값에 의한 초기화, 입력패턴 샘플에 의한 초기화, 입력패턴 부류의 평균값에 의한 초기화, 입력패턴 부류의 군집화에 의한 초기화 등을 이용한 방법이 있다.

#### 2.1.1 임의의 값에 의한 초기화

0과 1사이의 난수값으로 패턴벡터를 초기화하는 것으로 구현이 간단하여 가장 일반적으로 사용되는 방법이지만 패턴 벡터의 차원이 커질 경우 입력패턴과 너무 멀리 떨어져서(outside of convex hull) 초기화 될 경우가 생겨 학습이 효율적으로 이루어지지 않을 수 있는 문제점이 있다.

#### 2.1.2 입력패턴 샘플에 의한 초기화

위와 같은 문제점을 방지하기 위하여 임의의 샘플 입력패턴 자체를 초기 참조패턴으로 설정하는 방법으로 각 부류별로 선택된 임의의 N개의 샘플 입력패턴을 참조패턴으로 설정하는 방법이다.

#### 2.1.3 입력패턴 부류의 평균값에 의한 초기화

입력패턴을 각 부류별로 평균한 평균벡터로 각 참조패턴을 초기화하는 방법으로써 각 입력패턴 부류의 공통적인 특성을 반영하는 반면 다양한 특성을 간과할 수 있는 단점이 있다.

#### 2.1.4 입력 패턴 부류의 군집화를 통한 초기화

입력 패턴을 각 부류별로 나누어 K-means 알고리즘

등을 통하여 군집한 다음 이 군집들의 중심을 각각 참조패턴 벡터로 설정하는 방법이다. 따라서 각 부류별로 어느 정도 군집화된 몇 개의 참조패턴들로서 전체의 참조패턴이 형성된다.

## 2.2 통계적 클러스터링

### 2.2.1 K-means Clustering

K-means 알고리즘에 의한 클러스터링은 임의의 c개의 패턴으로 초기 클러스터 중심 패턴을 설정한 후 입력패턴과 클러스터의 중심패턴과의 거리가 최소가 되는 클러스터에 입력패턴을 멤버로 포함시키면서 k개의 멤버들의 평균으로 클러스터의 중심값을 조정하면서 입력패턴을 분류하는 방법으로 그 알고리즘은 표 1과 같다.

표 1 K-means Clustering 알고리즘

[단계1]	입력패턴 집합 $X = \{x_1, x_2, \dots, x_N\}$ , 참조패턴 집합 $V = \{v_1, v_2, \dots, v_c\}$ , $N$ : 입력패턴 수, $c$ : 클러스터 수, $T$ : 총 반복회수, $\theta$ : 허용오차값 설정
[단계2]	$V_0 = \{v_{1,0}, v_{2,0}, \dots, v_{c,0}\}$ : 0과 1사이의 난수로 초기 참조패턴 벡터 설정
[단계3]	For $t=1, 2, 3, \dots, T$ For $k=1, 2, 3, \dots, N$ a. 승자 참조패턴 선택 $x_k - v_{i,t} = \min(x_k - v_{j,t}), (1 \leq j \leq c)$ b. 승자 참조패턴 벡터 수정 $v_{i,t+1} = \frac{1}{N_j} \sum_{x_j \in S_{i,t}} x_j$ 여기서, $S_j(t)$ 는 시간 t에서 참조패턴 $v_{i,t}$ 에 속하는 패턴들의 집합, $N_j$ 는 $S_j(t)$ 의 크기 c. next k
[단계4]	반복에 따른 오차 계산 $E_t = V_t - V_{t-1} = \sum \sum v_{k,t} - v_{k,t-1}$
[단계5]	만약, $E_t \leq \theta$ 이면 학습 중단 아니면 학습률 조정, next t

### 2.2.2 계층적 클러스터링(Hierarchical Clustering)

계층적 클러스터링은 N개의 입력패턴이 있을 때 이를 유사한 것끼리 서로 병합(merge)시켜서 원하는 클러스터 개수가 될 때까지 반복하는 방법이다. 여기서 두 개의 클러스터를 병합할 때 클러스터의 참조패턴 벡터의 갱신은 K-means 알고리즘과 같이 각 클러스터 멤버 벡터들의 평균값을 이용한다. 이와 같은 반복 과정을 거치게 되면 입력패턴 집합에서 유사한 패턴들은 병합되고 그렇지 않은 패턴들은 남게 되므로써 다양한 입력패턴을 대표할 수 있는 참조패턴 집합으로 클러스터링 되는 장점이 있으나 패턴들간의 거리 측정 및 병합에 있어 시간이 많이 걸리는 단점이 있다. 표 2에서는 이와

같은 일련의 과정을 보여준다.

표 2 Hierarchical Clustering 알고리즘

[단계1]	입력패턴 집합 $X=\{x_1, x_2, \dots, x_N\}$ , 참조패턴 집합 $V=\{v_1, v_2, \dots, v_N\}$ , $N$ : 입력패턴 수
[단계2]	참조패턴 집합을 입력패턴 집합으로 할당 참조패턴 개수: $n = N$
[단계3]	a. 가장 유사한 참조패턴 쌍 $v_i, v_j$ 선택 b. 두 참조패턴을 $v_q$ 로 병합 $v_q = v_i \cup v_j$ 라 정의할 때, $v_q = \frac{1}{N_i + N_j} \left( \sum_{x \in S_i} x_i + \sum_{x \in S_j} x_j \right)$ 여기서, $S_i(t), S_j(t)$ 는 참조패턴 $v_i, v_j$ 에 속하는 패턴들의 집합, $N_i, N_j$ 는 각각 $S_i(t), S_j(t)$ 의 크기 c. $n = n - 1$
[단계4]	참조패턴 개수 $n$ 이 원하는 참조패턴 개수가 되면 학습 중단, 아니면 단계 3으로 이동

2.3 LVQ(Learning Vector Quantization)를 이용한 클러스터링

LVQ는 입력패턴  $X$ 를  $c$ 개의 출력층으로 클러스터링 하는 네트워크로 LVQ 알고리즘은 K-means 알고리즘 및 Kohonen의 SOFM과 유사하지만 지도학습 유형의 군집화 방법으로써, 학습 방법에 있어서 입력패턴과 참조패턴에 대하여 군집의 유사성 여부를 결정하여 그에 맞게 참조패턴을 갱신한다는 점이 다르다.

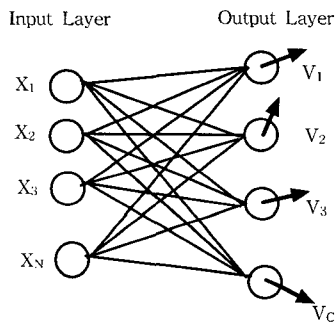


그림 1 LVQ Network

LVQ의 구조는 그림 1에서 보는 바와 같이 입력층에는  $X=\{x_1, x_2, \dots, x_N\}$ 으로 구성된 패턴벡터가 입력되며,  $c$ 개의 출력층은 참조패턴  $V=\{v_1, v_2, \dots, v_c\}$ 로 구성되고 여기서 출력층에 연결된 연결강도  $v_r$ 이 클러스터의 중심값이 된다. 학습은  $x_i$ 와  $v_r$ 의 거리를 계산하여 그 거리가 최소가 되는  $i$ 번째 출력노드가 승자노드가 되며 이 승자 노드에 연결된 참조패턴  $v_i$ 에 대해서만 식

(1)에 있는 학습규칙에 의하여 학습을 한다.

$$v_{i,t+1} = v_{i,t} + \alpha_t (x_k - v_{i,t}) \quad (1)$$

여기서,  $v_{i,t}$ 는 반복  $t$ 시점에서 승자 참조패턴 벡터( $1 \leq i \leq c$ )이고,  $x_k$ 는  $k$ 번째 입력패턴 벡터( $1 \leq k \leq N$ )이다. 식 (1)에 의한 학습을 할 때 참조패턴의 초기값  $v_{r,0}$ 은 0과 1사이의 임의의 난수로 결정하고, 초기 학습률  $\alpha$ 는 0과 1사이의 값으로 설정하되 일반적으로 식 (2)와 같이 반복이 계속되면서 점차 감소되도록 설정한다.

$$\alpha_t = \alpha_0 \left(1 - \frac{t}{T}\right) \quad (2)$$

여기서,  $\alpha_0$ 는 초기 학습률,  $t$ 는 현재 반복회수,  $T$ 는 전체 반복회수를 의미한다.

2.3.1 LVQ1

LVQ 알고리즘은 입력패턴  $x_k$ 와 가장 유사한 참조패턴  $v_p$ 에 대하여, 입력패턴과 같은 부류일 경우 식 (3)와 같이 참조패턴을 입력패턴과 더욱 가깝게 수정하고, 입력패턴과 다른 부류일 경우 식 (4)과 같이 입력패턴과 더욱 멀어지도록 조정한다.

$$v_{p,t+1} = v_{p,t} + \alpha_t (x_k - v_{p,t}) \quad (\alpha_t > 0) \quad (3)$$

$$v_{p,t+1} = v_{p,t} - \alpha_t (x_k - v_{p,t}) \quad (\alpha_t > 0) \quad (4)$$

이때, 학습의 종료는 오차가 설정된 범위 이내이거나 반복이 지정된 회수만큼 수행되면 종료한다. 표 3에 LVQ1 알고리즘이 요약되어 있다.

표 3 LVQ 1 알고리즘

[단계1]	입력패턴 집합 $X=\{x_1, x_2, \dots, x_N\}$ , 참조패턴 집합 $V=\{v_1, v_2, \dots, v_c\}$ , $N$ : 입력패턴 수, $c$ : 클러스터 수, $T$ : 총 반복회수, $\theta$ : 허용오차값 설정
[단계2]	$V_0=\{v_{1,0}, v_{2,0}, \dots, v_{c,0}\}$ : 0과 1사이의 난수로 초기 연결강도값 설정 학습률 $\alpha_0$ : 0과 1사이의 값 또는 지정된 값으로 초기학습률 설정
[단계3]	For $t=1,2,3,\dots,T$ For $k=1,2,3,\dots,N$ a. 승자 참조패턴 선택 $x_k - v_{p,t} = \min\{x_k - v_{i,t}\}, (1 \leq j \leq c)$ b. 승자 참조패턴 수정 $x_k$ 와 $v_{p,t}$ 가 같은 부류일 때, $v_{p,t+1} = v_{p,t} + \alpha_t (x_k - v_{p,t})$ $x_k$ 와 $v_{p,t}$ 가 다른 부류일 때, $v_{p,t+1} = v_{p,t} - \alpha_t (x_k - v_{p,t})$ d. next k
[단계4]	반복에 따른 오차 계산 $E_t = V_t - V_{t-1} = \sum \sum v_{k,t} - v_{k,t-1}$
[단계5]	만약, $E_t \leq \theta$ 이면 학습 중단 아니면 학습률 조정, next t

2.3.2 LVQ2

LVQ2 알고리즘은 학습시 입력패턴  $x_k$ 에 근접한 두 개의 참조패턴  $v_i$ 와  $v_j$ 를 동시에 갱신한다는 점을 제외하면 LVQ1 알고리즘과 동일하다. 선택된 두 개의 참조패턴 중 하나는 입력패턴과 같은 부류에 속하고 다른 하나는 다른 부류에 속해야 하며, 입력패턴  $x$ 는 일정한 윈도우 영역내 위치해야 한다. 이 윈도우는 선택된 참조패턴  $v_i$ 와  $v_j$ 의 중앙 평면상에 정의되는 영역으로,  $d_i$ 를 입력패턴  $x_k$ 와 참조패턴  $v_i$ 간의 거리로,  $d_j$ 를 입력패턴  $x_k$ 와 참조패턴  $v_j$ 간의 거리로 정의할 때, 다음의 식 (5)을 만족할 때 입력패턴  $x_k$ 는  $w$  크기의 윈도우 영역내에 위치하게 된다.

$$\min\left(\frac{d_i}{d_j}, \frac{d_j}{d_i}\right) > \frac{(1-w)}{(1+w)} \quad (5)$$

따라서 LVQ2는 두 개의 참조패턴이 입력패턴과의 유사성 비교에 따라 동시에 갱신되며 이는 식 (6),(7)와 같은 방법으로 참조패턴  $v_i$ 와  $v_j$ 가 갱신된다.

$x_k$ 와 같은 부류에 속하는 참조패턴  $v_{i,t}$ 에 대하여

$$v_{i,t+1} = v_{i,t} + \alpha_t (x_k - v_{i,t}) \quad (6)$$

$x_k$ 와 다른 부류에 속하는 참조패턴  $v_{j,t}$ 에 대하여

$$v_{j,t+1} = v_{j,t} - \alpha_t (x_k - v_{j,t}) \quad (7)$$

2.3.3 LVQ3,4

LVQ2 알고리즘은 두 부류 분포간의 경계를 기준으로 입력패턴  $x_k$ 를 참조하여 결정 경계를 서로 다르게 이동시키는 개념을 바탕으로 하고 있으나 혼란 과정이 진행됨에 따라  $v_i$  및  $v_j$ 가 그들의 부류 분포에 계속 근접하는 것을 보장할 필요가 있다. 따라서 LVQ3,4 알고리즘은 각각 입력패턴  $x_k$ 와 서로 같은 부류에 속하는 참조패턴  $v_i$  및  $v_j$ 에 대하여 식 (8)을, 서로 다른 부류에 속하는 참조패턴  $v_i$  및  $v_j$ 에 대하여 식 (9)을 추가하여 개선하였고, 여기서  $\epsilon_t$ 는 추가학습률로써 윈도우 크기에 비례하도록 설정한다.

$x_k$ 와 같은 부류에 속하는 참조패턴  $v_{i,t}$ 에 대하여

$$v_{i,t+1} = v_{i,t} + \epsilon_t \alpha_t (x_k - v_{i,t}) \quad (8)$$

$x_k$ 와 다른 부류에 속하는 참조패턴  $v_{j,t}$ 에 대하여

$$v_{j,t+1} = v_{j,t} - \epsilon_t \alpha_t (x_k - v_{j,t}) \quad (9)$$

2.3.4 GLVQ (General LVQ)

LVQ 알고리즘은 초기 참조패턴 벡터  $v_{i,0}$ 에 의해서 클러스터링 결과가 많은 영향을 받기 때문에 Pal et. al.은 학습을 승자 참조패턴 뿐만 아니라 승자가 아닌 참조패턴도 같이 학습을 시키는 GLVQ 클러스터링 알고리즘을 제안하였다. GLVQ 네트워크에서의 학습 규칙은 입력패턴  $x_k$ 와 참조패턴간의 거리에 대한 가중값을 준

표 4 GLVQ 알고리즘

[단계1]	입력패턴 집합 $X = \{x_1, x_2, \dots, x_n\}$ , 참조패턴 집합 $V = \{v_1, v_2, \dots, v_c\}$ , $c$ : 클러스터 수, $T$ : 총 반복회수, $\theta$ : 허용오차값 설정
[단계2]	$V_0 = \{v_{1,0}, v_{2,0}, \dots, v_{c,0}\}$ : 0과 1사이의 난수로 초기 연결강도값 설정 학습률 $\alpha_0$ : 0과 1사이의 값 또는 지정된 값으로 초기학습률 설정
[단계3]	For $t=1,2,3,\dots,T$ For $k=1,2,3,\dots,N$ a. 승자 참조패턴 발견 $x_k - v_{i,t-1} = \min\{x_k - v_{j,t-1}\} \quad (1 \leq j \leq c)$ b. 승자 참조패턴 수정 $v_{i,t} = v_{i,t-1} + \alpha \frac{(x_k - v_{i,t-1})}{D^2 - D + \ x_k - v_{i,t-1}\ ^2}$ c. 비승자 참조패턴 수정 $v_{j,t} = v_{j,t-1} - \alpha \frac{(x_k - v_{j,t-1})}{D^2}$ d. next k
[단계4]	반복에 따른 오차 계산 $E_t = V_t - V_{t-1} = \sum v_{k,t} - v_{k,t-1}$
[단계5]	만약, $E_t \leq \theta$ 이면 학습 중단 아니면 학습률 조정, next t

손실함수(Loss function)  $L_x$ 를 식 (10)와 같이 정의하고, 이 손실함수를 최소화시키는 방법으로 학습규칙을 유도하였다.

$$L_x = \sum_{i=1}^c g_{iv} \|x - v_i\|^2 \quad (10)$$

여기서, 승자 참조패턴일 경우,  $g_{iv} = 1$

승자 참조패턴이 아닐 경우,  $g_{iv} = \frac{1}{\sum_{i=1}^c \|x - v_i\|^2}$

로써,  $i$ 번째 패턴에 대한 가중값을 나타낸다.

이때 학습은 승자 참조패턴일 경우에는  $L_x$ 를  $v_i$ 로 미분하고, 승자 참조패턴이 아닐 경우에는  $v_j$ 로 미분하여 각각 식 (11),(12)에 의하여 모든 참조패턴에 대한 학습을 수행한다.

$$v_{i,t} = v_{i,t-1} + \alpha \frac{(x_k - v_{i,t-1})}{D^2 - D + \|x_k - v_{i,t-1}\|^2} \quad (11)$$

$$v_{j,t} = v_{j,t-1} - \alpha \frac{(x_k - v_{j,t-1})}{D^2} \quad (12)$$

여기서,  $D = \sum_{i=1}^c \|x - v_i\|^2, 1 \leq k \leq N, 1 \leq r \leq c$  이다.

GLVQ의 특징으로는 승자 참조패턴  $v_i$  뿐만 아니라 승자가 아닌 다른 참조패턴까지도 조정을 해주기 때문에 반복이 계속되면서  $v_i$ 가 아닌 다른 참조패턴들도 승자가 될 수 있고, 반복이 완료된 후의 모든 참조패턴 벡

터는 입력된 패턴을 각 클러스터별로 포함하는 중심값으로 설정될 수 있으며 초기 참조패턴의 값이나 학습률 등에 많은 영향을 받지 않는 장점이 있다.

### 3. F-LVQ(Fuzzy LVQ)

#### 3.1 퍼지 멤버십값에 의한 참조패턴 선정

본 논문에서는 기존의 LVQ 학습 알고리즘에서 학습할 참조패턴에 대한 선정시 윈도우 영역을 이용하는 방법을 보다 최적화하기 위하여 퍼지 멤버십 함수를 적용하였다. 이 윈도우 영역은 입력패턴과 부류가 다른 두 참조패턴 사이의 경계면을 나타내는 것으로, LVQ2.3 알고리즘은 이 경계면에 대한 근사화에 바탕을 두고 있다. 이때 이 윈도우 영역은 식 (5)의 조건에 의해 결정되는 영역으로서 이 영역에 위치하는 입력패턴은 부류가 다른 두 참조패턴  $v_i$  와  $v_j$ 와 유사한 패턴임을 의미하며 따라서 오분류의 원인이 되는 후보패턴임을 의미한다.

$$\min\left(\frac{d_i}{d_j}, \frac{d_j}{d_i}\right) > \frac{(1-w)}{(1+w)} \quad (5)$$

여기서  $d_i$ 를 입력패턴  $x$ 와 참조패턴  $v_i$ 간의 거리로,  $d_j$ 를 입력패턴  $x$ 와 참조패턴  $v_j$ 간의 거리라 하고 참조패턴  $v_i$ 가 입력패턴과 같은 부류이며, 참조패턴  $v_j$ 가 입력패턴과 다른 부류라고 정의할 때, 이들 거리에 대한 비  $d_i / d_j$ ,  $d_j / d_i$ 는 입력  $x$ 에 대하여 두 참조패턴  $v_i$  와  $v_j$ 와의 거리에 대해서만 고려된 것으로 다른 참조패턴에 대한 거리는 반영되지 않게 됨으로써 참조패턴  $v_i$  및  $v_j$ 의 조정에 의하여 다른 참조패턴이 영향을 받을 수 있는 가능성이 매우 커지게 된다. 따라서 이런 문제를 해결하기 위하여 식 (13)과 같이 퍼지 멤버십 함수를 적용하여 입력패턴과 각각의 참조패턴과의 유사성을 거리에 대한 상대적인 멤버십 값으로 적용함으로써 다른 참조패턴과의 관계를 상대적으로 고려하여 이들의 경계면을 보다 신뢰성있게 결정할 수 있도록 하였다.

$$\min\left(\frac{U_i}{U_j}, \frac{U_j}{U_i}\right) > \frac{(1-w)}{(1+w)} \quad (13)$$

#### 3.2 퍼지 멤버십 함수

퍼지 이론은 1965년 자데(Zadeh)에 의해 처음으로 도입되었다. 퍼지이론은 데이터 구조에 대한 보다 정확한 정보를 줄 수 있다. 0이나 1 중에 어느 하나만을 선택하는 이분법과는 달리 상대적인 값들을 가질 수 있게 해준다. 퍼지  $c$ -means 알고리즘은 던(Dunn)의 퍼지  $c$ -means 클러스터링 알고리즘에 근거하여 베즈택(Bezda)에 의해 개발되었으며 수렴성(convergence) 및 최적성(optimalty) 테스트 및 클러스터 타당성(validity)도 입증되었다. 폐구간  $[0,1]$ 에서의 퍼지 멤버십 함수는 식

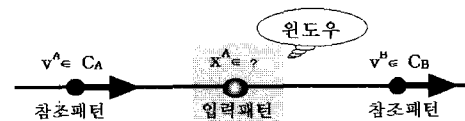
(14)의 베즈택의 식에 의해서 결정된다.

$$U_{ik} = \frac{1}{\sum_{l=0}^{c-1} \left(\frac{d_{lk}}{d_{ik}}\right)^2} \quad (14)$$

여기서,  $c$ 는 참조패턴의 개수이고  $U_{ik}$ 는 입력패턴  $x_k$ 의 참조패턴  $v_i$ 에 대한 퍼지 멤버십 값이며  $U_{ik}$ 의 총합계는 항상 1이 된다. 따라서,  $U_{ik}$ 는 현재 참조패턴  $v_i$ 과 입력패턴  $x_k$ 와의 유사도를 나타내며 이것은 다른 모든 참조패턴  $v_j$  ( $j \neq i$ )와 입력패턴과의 거리에 대한 상대적인 값이 된다. 즉, 이것은 입력패턴과 참조패턴과의 유사도를 거리만으로 표현하지 않고, 거리에 대한 상대적인 관계로 모호하게(fuzzy)하게 표현한 것으로서, 입력패턴이 어떤 참조패턴과 충분히 가까운 거리에 있다 하더라도 주변의 다른 참조패턴 또한 충분히 가깝다면 이때의 유사도는 거리에 의한 유사도보다 떨어지게 되는 성질을 가지고 있다.

#### 3.3 LVQ 알고리즘의 개선

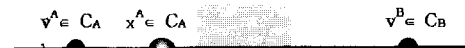
LVQ 학습 알고리즘에서는 입력패턴이 두 참조패턴과의 거리를 바탕으로 형성된 윈도우 영역에 존재할 때 이 두 참조패턴을 갱신하게 된다. 그림 2의 (a)와 같이 A 부류(class)의 입력 패턴( $x^A \in C_A$ )이 윈도우 영역 내 위치하는 경우는 부류(class)가 다른 두 개의 참조패턴  $v^A, v^B$  사이에 입력패턴이 존재하는 경우로써 오분류의 위험이 있는 패턴으로 간주되어 참조 패턴  $v^A, v^B$ 의 수정이 요구된다. 따라서 입력패턴과 같은 부류의 참조패턴( $v^A$ )은 입력패턴과 더 가까이 수정되고, 입력패턴과 다른 부류의 참조패턴( $v^B$ )은 입력패턴과 더 멀리 수정되도록 조정하는 방법을 사용하였다. 한편, (b)의 경우는 입력패턴이 윈도우 영역 바깥에 위치하는 경우로



(a) 윈도우 영역 내 입력패턴  $x^A$



(b) 참조패턴  $v^A$ 와 가까운 윈도우 영역 외 입력패턴  $x^A$



(c) 참조패턴  $v^B$ 와 가까운 윈도우 영역 외 입력패턴  $x^A$

그림 2 LVQ에서의 입력패턴 유형

써 같은 부류의 참조패턴과 가까이 위치하여 참조 패턴의 갱신이 필요 없는 경우이다. 그런데, (c)의 경우는 입력패턴이 윈도우 영역 바깥에 위치하며 다른 부류의 참조패턴과 가까이 위치함으로써 참조 패턴의 갱신이 이루어져야 함에도 불구하고 입력패턴이 윈도우내 존재하지 않기 때문에 식 (5)의 조건에 의해 수정이 되지 않는 현상이 발생하게 된다.

따라서 이러한 오류를 방지하기 위하여 입력패턴이 윈도우 영역 바깥에 위치하더라도 입력패턴과 다른 부류의 참조패턴이 입력패턴과 같은 부류의 참조패턴보다 입력패턴과 더 가까울 경우 이를 수정하도록 추가하여 다음의 식 (15)를 만족할 때 참조패턴을 각각 수정하도록 한다.

표 5 F-LVQ 알고리즘

[단계1] 입력패턴 집합  $X=(x_1, x_2, \dots, x_N)$ ,  
 참조패턴 집합  $V=(v_1, v_2, \dots, v_c)$   
 $N$  : 입력패턴 수,  $c$  : 클러스터 수,  
 $T$  : 총 반복회수,  $\theta$  : 허용오차값 설정

[단계2]  $V_0=(v_{1,0}, v_{2,0}, \dots, v_{c,0})$  : 0과 1사이의 난수로 초기 연결강도값 설정  
 학습률  $\alpha_0$  : 0과 1사이의 값 또는 지정된 값으로 초기 학습률 설정

[단계3] For  $t=1,2,3,\dots,T$   
 For  $k=1,2,3,\dots,N$   
 a. 입력패턴과 같은 부류의 참조패턴  $v_i$  선택  
 $x_k - v_{i,t} = \min\{x_k - v_{p,t}\}, (1 \leq p \leq c)$   
 b. 입력패턴과 다른 부류의 참조패턴  $v_j$  선택  
 $x_k - v_{j,t} = \min\{x_k - v_{p,t}\}, (1 \leq p \leq c)$   
 c. 참조패턴 갱신  
 if  

$$\min\left(\frac{U_{ik}}{U_{jk}}, \frac{U_{jk}}{U_{ik}}\right) > \frac{(1-w)}{(1+w)} \text{ or } d_{ik} > d_{jk}$$
 then  

$$v_{i,t+1} = v_{i,t} + \alpha_t (x_k - v_{i,t})$$

$$v_{j,t+1} = v_{j,t} - \alpha_t (x_k - v_{j,t})$$
 else  

$$v_{i,t+1} = v_{i,t} + U_{ik,t} \cdot \alpha_t (x_k - v_{i,t})$$

$$v_{j,t+1} = v_{j,t} - U_{jk,t} \cdot \alpha_t (x_k - v_{j,t})$$

$$U_{ik,t} = U_{ik} \left(1 - \frac{t}{T}\right),$$

$$U_{jk} = \frac{1}{\sum_{l=0}^t \left(\frac{d_{jk}}{d_{ik}}\right)^2}$$
 d. next k

[단계4] 반복에 따른 오차 계산  
 $E_t = V_t - V_{t-1} = \sum \sum v_{k,t} - v_{k,t-1}$

[단계5] 만약,  $E_t \leq \theta$  이면 학습 중단  
 아니면 학습률 조정, next t

$$\min\left(\frac{U_i}{U_j}, \frac{U_j}{U_i}\right) > \frac{(1-w)}{(1+w)} \text{ or } d_i > d_j \quad (15)$$

LVQ3.4 알고리즘에서는 훈련 과정이 진행됨에 따라  $v_i$  및  $v_j$  가 그들의 부류 분포에 계속 근접하는 것을 보장하기 위하여 식 (8),(9)와 같이 윈도우 크기에 비례한 추가 학습률( $\epsilon$ )을 사용하였다.

$x_k$ 와 같은 부류에 속하는 참조패턴  $v_{i,t}$ 에 대하여

$$v_{i,t+1} = v_{i,t} + \epsilon_t \cdot \alpha_t (x_k - v_{i,t}) \quad (8)$$

$x_k$ 와 다른 부류에 속하는 참조패턴  $v_{j,t}$ 에 대하여

$$v_{j,t+1} = v_{j,t} - \epsilon_t \cdot \alpha_t (x_k - v_{j,t}) \quad (9)$$

본 논문에서는 입력패턴과 참조패턴과의 거리에 대하여 퍼지 멤버십 함수를 적용한 것과 마찬가지로 학습률  $\epsilon$ 에 대해서도 식 (16),(17)과 같이 퍼지멤버십 값  $U_i$ 를 시간에 따라 감소하도록 하여 적용하였다.

$x_k$ 와 같은 부류에 속하는 참조패턴  $v_{i,t}$ 에 대하여

$$v_{i,t+1} = v_{i,t} + U_{ik,t} \cdot \alpha_t (x_k - v_{i,t}) \quad (16)$$

$x_k$ 와 다른 부류에 속하는 참조패턴  $v_{j,t}$ 에 대하여

$$v_{j,t+1} = v_{j,t} - U_{jk,t} \cdot \alpha_t (x_k - v_{j,t}) \quad (17)$$

여기서,  $U_{ik,t}$ 는  $t$ 회 반복시점에서 입력패턴  $x_k$ 에 대한 참조패턴  $v_i$ 의 퍼지 멤버십값으로 식 (18)과 같이 정의된다.

$$U_{ik,t} = U_{ik} \left(1 - \frac{t}{T}\right) \quad (18)$$

#### 4. 실험 및 결과 분석

##### 4.1 실험 데이터 구성

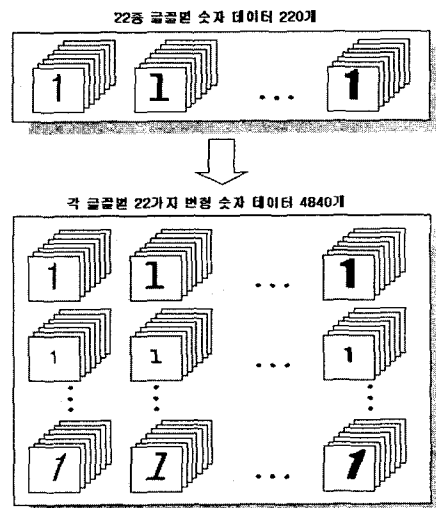


그림 3 숫자 데이터 변형

본 논문에 사용된 실험 데이터는 22개의 2한글 영문 조합 글꼴로 총 220개의 Gray-level 숫자 영상 파일을 기본데이터로 하였으며, 입력 데이터의 다양한 입력 변형을 포함할 수 있도록 기본 데이터에 대하여 다양하게 3) 변형한 총 4840개의 변형 데이터에 대한 인식 실험을 하였다. 또한 다양한 종류의 글꼴에 대한 신뢰성있고 강한 OCR 시스템의 구현을 위하여 학습은 단지 기본 220개의 데이터로만 수행하고, 인식 테스트는 변형된 4840개의 데이터로 실험하였다.

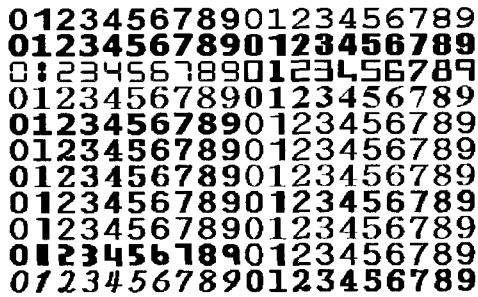


그림 4 학습 숫자 데이터 22종 220개

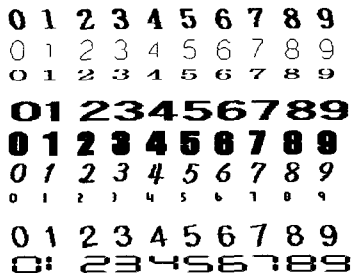


그림 5 인식 테스트 숫자 데이터 예

4.2 획 정규화를 이용한 패턴 벡터 생성

숫자 이미지 데이터로 입력패턴을 생성하기 위한 전처리 과정으로서 다양한 크기의 입력 이미지에 대한 정규화는 문자 영상에 대한 효율적인 패턴벡터를 생성하는데 중요한 역할을 한다. 본 논문에서는 이와 같은 정규화 방법으로써 획 정규화 방법[3]을 사용하였다.

- 2) 굴림, 궁서, 돌음, 명조, 바탕, 산세리프, 샘물, 시스템, 신명조, 안상수, 중고딕, 필기, 휴먼앳체, Arial, Courier, Fixdays, Impact, LED, Orbit-B, Symbol, Times New Roman, Verdana
- 3) 위치이동(4), 경사왜곡(4), 세선화(1), 중간값 필터링(1), 좌우 5°, 10° 회전(4), 모폴로지 확장, 가로확대축소(2), 세로확대축소(2), 가로세로확대축소(2) : 22종 \* 12(21가지 변형+원형) \* 10 = 4840개

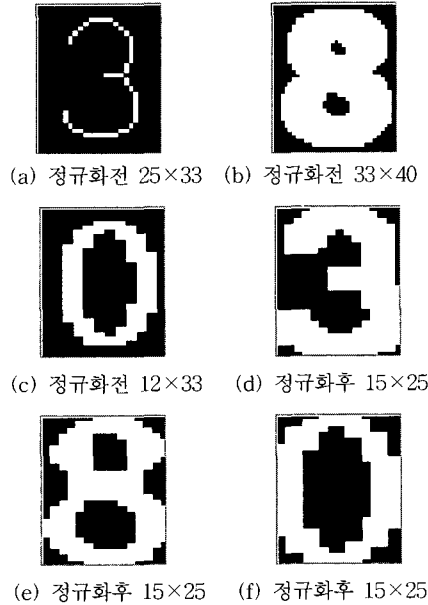


그림 6 다양한 크기의 문자 획 정규화

획 정규화 방법은 문자 이미지의 외접사각 영역을 설정하고 이 영역에 대하여 평균 광강도에 따라 적응적으로 모폴로지(Morphology) Dilation 또는 Erosion 연산을 수행하여 문자 획의 굵기를 일정하게 유지하기 위한 전처리 방법이다. 문자 이미지는 15x25로 정규화하였으며 0~255의 픽셀값을 0~1 사이의 부동소수점값으로 Scaling하여 패턴 벡터를 생성하였다.

4.3 실험 방법

다양한 환경에서의 실험을 위하여 LVQ1, LVQ2, LVQ3, GLVQ, 4)F-LVQ2, F-LVQ3 각각에 대해 참조패턴 초기화 방법(최소 N개의 샘플에 의한 초기화, K-means/Hierarchical Clustering에 의한 초기화), 참조패턴 개수 (10개,30개,50개), 학습률  $\alpha$  (0.2,0.4,0.6), 학습률  $\epsilon$  (0.1,0.3,0.5,0.7,0.9), 윈도우 영역 변수  $w$  (0.1,0.3, 0.5,0.7,0.9) 등을 변화시키면서 총 5)594회의 실험을 수행하였다. 여기서, 초기 참조패턴의 설정방법은 참조패턴 개수만큼의 입력패턴 샘플을 이용한 초기화와, 각 부류별 군집화를 통한 초기화로 K-means 및 Hierarchical Clustering에 의한 초기화 방법을 사용하였으며, 참조패

- 4) F-LVQ2는 퍼지 멤버십 함수를 이용하여 LVQ2를 개선한 방법이고, F-LVQ3는 LVQ3를 개선한 방법이다.
- 5) 594회 = 초기화방법(3) \* 참조패턴수(3) \* {GLVQ(3: 학습률  $\alpha$ ) + LVQ1(3: 학습률  $\alpha$ ) + LVQ2, LVQ3, F-LVQ2, F-LVQ3(60 : 4 \* 학습률  $\alpha$  (3) \* 윈도우  $w$ (5))}

턴 개수의 설정은 숫자 부류 0~9에 대하여 각각 1개, 3개, 5개씩의 참조패턴을 할당하여 10개, 30개, 50개로 설정하였다. 그리고 학습률  $\alpha$ 는 0.2에서 0.6까지, 윈도우 크기 변수  $w$ 는 0.1에서 0.9까지 0.2씩 증가하도록 하였으며 학습률  $\epsilon$ 의 값은 윈도우 영역에 반비례하도록  $\epsilon = 1.0 - w$ 로 하여 설정하였다. 또한 학습은 최대반복회수 200회를 초과하거나 모든 참조패턴 벡터의 값이 이전 반복시점과 변화가 없을 때 종료하도록 설정하였다.

4.4 실험 결과 및 분석

그림 7, 8은 각각 참조패턴 개수별, 초기화 방법별 평균 오인식 데이터 개수를 비교하고 있다. 참조패턴 개수 및 초기화 방법에 상관없이 전체적으로 F-LVQ2, F-LVQ3가 매우 우수한 성능을 보이고 있으며 LVQ1은 참조패턴 개수가 각 부류별로 1개씩 할당되어 있을 경우 좋지 않은 결과를 나타냈다.

그림 9, 10에서 참조패턴의 개수, 초기화 방법, 학습률 등의 변화에 대하여 클러스터링 성능의 안정성을 살펴보기 위하여 오인식 데이터 개수에 대한 표준 편차를 구하여 살펴본 결과, GLVQ가 다른 방법에 비하여 매우 안정적인 성능을 보였다. 실제로 Nikhil R. Pal은 Anderson의 IRIS 데이터를 가지고 반복회수와 학습률을 변화시켜

가면서 LVQ와 GLVQ로 클러스터링 한 결과를 비교하였는데 GLVQ는 150개의 IRIS 데이터중 에러의 개수가 17개로 일정한 반면에 LVQ인 경우에는 100개에서 17개의 에러를 나타내고 있다[16]. 하지만 본 실험에서와 같이 같은 부류에 대해서도 그 패턴의 유형이 다양한 경우에는 전체적으로 참조패턴을 수정하는 방식인 GLVQ가 인식률 면에서는 부적당한 것으로 판단된다. 한편, LVQ1의 경우에는 GLVQ와 마찬가지로 참조 패턴 개수의 변화에 상관없이 일정한 성능을 보임을 알 수 있으나 초기화 방법이 달라짐에 따라 인식 성능의 변화가 매우 심함을 알 수 있으며 이는 불안정한 성능을 가짐을 증명한다.

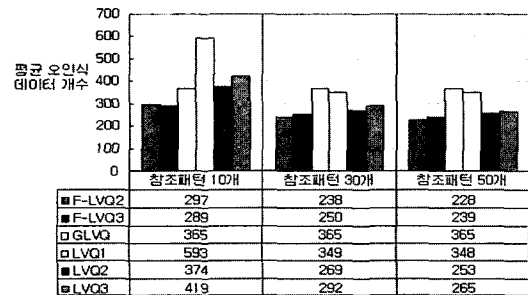


그림 7 참조패턴 개수별 평균 오인식 데이터 개수

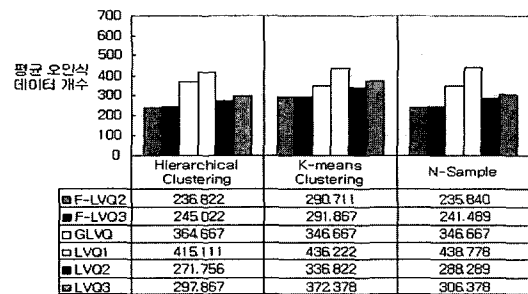


그림 8 초기화 방법별 평균 오인식 데이터 개수

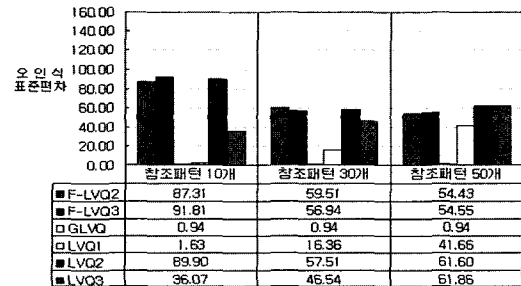


그림 9 참조패턴 개수별 오인식 개수 표준편차

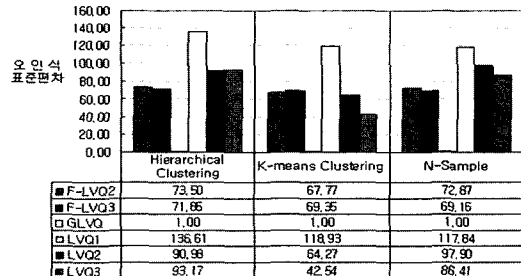


그림 10 초기화 방법별 오인식 개수 표준편차

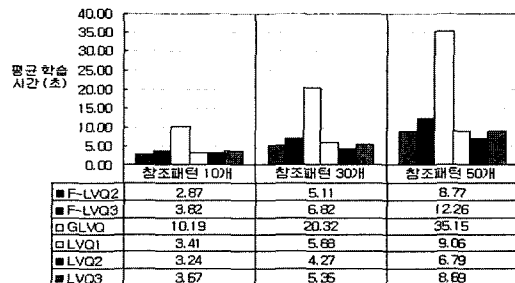


그림 11 참조패턴 개수별 평균 학습 시간



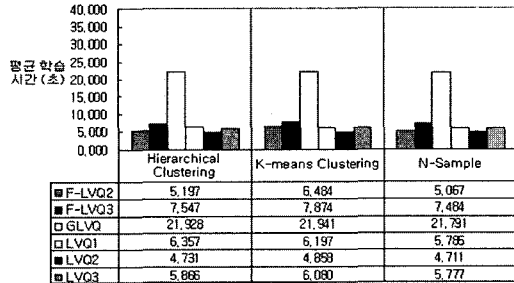


그림 12 초기화 방법별 평균 학습 시간

학습에 소요되는 시간은 일반적으로 알고리즘의 복잡도에 좌우되는데, 그림 11, 12에서 살펴볼 수 있는 바와 같이 GLVQ는 모든 참조패턴을 수정하기 때문에 상당히 많은 학습시간을 요구하고 있다. LVQ1은 학습 알고리즘이 단순함에도 불구하고 LVQ2,3 및 F-LVQ2보다 평균적으로 시간이 많이 걸리는 것은 다른 알고리즘에 비해 빨리 수렴하지 못하고 최대회수의 반복을 수행하였기 때문이다. 마찬가지로 F-LVQ2,3의 방법이 퍼지멤버십 값을 계산하는데 상당한 시간을 소요하지만 LVQ2, LVQ3보다 더 빨리 수렴하는 경우가 많았기 때문에 평균적으로

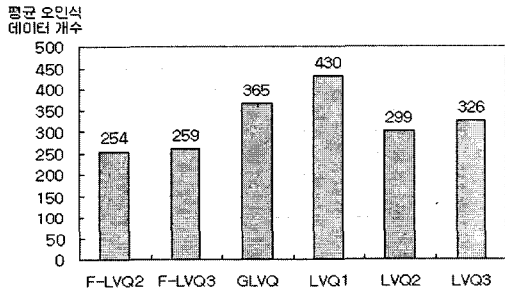


그림 13 전체 평균 오인식 데이터 개수

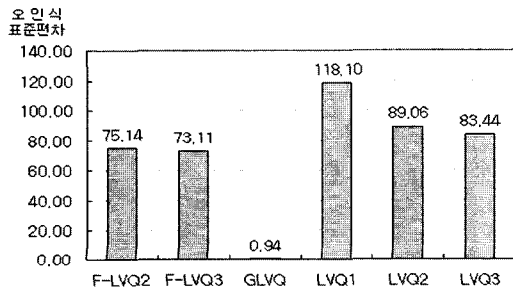


그림 14 전체 오인식 데이터 개수 표준 편차

거의 근소한 차이를 나타내는 것을 알 수 있다.

그림 13~16은 각각 전체 데이터에 대한 평균 오인식 데이터 개수, 오인식 데이터 개수의 표준 편차, 평균 학습 시간 및 평균 인식률을 종합적으로 나타내었다.

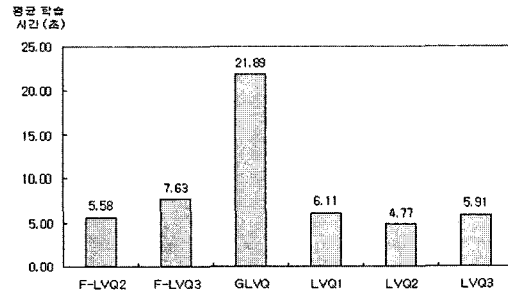


그림 15 전체 평균 학습 시간

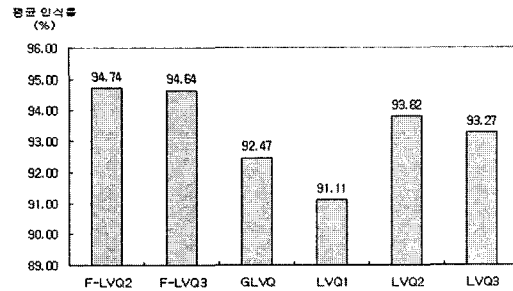


그림 16 전체 평균 인식률

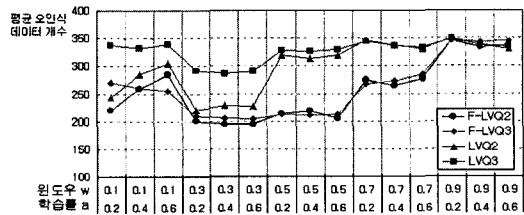


그림 17 학습률 및 윈도우 크기에 대한 오인식 데이터 개수 비교

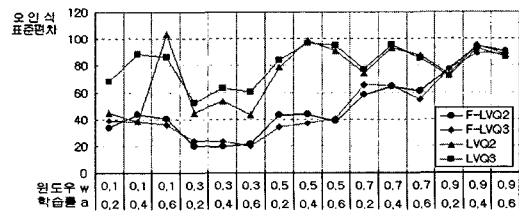


그림 18 학습률 및 윈도우 크기에 대한 오인식 표준 편차 비교

그림 17, 18에서는 성능이 대체적으로 좋은 LVQ2, LVQ3, F-LVQ2, F-LVQ3에 대한 오인식 데이터 개수와 표준편차를 비교하고 있다. 그림에서도 알 수 있듯이 윈도우 크기 변수  $w$ 가 0.9일 때는 모든 알고리즘이 오인식 데이터가 상대적으로 많았으며 표준편차도 크게 나타나서 분별력을 갖지 못했다. 그 이외에는 F-LVQ 알고리즘이 학습률 및 윈도우 크기 변수에 덜 민감하게 반응하여 완만하게 변하고 있으며 오인식 데이터 개수 또한 작음을 알 수 있다. 위의 분석 결과를 종합해 때 F-LVQ 알고리즘은 학습률  $\alpha$ 에는 크게 영향을 받지 않으며 윈도우 크기 변수  $w$ 가 0.3~0.5사이에서 가장 안정적이고 정확한 분류 성능을 보였다.

표 6은 종합적인 실험의 분석 결과로써 F-LVQ의 방법을 비교하고 있다. 제안하는 방법은 평균 인식률에 있어서 약 95%로 가장 우수하였으며, 오인식 데이터를 살펴볼 때 총 4840개의 테스트 데이터에 대하여 평균적으로 40~170개 정도의 데이터를 다른 알고리즘보다 올바르게 분류·인식하는 것을 알 수 있다. 또한 오인식 데이터에 대한 표준편차도 GLVQ를 제외한 다른 알고리즘에 비하여 작으며 이것은 학습률 등의 변화에 덜 민감한 강인한 성능을 나타낸다. 평균 학습 시간은 모든 참조패턴을 수정하는 GLVQ가 가장 많은 시간을 소요하였으며, F-LVQ3 알고리즘은 GLVQ를 제외한 LVQ 알고리즘에 비하여 다소 많이 걸리는 것으로 나타났다.

### 5. 결론 및 향후 연구 방향

본 논문은 LVQ를 이용한 패턴 분류 모델 설계에 있어서 퍼지 멤버십 함수를 적용함으로써 기존의 알고리즘에 비하여 신뢰성있고 안정적인 최적화된 분류 모델인 F-LVQ를 제안하였다. 총 22개 글꼴을 바탕으로 한 220개의 기본 숫자 데이터를 바탕으로 학습을 시키고 이를 다양하게 변형한 4840개의 변형 데이터에 대하여 분류 인식 테스트를 수행한 결과 다른 클러스터링 알고리즘에 비하여 제안한 알고리즘이 인식률 및 안정성에서 우수한 성능을 보였다. 하지만 퍼지 멤버십 값을 계산함에 있어서 계

산량이 증가함으로써 학습 수행 속도가 다소 늦어지는 점이 있었으며 이를 개선하는 방법으로 퍼지 멤버십 값을 계산하기 위한 Look-up 테이블을 효율적으로 구현하기 위한 연구가 필요하며, 추후 숫자 데이터 뿐만 아니라 영어 및 한글에 대한 검증도 수행되어야 할 것으로 생각된다.

### 참고 문헌

- [1] 박창순, 김두영, "오프라인 필기체 숫자 인식을 위한 다양한 특징들의 비교 및 인식을 개선 방안", 한국 정보처리학회 논문지, 제3권 4호, 1996년 7월
- [2] 이성환, 박희선, "고리 투영을 이용한 위치, 크기 및 회전 변형에 무관한 한글 문자 인식", 인지과학 Korean Journal of Cognitive Science, VOL.3, No.1, pp.139~160, 1991년
- [3] 김도현, 강동구, 강민경, 차의영, "문자 인식을 위한 효율적인 획 정규화", 한국 정보처리학회 추계학술대회 상권, 2001년
- [4] Sergios Theodoridis, Konstantinos Koutroumbas, "Pattern Recognition," ACADEMIC PRESS, 1999
- [5] 김대수, "신경망 이론과 응용 (I), (II)", 하이테크정보, 1994
- [6] Teuvo Kohonen, "The Self-Organizing Map," Proceeding of the IEEE, Vol.78, No.9, pp.1464~1480, Sep. 1990
- [7] 송희현, 이성환, "개선된 LVQ3와 시뮬레이티드 어닐링의 결합을 통한 대용량 필기체 문자 인식을 위한 최적의 참조모델 설계", 한국 정보과학회 논문지, 제 21권 6호, pp. 1059~1067, 1994년 6월
- [8] 김대국, 이상희, 김백섭, "변형된 CNN에 기초한 LVQ2의 초기 참조벡터 결정 방법", 정보과학회 논문지, 제 21권 9호, pp.1675~1682, 1994년 9월
- [9] 김백섭, "최근접 이웃망에 의한 참조벡터 학습", 전자공학회 논문지, 제 31권 7호, pp.170~178, 1994년 7월
- [10] 민준영, 조형기, "Generalized Clustering Network를 이용한 전방향 학습 알고리즘", 한국정보처리학회 논문지, 제 2권 5호, pp.619~625, 1995년 9월
- [11] 전종원, 민준영, "GLVQ 클러스터링을 위한 필기체 숫자의 효율적인 특징 추출 방법", 한국정보처리학회 논문지, 제 2권 6호, pp.995~1001, 1995년 11월
- [12] Jing Wu, Hong Yan, "Combined SOM and LVQ

표 6 실험 결과 종합 분석

학 습 알고리즘	평균정인식률 (%)	평균 정인식 데이터 개수	평균오인식률 (%)	평균 오인식 데이터 개수	평균 학습 시간 (초)	오 인 식 표준편차
GLVQ	92.45	4475	7.55	365	21.89	0.94
LVQ1	91.11	4410	8.89	430	6.11	118.10
LVQ2	93.82	4541	6.18	299	4.77	89.06
LVQ3	93.27	4514	6.73	326	5.91	83.44
F-LVQ2	94.75	4586	5.25	254	5.58	75.14
F-LVQ3	94.65	4581	5.35	259	7.63	73.11

Based Classifiers for Handwritten Digit Recognition," Proceeding of the IEEE on Neural Networks, Vol.6, pp.3074~3077, 1995

[13] Kitajima, N. "A New Method for Initializing Reference Vectors in LVQ," Proceeding of the IEEE on Neural Networks, Vol.5, pp.2275~2279, 1995

[14] Seong-Whan Lee, Jong-Soo Kim, "Multi-lingual, multi-font and multi-size large-set character recognition using self-organizing neural network," Proceedings of the Third International Conference on Document Analysis and Recognition, Vol.1, pp.28~33, 1995

[15] Dae Su Kim, "A Design of the Fuzzy Neural Network Image Recognizer," Journal of Fuzzy Logic and Intelligent Systems, Vol.2, No.3, pp.50~57, 1992

[16] Nikhil R. Pal, James C. Bezdek, "Generalized Clustering Networks and Kohonen's Self-Organizing Scheme," IEEE Transaction on Neural Networks, Vol.4, No.4, pp.549~557, July 1993



김도현

2001년 부산대학교 전자계산학과 졸업(학사). 2001년 ~ 현재 부산대학교 대학원 전자계산학과 석사 재학중. 관심분야는 신경회로망, 문자/패턴인식, 영상처리, 컴퓨터비전



강민경

2000년 인제대학교 전자계산학과 졸업(학사). 2002년 부산대학교 대학원 멀티미디어학과 졸업(공학석사). 2002년 현재 부산디지털정보기능대학 컴퓨터 강사. 관심분야는 신경회로망, 패턴분석, 컴퓨터비전



차의영

1979년 경북대학교 전자공학과 졸업(학사). 1982년 서울대학교 대학원 전자계산학과(이학석사). 1998년 서울대학교 대학원 컴퓨터공학과(공학박사). 1981년 ~ 1985년 한국전자기술연구소 연구원. 1995년 ~ 1996년 University of London 방문교수. 1985년 ~ 현재 부산대학교 전자계산학과 교수. 관심분야는 컴퓨터비전, 신경회로망 이론, 웨이블릿, 워터마킹