Generalized Clustering Network를 이용한 전방향 학습 알고리즘

민 준 영† 조 형 기††

요 약

본 연구에서는 역전파(backpropagation)학습 알고리즘에 대체될 수 있는 전방향 학습 알고리즘에 준하는 혼합 인식모형을 구성한다. 본 알고리즘은 Nikhil R. Pal(1993)이 제안한 GLVQ(Generalized Learning Vector Quantization)를 이용하여 패턴을 클러스터링을 한 다음 비유사성(dissimilarity)을 가진 패턴끼리 재구성(regrouping)하여 단순 퍼셉트론(simple perceptron)을 이용하여 group별 학습을 한다. 일반적으로 역전파학습인 경우 학습시간이 많이 소요된다는 단점이 있다[1]. 본 알고리즘의 특징으로는 feed-forward학습이기 때문에 학습시간이 단축될 뿐만 아니라 전체 패턴을 그룹별로 나누어 학습을 하기 때문에 인식률도 향상 시킬 수 있다. 본 알고리즘에 적용한 데이타는 250개의 ASCII코드를 16×8격자에 정규화시킨 비트패턴(bit pattern)을 이용하였다. 실험결과 250개의 패턴을 10개의 클러스터로 나누어 학습을 시켰을 때 각 클러스터별 평균반복횟수 94.7회만에 250개의 ASCII코드를 100% 인식할 수 있었다.

Feed-forward Learning Algorithm by Generalized Clustering Network

Min Joon Young† and Cho Hyung Ki††

ABSTRACT

This paper constructs a feed-forward learning complex algorithm which replaced by the backpropagation learning. This algorithm first attempts to organize the pattern vectors into clusters by Generalized Learning Vector Quantization(GLVQ) clustering algorithm(Nikhil R. Pal et al, 1993), second, regroup the pattern vectors belonging to different clusters, and the last, recognize into regrouping pattern vectors by single layer perceptron. Because this algorithm is feed-forward learning algorithm, total learning time is less than backpropagation algorithm and the recognition rate is increased. We use 250 ASCII code bit patterns that is normalized to 16×8. As experimental results, when 250 patterns devide by 10 clusters, average iteration of each cluster is 94.7, and recognition rate is 100%.

1. 서 론

신경망은 1957년 Frank Rosenblatt에 의하여 처음 제안된 이후 역전파 학습 알고리즘(back-propagation learning algorithm)등의 교사학습(supervised learning)과 Kohonen network, SONN등 비교사 학습(unsupervised learning)방법으로 크게 구분할 수 있다[2, 3]. 그러나 비교사학습이나 교사학습 모두 인식(recognition) 또는 분류(classification)를 하는데 있어서는 그 나름대로의 단점을 갖고 있다. 교사학습 중에서 단순 퍼셉트론(simple perceptron)인 경우에는 인식할 수 있는 문자의 종류가 극히 제한되어 있고, 역전파 학습 알고리즘인 경우에는 학습 도중 지역최소점(local minimum)이 될 경우에는 연결강도(weight)의 변화가 없어서 패턴을 더 이상 인식을 못하고 학습이 종료될 뿐만 아니라, 또한 은닉층에 대한 역할이 모호하기 때문에 패턴을 인식하는데 소요되는 시간이 매우 길다는

† 정 회 원 : 상지대학교병설전문대학 전자계산과 교수.
†† 정 회 원 : 아주대학교 산업공학과 박사과정, 상명여대 강사.
논문접수 : 1995년 1월 23일, 심사완료 : 1995년 10월 6일.

단점을 갖고 있다[4, 9]. 비교사 학습은 교사학습이 현실적으로 목표패턴을 구하기 난해하기 때문에 입력패턴들의 통계적 특징을 파악하여 비슷한 입력패턴들은 비슷한 출력패턴을 생성할 수 있도록 연결강도를 조정하는 방식이다. 그러나 초기 연결강도 벡터의 설정이나 연결강도조정 범위에 따라 전혀 다른 출력값을 얻게 되는 경우가 있다.

본 논문은 실제 패턴을 인식하는 데 학습하는 데 소요되는 시간이 매우 길고, 또한 학습과정에 있어서 지역최소점이나 정상점(stationary point)이 발생되기 때문에 인식률(recognition rate)이 낮아진다는 문제점을 갖는 기존의 역전파 학습 알고리즘에 의한 패턴인식에 대체되는 알고리즘을 제안한다. 실제로 역전파 알고리즘인 경우 XOR를 인식하는 데 최소 2,000회 이상의 반복이 이루어 져야지만 인식이 된다. 본 알고리즘은 비교사학습을 통하여 패턴을 구분하고 각 구분된 패턴을 교사학습으로 인식하는 방법으로 전체 패턴이 입력되었을 때 조정되지 못하는 연결강도를 본 논문에서는 우선 각각의 패턴을 Nikhel R. Pal et al.이 제안한 Generalized Learning Vector Quantization(GLVQ)클러스터링 알고리즘[1]을 이용하여 각각의 패턴을 유사성(similarity)이 있는 패턴끼리 클러스터링을 한 다음 각각의 클러스터에 있는 패턴을 하나씩 추출하여 비유사성(dissimilarity)을 갖는 패턴으로 재구성(regrouping)하고, 그 패턴들을 단순 퍼셉트론 알고리즘을 이용하여 인식시키는 알고리즘을 제안한다. 본 논문에서는 전처리과정(preprocessing)을 거쳐 16×8격자에 정규화(normalization)시킨 ASCII 코드 250문자를 대상으로 실험을 하였다.

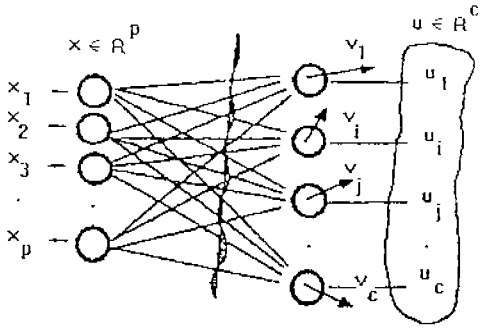
2. Clustering Networks

2.1 Learning Vector Quantization(LVQ)

클러스터링은 패턴의 유사성을 측정하여 유사한 패턴끼리 몇개의 그룹으로 조직화하는 것으로 이러한 문제에 접근하는 이론으로는 Kohonen, Bezdek, Duda와 Hart, Tou와 Gonzalez, Hartigan, Jain과 Dubes에 의하여 제안되어 왔다[1]. 최근에 발표된 것으로 Kohonen이 제안한 LVQ(Learning Vector Quantization)알고리즘이 있다.

실제로 LVQ 그 자체는 클러스터링 알고리즘이 아니라 데이타의 프로토타입을 설계하기 위해서 데이타 중에서 가장 가까운 프로토타입(nearest prototype, NP)끼리 묶어서 c개의 부분(c-partitions)으로 만드는 알고리즘이다. 데이타의 집합을 $X=\{x_1, x_2, \cdots, x_n\}\subset R^p$하고, 이 데이타를 c개의 partition으로 그룹을 진다고 했을 때 프로토타입 $V=(v_1, \cdots, v_c)\in R^p$는 각 클러스터의 중앙값(cluster center)가 되며, LVQ알고리즘은 학습을 통해서 $\{v_i\}$를 찾는 것이다.

LVQ알고리즘의 구성은 (그림 1)과 같다[1, 5].



(그림 1) LVQ 알고리즘 네트워크
(Fig. 1) LVQ Algorithm Network

LVQ 클러스터링 알고리즘[1, 5, 6, 7, 8]

[단계 1] 입력패턴 벡터 $X=\{x_1, x_2, \cdots, x_n\}\subset R^p$
c, T, $\varepsilon>0$를 결정.
여기서, c : 클러스터의 수.
T : 최대 반복횟수.
ε : $\| v_t - v_{t-1} \|$ 의 허용오차.

[단계 2] 초기 연결강도값 결정 $V_0=v_{1,0}, \cdots, v_{c,0})\in R^p$
초기 학습률 결정 $\alpha_0\in(1, 0)$.

[단계 3] For t=1, 2, … T ;
For k=1, 2, …, n ;
a. 승자노드(winner node)의 발견

$$\| x_k - v_{i,\,t-1} \| = \min_{1\le j\le c} \{ \| x_k - v_{j,\,t-1} \| \}$$

b. 승자노드의 연결강도 수정.

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

c. Next k.

d. 행렬 u를 결정

$$u_{LVQ\,i\ k}=\begin{cases}1: \|x_k-v_i\| \le \|x_k-v_j\|, \\ \quad 1\le j\le c,\ j\ne i \\ 0: otherwise.\end{cases}$$

$1\le i\le c$ and $1\le k\le n$.

$U_{LVQ}=[u_{LVQ\,i\,k}]$: $c\times n$ 행렬로써 x_k 패턴이 클러스터 c에 속해 있는 상태를 나타내는 행렬.

[단계 4] 에러 계산.

$$E_t=\sum_{k=1}^{n}\sum_{r=1}^{c}|v_{rk,\ t}-v_{rk,\ t-1}|.$$

[단계 5] if $E_t\le\varepsilon$ then STOP ; Else 학습률 수정 α_t ; Next t.

$\alpha_t=\alpha_0(1-t/T)$

그러나 LVQ알고리즘의 경우 모든 경우에 항상 만족하는 클러스터링을 못한다는 단점을 갖고 있다. 즉, 클러스터의 초기 중앙값 $v_{i,0}$가 입력 데이타와 거리(여기서의 거리는 Euclidean 거리를 의미한다)가 멀리 떨어져서 주어질 경우(outside of convex hull)에 학습이 완료된 후의 결과가 만족할 만한 클러스터링이 안된다. 그 이유는 첫째, 초기 중앙값 설정에 문제가 있을 경우와, 둘째, 학습과정에 있어서 승자노드(winner node)의 연결강도만을 수정(update)해 주기 때문이다.

따라서 Nikhil R. Pal은 Generalized Learning Vector Quantization(GLVQ, 1993) 알고리즘을 제안하였다[1].

2.2 Generalized Learning Vector Quantization(GLVQ)

LVQ알고리즘에서는 초기 연결강도 $v_{i,0}$에 의해서 클러스터링 결과가 많은 영향을 받기 때문에 Nikhel R. Pal et al.은 학습을 승자노드 뿐만 아니라 승자가 아닌 노드도 같이 학습을 시키는 Generalized Learning Vector Quantization (GLVQ)클러스터링 알고리즘을 제안하였다.

GLVQ는 손실함수(loss function) L_x (식 1)를 최소화 시키는 학습방법을 적용하였다[1].

$$L_x=\sum_{r=1}^{c} g_{ir}\|x-v_r\|^2$$

여기서, $$g_{ir}=\begin{cases}1 & , \text{if } r=i \\ \dfrac{1}{\sum_{j=1}^{c}\|x-v_j\|^2} & , \text{if } r\ne i\end{cases} \quad (식\ 1)$$

여기서 승자노드일 경우에는 L_x를 v_i(winner class center)로 미분하고, 승자노드가 아닌 경우에는 v_j(non-winner class center)로 미분하여 (식 2)와 (식 3)로써 학습을 한다.

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

$$v_{r,\ t}=v_{r\ t-1}+\alpha_t(x_k-v_{r\ t-1})\ \frac{\|x_k-v_{i,\ t-1}\|^2}{D^2}\ \text{if } r\ne i \quad (식\ 3)$$

여기서, $D=\sum_{r=1}^{c}\|x_k-v_r\|^2$, k=1, 2, …, n ;

r=1, 2, …, c이고,

α_t는 time t에서의 학습률(learning rate)이다.

각 단계별 GLVQ 클러스터링 알고리즘은 다음과 같다.

GLVQ 클러스터링 알고리즘.

[단계 1] 패턴벡터 $X=\{x_1, x_2, \cdots, x_n\}\subset R^p$

c, T, $\varepsilon>0$ 결정.

여기서, c : 클러스터의 수.

T : 최대 반복횟수.

ε : $\|v_t-v_{t-1}\|$의 허용오차.

[단계 2] 초기 연결강도 결정.

$V_0=(v_{1,0}, \cdots, v_{c,0})\in R^p$

초기 학습률 결정. $\alpha_0\in(1, 0)$.

[단계 3] For t=1, 2, … T ;

반복횟수에 따라 학습률 계산. $\alpha_t=\alpha_0(1-t/T)$

For k=1, 2, …, n ;

a. 승자노드의 발견

$$\| x_k - v_{i,\ t-1} \| = \min_{1 \le j \le c} \{ \| x_k - v_{j,\ t-1} \| \}$$

b. 승자노드 및 승자가 아닌 노드의 연결강도 수정

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

승자노드인 경우

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

승자노드가 아닌 경우

Next k.

[단계 4] 에러 계산, $E_t = \sum_{k=1}^{n} \sum_{r=1}^{c} | v_{rk,\ t} - v_{ri,\ t-1} |$.

[단계 5] if $E_t \le \varepsilon$ then STOP ; Else Next t.

$$u_{Li,Q,\ k} = \begin{cases} 1 : \| x_k - v_i \| \le \| x_k - v_j \|, & 1 \le j \le c,\ j \neq i \\ 0 : \textit{otherwise.} & 1 \le i \le c \ \textit{ and } \ \le k \le n. \end{cases}$$

$U_{Li,Q}[u_{Li,Qk}]$: $c \times n$ 행렬로써 x_k 패턴이 클러스터 c에 포함되어 있는 상태를 나타내는 행렬.

실제로 Nikhil R. Pal은 Anderson의 IRIS데이타를 가지고 반복횟수와 학습률을 변화시켜 가면서 LVQ와 GLVQ로 클러스터링 한 결과를 비교하였는데 GLVQ는 150개의 IRIS데이타중 에러의 갯수가 17개로 일정한 반면에 LVQ인 경우에는 100개에서 17개의 에러를 나타내고 있다[1].

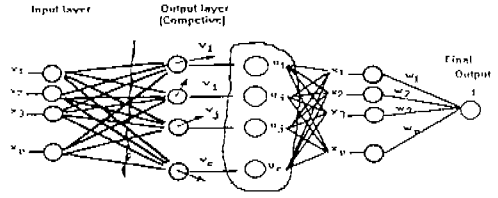
3. Feed-forward학습 알고리즘

3.1 학습알고리즘

본 알고리즘은 패턴 $X = \{x_1, x_2, \cdots, x_n\} \subset R^p$를 GLVQ 클러스터링 알고리즘을 이용하여 c-partition으로 클러스터링을 하면 각 클러스터에는 $\{n_1, n_2, \cdots, n_c\}$, $\sum_{i=1}^{c} n_i = n$개의 패턴이 분포하게 된다. 여기서 각 클러스터에 있는 패턴끼리는 유사성이 있게 되는데 유사성이 있는 패턴은 단순퍼셉트론으로 학습을 할 경우 인식률이 매우 저조하게 된다. 예로서 본 논문에서 이용한 데이타를 단순퍼셉트론 네트워크에 입력 시켰을 경우 ASCII코드값 179에서 245까지의 semigraphic코드 66개 패턴을 인식할 경우에 43%의 인식률에 불과하다. 따라서 GLVQ 클러스터링 알고리즘에 의하여 구분된 각 패턴을 유사성이 없는 패턴끼리 재구성을 해야 한다.

본 알고리즘에서는 각 클러스터에서 i번째 $1 \le i \le n_{max}$(n_{max} : maximum number of patterns in cluster)패턴들을 추출하여 c개씩 n_{max}개로 재구성하였고, 재구성한 n개의 패턴을 입력층의 노드 p개와 출력층의 노드 1개의 단순퍼셉트론으로 학습을 시킨다.

본 논문에서 제안한 알고리즘의 구성은 (그림 2)와 같다.



(그림 2) GLVQ를 이용한 전방향 학습 알고리즘
(Fig. 2) Feed-forward learning algorithm by GLVQ

본 논문에서 제안한 알고리즘은 다음과 같다.

[단계 1] GLVQ 알고리즘으로 N개의 패턴을 c개의 클러스터로 클러스터링.

[단계 2] c개의 클러스터에 있는 패턴을 하나씩 추출하여 이 패턴끼리 재구성.

For i=1, 2, ···, n_{max}.

Regroup_Pattern$_{i,\ j}$=$x_{j,\ i}$ 여기서, $1 \le j \le c$

$x_{j,\ i}$: j번째 클러스터에 속해있는 i번째 패턴 벡터.

[단계 3] 재구성한 Regroup_Pattern$_{i,\ j}$를 다음의 알고리즘으로 그룹별로 학습.

(단계 3.1) 초기 학습률, 최대 반복횟수, 허용오차 결정 $\alpha \in (1, 0)$, T, $\varepsilon > 0$.

(단계 3.2) For i=1, 2, ···, n_{max}.

초기 연결강도 결정 $W_0 = (w_{n,\ 0}, \cdots, w_{1,\ 0}) \in (1, 0)$

For j=1, 2, ···, c

{j, i번째 패턴 $P_{j,\ i}$, Read}

For=1, 2, ···, T

에러계산 : E_t= | Regroup_Pattern$_{t,\ j}$−

$$\sum_{k=1}^{p} P_{j\ i} w_{ik\ t} |$$

if $E_i \leq \varepsilon$ then Next i;
Else 연결강도 수정.

$$w_{ik\ t} = \alpha(Regroup_Pattern_{ij} - \sum_{k=1}^{p} w_{ik\ t-1} P_{jk}) P_{jk}$$

Next t.

3.2 인식(recognition)알고리즘

학습이 완료된 다음 정규화된 미지(unknown)의 패턴 $x_{unknown}$이 읽혀 졌을 때, 가장 거리가 가까운 클러스터를 결정하고, 그 클러스터에 있는 기존의 패턴과 비교하여 오차가 가장 적은 패턴의 위치 i를 찾는다. 다음 i번째 weight벡터를 입력패턴과 곱했을 때 $\sum_{i=1}^{p} x_{unknown}\ w_i$의 출력값으로 인식할 수 있다.

인식하는 알고리즘은 다음과 같다.

인식 알고리즘

[단계 1] $x_{unknown}$ 패턴이 속해있는 클러스터의 발견.

$$\| x_{unknown} - v_i \| = \min_{1 \leq j \leq c} \{ \| x_{unknown} - v_j \| \}$$

[단계 2] 클러스터 안에 있는 패턴들 중 $x_{unknown}$ 패턴이 있는 위치의 발견.

$$x_i = \min_{1 \leq j \leq n_{max}} \{ \| x_{unknown} - x_{cj} \| \}$$

[단계 3] $x_{unknown}$ 패턴을 인식.

$$f = \sum_{k=1}^{p} x_{unknown} W_i$$

4. 실험및 결과분석

본 알고리즘에 적용한 데이타는 전처리과정(preprocessing)을 통하여 얻은 250개의 ASCII 코드이다. 각각의 데이타는 16×8격자(128bit)로 주어졌고 GLVQ알고리즘을 통하여 각 패턴을 10개의 클러스터로 나누었다. (그림 3)은 ASCII코드 '65'에 해당하는 알파벳 'A'를 정규화한 후의 입력 패턴을 나타낸 것이고, 〈표 1〉은 각각의 클러스터에 포함되는 패턴에 대한 ASCII 코드값을 의미한다.

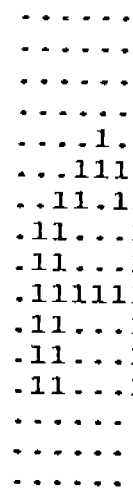
이를 각 클러스터에 속해 있는 패턴을 차례로 하나씩의 패턴을 추출하여 재구성을 하면

$REGROUP_0$={11, 174, 32, 0, 21, 2, 181, 10, 7, 17},
$REGROUP_1$={20, 217, 35, 1, 39, 3, 185, 127, 9, 22},
⋮

$REGROUP_{47}$={234}이 되고, 이 $REGROUP_i$ 집합에 속한 패턴을 독립적으로 구성된 단순퍼셉트론에 입력시켜 학습을 시킨다.

〈표 1〉에서 각 클러스터별로 i($1 \leq i \leq 47$)번째 패턴을 c(=10)개 추출하여 학습률 α=0.01, T=2000으로 학습을 시킨 결과 각 클러스터별 평균반복횟수 94.7회만에 250개의 ASCII코드를 100% 인식할 수 있었다.

본 알고리즘은 학습하는데 있어서 크게 두가지로 구분하였다. 1차적으로 패턴을 클러스터링을 위한 학습을 하였고, 2차적으로 각 패턴을 독립적인 네트워크에 입력시켜 인식할 수 있는 패턴의 수를 분할하였다. 또한 각 패턴의 상호 유사성까지 고려하여 패턴을 분할하였기 때문에 네트워크의 위상(topology)이 간단한 단순퍼셉트론으로도 인식이 가능하였다. 실제로 본 논문에서 이용한 데이타를 128개의 입력노드, 50개 노드



(그림 3) 정규화된 입력패턴(알파벳 'A')
(Fig. 3) Normalized input pattern(alphabet 'A')

를 갖는 은닉층, 8개의 출력노드를 갖는 역전파 학습 알고리즘으로 학습을 시킨 결과 98개의 패턴만을 인식하고, 나머지 패턴에 대해서는 반복 횟수가 증가하더라도 연결강도의 변화가 없게 되어 인식할 수가 없게 나타났다. 물론 이 패턴들을 인식하기 위해서 은닉층을 증가시키거나 또는 은닉층의 노드 수를 조정하여 네트워크를 다시 구성 하여 학습 시킨다면 언젠가는 모든 패턴을 인식할 수는 있겠으나 네트워크의 구성은 이론적으로 정립되어 있지 않기 때문에 여러번의 시행착오를 거쳐야만이 할 수 있는 방법이다.

〈표 1〉 클러스터에 포함된 패턴
(Table 1) Patterns belong to Clusters

Cluster	11 20 126 169 170 218
Cluster 1	174 217
Cluster 2	32 35 37 38 42 44 45 53 56 57 59 60 61 87 89 91 92 93 94 124 153 164 165 221 233 235 237 238 241 243 244 245 247 248
Cluster 3	0 1 12 13 18 19 33 34 36 46 48 49 51 52 54 55 62 64 65 66 67 68 69 70 73 74 75 76 77 78 79 80 81 83 84 85 86 88 96 100 102 105 142 163 225 229 234
Cluster 4	21 39 63 72 95 97 98 99 101 104 107 108 109 110 111 112 113 115 116 117 118 119 120 125 150 167 168 220
Cluster 5	2 3 4 5 6 8 14 15 16 25 26 27 28 29 30 40 43 47 50 58 90 133 143 155 172 173 194 203 222 224 226 231 232 242 246 249
Cluster 6	181 185 199 201 208 212
Cluster 7	10 127 128 129 130 131 132 134 135 136 140 141 144 145 146 147 148 149 151 152 154 156 158 160 161 162 166 219 223 227 228 230
Cluster 8	7 9 175 176 180 183 184 186 187 197 198 200 202 204 206 209 213
Cluster 9	17 22 23 24 31 41 71 82 103 106 114 121 122 123 137 138 139 157 159 171 177 178 179 182 188 139 190 191 192 193 195 196 205 207 210 211 214 21 216 236 239 240

5. 결 론

본 논문에서 제안한 알고리즘은 비교사학습인 GLVQ 클러스터링 알고리즘과 교사학습인 단순 퍼셉트론 알고리즘을 결합한 방법이다. 본 논문에서는 우선 각각의 패턴을 GLVQ 클러스터링 알고리즘을 이용하여 각각의 패턴을 유사성이 있는 패턴끼리 클러스터링을 한 다음 각각의 클러스터에 있는 패턴을 하나씩 추출하여 비유사성을 갖는 패턴으로 재구성하고, 그 패턴들을 단순 퍼셉트론 알고리즘을 이용하여 인식시키는 알고리즘을 제안하였다.

본 알고리즘의 특징은 다음과 같다.

첫째, 일반적으로 교사학습, 특히 역전파 학습 알고리즘인 경우 학습시간이 너무 긴 반면에 본 알고리즘은 학습이 전방향으로 이루어 지기 때문에 학습시간이 짧다.

둘째, 구분지어 놓은 패턴끼리 독립적인 시냅스에 의해서 학습을 하기 때문에 인식률이 높다.

역전파알고리즘는 모든 패턴이 하나의 네트워크에 입력되기 때문에 학습에 필요한 소요시간의 증가와 정상점 또는 지역최소점에서의 학습불능 상태에 따른 인식률 저하를 가져온다. 그러나 본 논문에서는 패턴을 각 클러스터로 분리한 후 이 분리된 패턴을 독립된 네트워크의 입력패턴으로 하여 학습을 시켰기 때문에 입력패턴 수의 감소로 인한 학습시간의 단축과 연결강도의 변화량을 항상 유지할 수 있도록 하여 패턴의 인식률을 높였다.

본 논문은 그 응용 범위를 우선 16×8격자(128 bit)의 ASCII코드에 한정시켰으나 앞으로의 연구에서는 한글인식, 필기체인식 등과 같은 보다 다양한 패턴에 적용이 필요할 것으로 생각된다.

참 고 문 헌

[1] Nikhil R.Pal, James C.Bezdek,Eric C.K. Tsao, "Generalized clustering networks

and Kohonen's Self-Organizing Scheme," *IEEE Trans.on Neural Networks,* Vol.4, No.4, pp.549-557, Jul. 1993.

[2] John Hertz, Anders Krogh, Richard G. Palmer, 'Introduction to the Theory of Neural Computation,' Addison-Wesley Publishing Co., 1991.

[3] Yoh-Han Pao, 'Adaptive Pattern Recognition and Neural Networks',Addison-Wesley Publishing Co., 1989.

[4] Terence D.Sanger, "Optimal Unsupervised Learning in a Single-Layer Linear Feedforward Neural Network," *Neural Networks,* Vol.2, p.459, 1989.

[5] Teuvo Kohonen, 'Self-Organization and Associative Memory,' Berlin, Germany, Springer-Verlag, 3rd Ed., 1989.

[6] Teuvo Kohonen, "The Self-Organizing Map," *Proceeding of the IEEE,* Vol.78, No.9 ,pp.74-82, Sep. 1990.

[7] Helge Ritter, Thomas Martinets, Klaus Schulten, 'Neural Computation and Self-Organizing Maps,' Addison-Wesley Publishing Co., 1992.

[8] 송희헌, 이성환, "개선된 LVQ3와 시뮬레이티드 아닐링의 결합을 통한 대용량 필기체 문자인식을 위한 최적의 참조 모델설계," 정보과학회논문지, Vol.21, No.6, Jun. 1994.

[9] 김대수, '신경망 이론과 응용(Ⅰ)', 하이테크정보, 1992.

민 준 영

1982년 아주대학교 산업공학과 졸업(학사)
1989년 성균관대학교 경영대학원 정보처리학과(석사)
1995년 성균관대학교 대학원 전산통계전공(박사)
1993년～현재 상지대학교병설 전문대학 전자계산과 전임강사
관심분야 : 신경회로망, 패턴인식

조 형 기

1987년 아주대학교 산업공학과 졸업(학사)
1989년 아주대학교 대학원 산업공학과(석사)
1994년～현재 아주대학교 대학원 박사과정
1991년～현재 아주대학교 교통연구소 전임연구원
1993년～현재 상명여자대학교 정보처리학과 강사
관심분야 : 신경회로망, IVHS.