

論文2003-40CI-5-5

SOM을 이용한 LVQ 네트워크 설계

(LVQ Network Design using SOM)

鄭慶權*, 李鎔求**, 嚴基煥*

(Kyung Kwon Jung, Yong Gu Lee, and Ki Hwan Eom)

요약

본 연구에서는 SOM을 이용하여 LVQ 네트워크 설계 방식을 제안한다. 제안한 방식은 SOM을 이용하여 LVQ 네트워크의 서브 클래스를 결정하고, 기준 벡터의 초기값을 설정하는 방식으로 LVQ 네트워크의 분류 성능을 향상시킨다. 제안한 방식으로 설계된 LVQ 네트워크의 유용성을 확인하기 위하여 Fisher의 Iris 데이터와 문자 인식에 적용하여 기존 LVQ 네트워크의 초기 기준 벡터를 설정하는 방식들과 비교, 검토한 결과 우수한 분류 성능을 확인하였다.

Abstract

In this paper, we propose a design method of the LVQ network using the SOM. The proposed method determines subclasses and initial reference vectors of the LVQ network using the SOM. The efficacy of the proposed method is verified by means of simulations on iris data of Fisher and character recognition. The results show that the proposed method improves considerably on the performance of the conventional LVQ network.

Keywords : LVQ, SOM, initial reference vector, subclass

I. 서론

SOM(Self-Organizing Map)의 학습은 거리 측정법에 기반을 둔 방식이다. 일반적으로 거리 측정 방법에는 성능 면에서 우수하고, 정규화가 필요 없는 유클리드(Euclid) 방식을 널리 사용하고 있다. 따라서 SOM은 유사한 패턴 벡터에 대하여 승리 뉴런으로부터 일정한 거리 내에 위치한 벡터들의 부류를 하나의 클러스터로 분류하는 데 우수한 분류 성능을 갖는다.

패턴을 보다 정밀하게 분류하기 위하여 하나의 클러스터 혹은 클래스에 여러 개의 서브 클래스를 둘 수 있다. 따라서 학습을 통하여 서브 클래스들을 분류하고 분류된 서브 클래스들을 하나의 클래스로 지정한다면 패턴 분류의 성능은 보다 정밀하고 우수한 분류 성능을 갖게 된다. 이러한 클래스 속의 서브 클래스를 학습시켜서 클래스로 분류하는 신경회로망이 바로 LVQ(Learning Vector Quantization) 네트워크이다. 따라서 LVQ 네트워크는 SOM을 보다 정교하게 분류하는 네트워크로 고려할 수 있고, 같은 클래스의 패턴 벡터들이 패턴 공간 상에 흩어져서 서브 클래스를 형성하는 경우에 SOM은 패턴을 올바르게 분류할 수 없지만 LVQ 네트워크는 패턴 공간 상에 흩어져서 위치하는 패턴 벡터들을 부분 부분의 서브 클래스로 학습시키고, 각 서브 클래스를 하나의 클래스로 묶어 주는 역할을 하게 된다¹⁾.

* 正會員, 東國大學校 電子工學科

(Dept. of Electronic Eng., Dongguk University)

** 正會員, 翰林情報産業大學 電子情報通信科

(Dept. of Electronic Communication, Hallym College of Electronic Information & Industry)

接受日:2002年12月11日, 수정완료일:2003年8月20日

LVQ 네트워크의 구조는 SOM과 같으나 근본적인 차이는 LVQ 네트워크에서는 SOM에서와 같이 학습 시 이웃 반경을 고려하지 않고, SOM에서 출력 뉴런은 하나의 클러스터(클래스)를 형성하지만, LVQ 네트워크에서의 출력 뉴런은 하나의 서브 클래스가 되고 여러 개의 서브 클래스를 묶어서 하나의 클래스로 지정한다. 또 하나의 차이점은 SOM은 학습 과정에서 입력 패턴 벡터만 이용하지만 LVQ 네트워크는 학습 과정에서 입력 패턴 벡터와 각각의 입력 패턴 벡터가 갖는 목표 벡터를 사용한다^{15 7)}.

위에서 논의한 것과 같이, LVQ 네트워크는 공간적으로 흩어져 있거나 이웃하는 클래스가 서로 겹치거나 군집화 된 패턴 벡터들을 보다 정교하게 분류하는 경우에 우수한 분류 성능을 갖지만 LVQ 네트워크의 설계에는 몇 가지 해결해야 할 문제를 가지고 있다. 첫 번째, LVQ 네트워크는 서브 클래스 수를 얼마로 할 것이며, 각 출력 뉴런이 어떤 서브 클래스를 지정하는지를 결정하는 것이다. 두 번째는 네트워크의 기준 벡터(reference vector)의 초기값을 어떻게 정하는가 하는 문제이다^{18 9)}.

본 논문에서는 이러한 LVQ 네트워크의 설계 문제를 해결하기 위하여 LVQ 네트워크의 서브 클래스의 수를 결정하고, 기준 벡터의 초기값을 정하기 위한 전처리 네트워크로 SOM을 이용하는 방식을 제안한다. 제안한 방식은 입력 패턴 벡터를 SOM을 이용하여 1차원 출력으로 설계를 하고, 반복 학습을 하여 LVQ 네트워크의 서브 클래스의 수를 결정한다. 또한 SOM과 LVQ 네트워크는 구조적으로 유사성이 있으므로 학습에서 얻어진 연결 강도를 LVQ 네트워크의 초기 기준 벡터로 사용하여 학습을 통해 보다 정교한 분류를 하는 방식이다. 제안한 방식으로 설계된 LVQ 네트워크의 분류 성능을 확인하기 위하여 Fisher의 Iris 데이터와 문자 인식에 적용하여 기존 LVQ 네트워크의 초기 기준 벡터 결정 방식인 입력 패턴 방식, 랜덤 방식 및 평균값 방식 등과 성능을 비교 검토한다.

II. SOM과 LVQ 네트워크

1. SOM

1984년 Kohonen이 제안한 SOM은 비지도 학습(unsupervised learning) 알고리즘의 일종으로 단일 경

쟁 뉴런층으로 구성된 신경회로망이다. 경쟁 뉴런은 1차원이나 2차원 혹은 그 이상의 차원에서 물리적으로 정돈되어 있고, 각 뉴런은 특정 반경의 이웃 뉴런을 가지게 된다.

일반적인 경쟁학습에서 뉴런들은 승자 뉴런을 결정하고, 승자 뉴런의 연결 강도를 강화하는 winner-takes-all 방식을 취한다. SOM이 일반적인 경쟁 학습과 다른 점은 승자 뉴런의 연결 강도뿐 아니라 이웃 뉴런의 연결 강도까지 갱신한다는 점이다. 학습이 진행되는 동안 결과적으로 승자 뉴런과 이웃하는 뉴런들은 비슷한 연결 강도를 가지게 되고, 승자 뉴런과 이웃 뉴런들은 유사한 입력 벡터에 반응하게 된다^{1 4, 8 9)}.

각 입력 패턴 x 에 대하여 고정 이웃 반경 내에서 연결 강도 w_j 는 식 (1)과 같이 조정한다.

$$w_j(t+1) = w_j(t) + h_{cj} [x - w_j(t)] \quad (1)$$

여기서, h_{cj} 는 이웃반경 함수이고, c 는 best-matching node, j 는 승리뉴런이다.

2. LVQ 네트워크

LVQ 네트워크의 기본 네트워크 구조는 SOM과 근본적으로 같지만, 다음과 같은 차이점이 있다. 출력 뉴런에 대하여 가해진 이웃반경이 없고, 각각의 출력 뉴런은 서브 클래스를 나타내고, 이 서브 클래스는 알려진 클래스에 속한다. 학습을 하는 동안 LVQ 네트워크의 출력 뉴런들은 이론적인 Bayes 분류기의 결정 표면에 접근하고, 학습 후에 LVQ 네트워크는 기준 벡터와 입력 벡터가 가장 가깝게 위치하는 출력 뉴런으로 입력 벡터를 분류한다. 따라서 입력 벡터의 클래스(T)와 승리뉴런의 클래스(C_j)가 같으면 기준 벡터의 학습은 새로운 입력 벡터의 방향으로 향하고, 입력 벡터의 클래스와 승리뉴런의 클래스가 다르면 기준 벡터의 학습은 새로운 입력 벡터와 반대방향으로 이동한다^{11 4, 8 9)}.

$$\begin{aligned} m_j(t+1) &= m_j(t) + \alpha [x - m_j(t)], \quad T = C_j \\ m_j(t+1) &= m_j(t) - \alpha [x - m_j(t)], \quad T \neq C_j \end{aligned} \quad (2)$$

여기서 m_j 는 기준 벡터이고, α 는 학습율이다.

일반적으로 기준 벡터를 초기화 하는 방법에는 LVQ 네트워크의 출력 뉴런 수와 같은 수의 입력 벡터를 기준 벡터로 사용하고 나머지 입력 벡터를 학습에 사용

하는 방법, 초기 기준 벡터를 랜덤하게 지정하는 방법, 입력 벡터의 최대, 최소 값의 평균값으로 모든 기준 벡터를 동일하게 초기화 하는 방법 등이 있다.

III. 제안한 설계 방식

LVQ 네트워크의 기준 벡터는 각 서브 클래스의 벡터를 나타내고, 그들의 초기 위치는 학습에 의한 네트워크 출력의 수렴에 영향을 주기 때문에 기준 벡터의 초기값 설정은 LVQ 네트워크의 분류 성능에 영향을 미친다. LVQ 네트워크에서 기준 벡터의 초기값이 국소적 접근 안정 평형점(locally asymptotic stable equilibrium)에 가까이 존재하고, 기준 벡터가 이곳으로부터 멀리 움직이지 않는다면, 이 기준 벡터는 수렴한다^[10-13]. 이 연구 결과로부터 LVQ 네트워크의 기준 벡터의 초기 위치의 설정이 전체 분류 성능을 결정하게 된다.

<그림 1>과 같이 확률 분포에서 클래스 A와 B가 서로 인접하고, 일부분이 겹친 경우에 LVQ 네트워크의 분류 성능을 향상시키기 위해서는 인접하고 겹쳐진 부분에 많은 기준 벡터가 위치하여 서브 클래스를 형성하면 클래스의 경계면을 확실하게 분류하여 높은 분류 성능을 얻을 수 있다^[14-15].

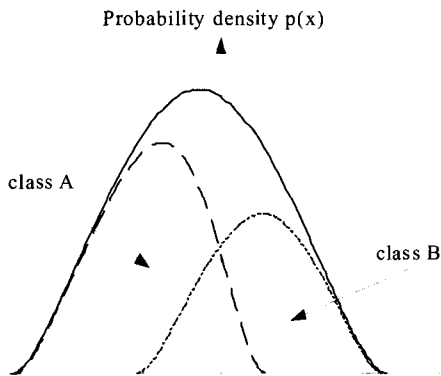


그림 1. 입력 데이터의 확률 분포
Fig. 1. A probability distribution of input data.

훈련 데이터(입력 벡터)의 확률 분포에 맞게 LVQ 네트워크의 초기 기준 벡터를 위치시키기 위해서 SOM을 이용한다. 입력 벡터를 $x \in \mathfrak{R}^n$ 이라 하고, LVQ 네트워크의 기준 벡터를 $m_i \in \mathfrak{R}^n$ 이라 하면 LVQ 네트워크는 입력 x 에 가장 근접한 m_c 를 찾는 것이다.

$$\begin{aligned} \|x - m_c\| &= \min_i \{\|x - m_i\|\}, \\ c &= \arg \min_i \{\|x - m_i\|\} \end{aligned} \quad (3)$$

양자화 오차를 식 (4)와 같이 정의하고, E 가 최소가 되도록 gradient를 계산하면 최적의 m_c 를 구할 수 있다.

$$E = \int \|x - m_c\|^2 p(x) dx \quad (4)$$

여기서 $p(x)$ 는 입력 벡터의 확률 밀도 함수이다. $\{a_i\}$ 를 양의 실수 집합이라 하면

$$\min_i \{a_i\} \equiv \lim_{r \rightarrow \infty} \left[\sum_i a_i^r \right]^{\frac{1}{r}} \quad (5)$$

로 정의할 수 있다. 식 (4)에서 $\|x - m_c\|^2$ 는

$$\|x - m_c\|^2 = \left[\min_i \{\|x - m_i\|\} \right]^2 = \lim_{r \rightarrow \infty} \left(\sum_i \|x - m_i\|^r \right)^{\frac{2}{r}} \quad (6)$$

이고, 식 (4)를 최소화 하기 위한 식은

$$\nabla_{m_j} E = \int \lim_{r \rightarrow \infty} \nabla_{m_j} \left(\sum_i \|x - m_i\|^r \right)^{\frac{2}{r}} p(x) dx \quad (7)$$

과 같다. 여기서

$$\sum_i \|x - m_i\|^r = A \quad (8)$$

라 하면, 식 (7)의 우변항은

$$\begin{aligned} \nabla_{m_j} A^{\frac{2}{r}} &= \frac{2}{r} A^{\left(\frac{2}{r}\right)-1} \cdot \nabla_{m_j} \left(\|x - m_j\|^r \right) \\ &= \frac{2}{r} A^{\left(\frac{2}{r}\right)-1} \cdot \nabla_{m_j} \left(\|x - m_j\|^2 \right)^{\frac{r}{2}} \end{aligned} \quad (9)$$

이고, 식 (9)를 정리하면

$$\nabla_{m_j} A^{\frac{2}{r}} = -2 \cdot \left(\frac{2}{r} \right) \cdot \frac{\left(\|x - m_j\|^2 \right)^{\left(\frac{r}{2}\right)-1}}{A} \cdot (x - m_j) \quad (10)$$

과 같다.

식 (6)에 식 (8)을 대입하면

$$\lim_{r \rightarrow \infty} A^{\frac{2}{r}} = \|x - m_c\|^2 \quad (11)$$

이다. 식 (10)에서 우변의 세번째 값을 B 라고 하면

$$B = \frac{\left(\|x - m_j\|^2\right)^{\left(\frac{r}{2}\right)^{-1}}}{A} = \frac{\|x - m_j\|^r}{\sum_i \|x - m_i\|^r} \cdot \|x - m_j\|^{-2}$$

$$= \left(\sum_i \frac{\|x - m_i\|^r}{\|x - m_j\|^r} \right)^{-1} \cdot \|x - m_j\|^{-2} \quad (12)$$

이다. $r \rightarrow \infty$ 가 되면 $\|x - m_i\|^r / \|x - m_j\|^r$ 은 $m_i = m_c$ 일 때 최대값을 갖는다. 그러므로

$$\lim_{r \rightarrow \infty} B = \lim_{r \rightarrow \infty} \left(\frac{\|x - m_j\|^r}{\|x - m_c\|^r} \right)^{-1} \cdot \|x - m_j\|^{-2}$$

$$= \delta_{cj} \|x - m_j\|^{-2} \quad (13)$$

라고 할 수 있다. 여기서 δ_{cj} 는 Kroneker Delta 함수이다.

이렇게 계산된 식들로부터 식 (10)은 식 (14)로 도출되고, 최종적으로 식 (15)의 결과를 얻을 수 있다.

$$\lim_{r \rightarrow \infty} \nabla_{m_j} A^{\frac{2}{r}} = -2 \cdot \|x - m_c\|^2 \cdot \delta_{cj} \cdot \|x - m_j\|^{-2} \cdot (x - m_j)$$

$$= -2 \cdot \delta_{cj} \cdot (x - m_j) \quad (14)$$

$$\nabla_{m_j} E = \int \lim_{r \rightarrow \infty} \nabla_{m_j} A^{\frac{2}{r}} \cdot p(x) dx$$

$$= -2 \int \delta_{cj} (x - m_j) p(x) dx$$

$$= -2 \delta_{cj} (x - m_j) \quad (15)$$

이와 같은 방식으로 SOM에서의 연결 강도 w_c 를 구할 수 있다. 우선 SOM은 이웃반경 함수 h_{ci} 가 학습에 의해 w_c 에 영향을 미치므로 왜곡 척도(distortion measure) e 를 정의한다.

$$e = \sum_i h_{ci} \|x - w_i\|^2 \quad (16)$$

식 (16)을 식 (6)과 같이 변형하여 표현하면 식 (17)과 같다.

$$e = \sum_i h_{ci} \|x - w_i\|^2$$

$$= \min_k \left\{ \sum_i h_{ki} \|x - w_i\|^2 \right\}$$

$$= \lim_{r \rightarrow \infty} \left(\sum_k \sum_i h_{ki} \|x - w_i\|^r \right)^{\frac{2}{r}} \quad (17)$$

과 같다.

양자화 오차를 식 (18)과 같이 구할 수 있다.

$$E = \int ep(x) dx \quad (18)$$

E 가 최소가 되도록 gradient를 계산하면 최적의 w_c 를 구할 수 있다.

$$\nabla_{w_j} E = \int \nabla_{w_j} ep(x) dx \quad (19)$$

$$\nabla_{w_j} e = \lim_{r \rightarrow \infty} \nabla_{w_j} \left(\sum_k \sum_i h_{ki} \|x - w_i\|^r \right)^{\frac{2}{r}} \quad (20)$$

여기서,

$$\sum_k \sum_i h_{ki} \|x - w_i\|^r = A \quad (21)$$

라 하면 식 (20)은

$$\nabla_{w_j} e = \lim_{r \rightarrow \infty} \nabla_{w_j} A^{\frac{2}{r}} \quad (22)$$

과 같다. 이 식을 풀면

$$\nabla_{w_j} A^{\frac{2}{r}} = \frac{2}{r} A^{\left(\frac{2}{r}\right)^{-1}} \cdot \nabla_{w_j} \left(\sum_k \sum_i h_{ki} \|x - w_i\|^r \right)$$

$$= \frac{2}{r} A^{\left(\frac{2}{r}\right)^{-1}} \cdot \nabla_{w_j} \left(\sum_k h_{kj} \|x - w_j\|^r \right)$$

$$= \frac{2}{r} A^{\left(\frac{2}{r}\right)^{-1}} \cdot \nabla_{w_j} \left(\sum_k h_{kj} \|x - w_j\|^2 \right)^{\frac{r}{2}} \quad (23)$$

이고 여기서

$$\begin{aligned} \nabla_{w_j} \left(\sum_k h_{kj} \|x - w_j\|^2 \right)^{\frac{r}{2}} &= \frac{r}{2} \left(\sum_k h_{kj} \|x - w_j\|^2 \right)^{\left(\frac{r}{2}\right)-1} \\ &\quad \cdot \sum_k h_{kj} \nabla_{w_j} \left(\|x - w_j\|^2 \right) \\ &= \frac{r}{2} \left(\sum_k h_{kj} \|x - w_j\|^2 \right)^{\left(\frac{r}{2}\right)-1} \cdot (-2) \cdot \sum_k h_{kj} (x - w_j) \end{aligned} \quad (24)$$

이므로 식 (23)은

$$\begin{aligned} \nabla_{w_j} A^{\frac{2}{r}} &= -2 \cdot \left(A^{\frac{2}{r}} \right) \cdot \frac{\left(\sum_k h_{kj} \|x - w_j\|^2 \right)^{\left(\frac{r}{2}\right)-1}}{A} \\ &\quad \cdot \sum_k h_{kj} (x - w_j) \end{aligned} \quad (25)$$

과 같다. 식 (17)에 의해서

$$\lim_{r \rightarrow \infty} A^{\frac{2}{r}} = \sum_i h_{ci} \|x - w_i\|^2 \quad (26)$$

이고 식 (25)의 두번째 항을 B 라고 하면

$$\begin{aligned} B &= \frac{\left(\sum_k h_{kj} \|x - w_j\|^2 \right)^{\left(\frac{r}{2}\right)-1}}{A} = \frac{\sum_k h_{kj} \|x - w_j\|^r}{\sum_k \sum_i h_{ki} \|x - w_i\|^r} \\ &\quad \cdot \left(\sum_k h_{kj} \|x - w_j\| \right)^{-2} \\ &= \left(\sum_k \sum_i \frac{h_{ki} \|x - w_i\|^r}{h_{kj} \|x - w_j\|^r} \right)^{-1} \cdot \left(\sum_k h_{kj} \|x - w_j\| \right)^{-2} \end{aligned} \quad (27)$$

이다. $r \rightarrow \infty$ 가 되면 $\|x - w_i\|^r / \|x - w_j\|^r$ 은 $w_i = w_c$ 이고 $k=c$ 일 때 최대값을 갖는다. 그러므로

$$\begin{aligned} \lim_{r \rightarrow \infty} B &= \lim_{r \rightarrow \infty} \left(\frac{h_{cj} \|x - w_j\|^r}{h_{ci} \|x - w_c\|^r} \right)^{-1} \cdot \left(\sum_k h_{kj} \|x - w_j\| \right)^{-2} \\ &= \delta_{cj} \cdot \left(\sum_k h_{kj} \|x - w_j\| \right)^{-2} \end{aligned} \quad (28)$$

이다.

$$\begin{aligned} \lim_{r \rightarrow \infty} \nabla_{w_j} A^{\frac{2}{r}} &= -2 \cdot \left(A^{\frac{2}{r}} \right) \cdot \delta_{cj} \cdot \left(\sum_k h_{kj} \|x - w_j\| \right)^{-2} \\ &\quad \cdot \sum_k h_{kj} (x - w_j) = -2 \cdot \left(\sum_i h_{ci} \|x - w_i\|^2 \right) \cdot \delta_{cj} \\ &\quad \cdot \left(\sum_k h_{kj} \|x - w_j\| \right)^{-2} \cdot \sum_k h_{kj} (x - w_j) \\ &= -2 \cdot \delta_{cj} \cdot \sum_k h_{kj} (x - w_j) \end{aligned} \quad (29)$$

$k=c$ 를 대입하면 하여 식 (29)는

$$\lim_{r \rightarrow \infty} \nabla_{w_j} A^{\frac{2}{r}} = -2 \cdot \delta_{cj} \cdot h_{cj} (x - w_j) \quad (30)$$

로 나타낼 수 있다.

최종적으로 식 (19)는

$$\begin{aligned} \nabla_{w_j} E &= \int \lim_{r \rightarrow \infty} \nabla_{w_j} A^{\frac{2}{r}} \cdot p(x) dx \\ &= -2 \int \delta_{cj} h_{cj} (x - w_j) p(x) dx \\ &= -2 \delta_{cj} h_{cj} (x - w_j) \end{aligned} \quad (31)$$

가 된다.

위의 SOM과 LVQ 네트워크에서 학습을 진행하는 동안 안정상태에 도달하려면 $\lim_{r \rightarrow \infty} \nabla_{m_i} E \rightarrow 0$, $\lim_{r \rightarrow \infty} \nabla_{w_j} E \rightarrow 0$ 이 되어야 하기 때문에 식 (15)와 식 (31)에서 $(x - m_i) \rightarrow 0$, $(x - w_j) \rightarrow 0$ 이 되어야 한다. 즉, 같은 입력 벡터 x 에 대해서 LVQ 네트워크의 초기 기준벡터인 m_j 와 SOM의 연결 강도인 w_j 는 $m_j \approx w_j$ 이 성립해야 한다는 것이다. 이러한 결과는 충분히 학습된 SOM의 연결강도는 학습된 LVQ 네트워크의 기준 벡터와 거의 같다는 것을 의미한다. 따라서 SOM으로 학습된 연결 강도를 LVQ 네트워크의 초기 기준 벡터로 사용한다면, LVQ 네트워크는 학습하는 동안 SOM으로 설정된 초기 기준 벡터의 이동량이 매우 작게 되어 기준 벡터는 수렴한다.

제한한 SOM을 이용한 LVQ 네트워크 설계 절차는 크게 두 단계로 구성된다. 첫 번째 단계는 1차원 구조의 SOM을 설정하고, 입력 벡터로 학습시킨다. 입력 벡터의 클래스마다 SOM의 승리 뉴런의 수가 LVQ 네트

워크의 서브 클래스의 수가 된다. 두 번째 단계는 SOM의 학습된 연결 강도를 LVQ 네트워크의 기준 벡터의 초기 값으로 정하고, 입력 벡터를 이용하여 LVQ 네트워크를 학습 시킨다.

<그림 2>는 제안한 방식의 블록 선도이다.

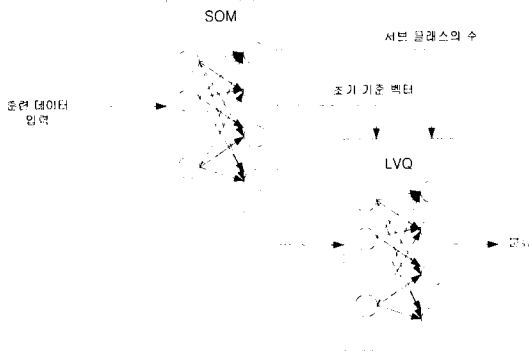


그림 2. 제안한 방식의 블록 선도
Fig. 2. A block diagram of proposed method.

IV. 시뮬레이션

제안한 방식의 유용성을 확인하기 위하여 Fisher의 Iris 데이터와 문자 인식을 시뮬레이션한다. 패턴 데이터를 훈련용과 시험용으로 나누고 분류 에러율을 가지고 기존 LVQ 네트워크와 제안한 방식의 분류 성능을 비교하였다. 기존 LVQ 네트워크의 초기 기준 벡터 설정은 입력 패턴 방식, 랜덤 방식, 평균값 방식에 대해서 MATLAB을 이용하여 Pentium III-800MHz PC상에서 수행하였다.

1. Fisher의 Iris 데이터 분류

Fisher의 Iris 데이터는 세가지 품종 'Setosa', 'Versicolor', 'Virginica'의 붓꽃(iris)으로부터 각각 50개, 총 150개의 개체들을 추출해서 측정한 자료로 부록의 <표 1>과 같다. 측정 변수는 꽃받침 조각의 길이(Sepal Length), 꽃받침 조각의 폭(Sepal Width), 꽃잎의 길이(Petal Length), 꽃잎의 폭(Petal Width)이다. 각각의 종에서 25개씩 75개의 데이터를 가지고 학습을 하였고, 나머지 75개의 데이터로 시험을 하여 분류 성능을 비교하였다.

기존 LVQ 네트워크의 구성은 입력 4개, 클래스 3개, 서브 클래스 5개, 학습율은 0.05로 하였다. 입력 패턴 방식은 훈련 데이터 중에서 15개를 초기 기준 벡터로 사용하였고, 랜덤 방식은 [0,1] 사이에서 랜덤하게 결정

하였고, 평균값 방식은 훈련 데이터 중에서 최대 값과 최소값의 평균인 0.39로 결정하였다.

제안한 방식에서 SOM의 구성은 입력 4개, 출력 15개, 초기 연결 강도는 임의로 선택하였고, 학습율은 0.9에서 학습에 따라 감소되며, 이웃 반경은 5에서 0으로 학습에 따라 감소된다. SOM을 1000회 학습한 후 그 연결 강도를 LVQ의 초기 기준 벡터로 사용하고, 서브 클래스의 수는 <표 1>과 같이 5개로 하여 결정하였다.

표 1. SOM의 출력
Table 1. Output of SOM.

		SOM의 승리뉴런														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
class	Versicolor						Virginic					Setosa				

LVQ 네트워크의 구성은 입력 4개, <표 1>과 같이 SOM과 동일하게 15개의 출력 뉴런으로 클래스 3개, 서브 클래스 5개로 하였다.

<표 2>와 <표 3>은 기존 LVQ 네트워크와 제안한 방식의 분류 성능 결과이다.

표 2. 훈련 데이터의 분류 에러
Table 3. Classification error of training data.

Iteration	500	1000	1500	2000	3000	5000	10000
입력패턴 방식	42	30	24	11	9	8	5
랜덤 방식	31	18	10	8	7	5	5
평균값 방식	27	12	9	9	8	6	5
제안한 방식			4	3	2	2	2

표 3. 시험 데이터의 분류 에러
Table 3. Classification error of test data.

Iteration	500	1000	1500	2000	3000	5000	10000
입력패턴 방식	40	33	30	23	15	8	8
랜덤 방식	46	21	17	14	11	9	8
평균값 방식	37	28	20	13	12	12	7
제안한 방식			4	4	3	2	2

<표 2>에서 제안한 방식이 기존 LVQ 네트워크의 초기 기준 벡터 설정 방식들에 비해 분류를 못하는 오차의 개수도 2개로 가장 적다. 학습된 기준 벡터를 이용하여 시험 데이터를 분류한 결과에서도 제안한 방식이 훈련 데이터와 유사하게 우수한 분류 성능을 확인

할 수 있었다.

2. 문자 인식

3가지 형태의 폰트 데이터를 이용하여 제안한 방식의 분류 성능을 확인하였다. 각 폰트는 7개의 영문 대문자로 구성되어 있다. 부록의 <그림 1>은 훈련 데이터이고 <그림 2>는 시험 데이터이다. 시험 데이터는 훈련 데이터에 노이즈가 첨가된 폰트이다^[8].

기존 LVQ 네트워크의 구성은 입력 63개, 클래스 7개, 서브 클래스 3개, 학습율은 0.05로 하였다. 입력 패턴 방식은 훈련 데이터 중에서 21개를 초기 기준 벡터로 사용하였고, 랜덤 방식은 [0,1]사이에서 랜덤하게 결정하였고, 평균값 방식은 0.5로 결정하였다.

제안한 방식에서 SOM의 구성은 입력 63개, 출력 21개, 초기 연결 강도는 임의로 선택하였고, 학습율은 0.9에서 학습에 따라 감소되며, 이웃 반경은 5에서 0으로 학습에 따라 감소된다. SOM을 1000회 학습한 후 그 연결 강도를 LVQ의 초기 기준 벡터로 사용하고, 서브 클래스의 수는 <표 4>과 같이 3개로 하여 결정하였다.

LVQ 네트워크의 구성은 SOM과 동일하게 입력 63개, 출력뉴런은 <표 4>와 같이 클래스 7개, 서브 클래스

표 4. SOM의 출력
Table 4. Output of SO.

	SOM 출력																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
class	C			B			J			K			A			D			E		

표 5. 훈련 데이터의 분류 에러
Table 5. Classification error of training data.

Iteration	500	1000	1500	2000	3000	5000	10000
입력패턴 방식	19	10	7	4	2	1	0
랜덤 방식	17	9	4	2	2	1	1
평균값 방식	19	15	10	4	2	1	0
제안한 방식			3	1	0	0	0

표 6. 시험 데이터의 분류 에러
Table 6. Classification error of test data.

Iteration	500	1000	1500	2000	3000	5000	10000
입력패턴 방식	15	8	5	3	2	2	0
랜덤 방식	18	11	4	2	2	2	1
평균값 방식	20	16	9	5	2	1	1
제안한 방식			4	1	0	0	0

스 3개로 하였다.

<표 5>과 <표 6>는 기존 LVQ 네트워크와 제안한 방식의 분류 성능 결과이다.

<표 5>와 <표 6>에서 기존 LVQ 네트워크의 초기 기준 벡터 설정 방식들에 비해 분류 오차의 개수가 학습 횟수 3000번에서 0개로 되어 제안한 방식이 훈련 데이터와 유사하게 우수한 분류 성능을 확인할 수 있었다.

이 두 가지의 시뮬레이션을 통해서 제안한 방식이 기존 LVQ 네트워크의 초기 기준 벡터를 설정하는 방식들에 비해 분류 성능이 우수함을 확인하였다.

V. 결 론

본 논문에서는 LVQ 네트워크의 설계 문제를 해결하기 위하여 SOM을 이용하여 LVQ 네트워크를 설계하는 방식을 제안하였다. 제안한 방식은 LVQ 네트워크의 서브 클래스들의 수를 결정하고, 기준 벡터의 초기값을 결정하기 위한 전처리 부분으로 SOM을 이용하는 방식이다. SOM을 이용하여 서브 클래스의 개수를 결정하고, SOM과 LVQ 네트워크는 구조적으로 유사성이 있으므로 학습에서 얻어진 연결 강도를 LVQ 네트워크의 초기 기준 벡터로 사용하여 학습을 통해 보다 정교한 분류를 할 수 있었다.

제안한 방식으로 설계된 LVQ의 분류 성능을 확인하기 위하여 Fisher의 Iris 데이터 분류와 문자 인식에 적용하였다. 시뮬레이션 결과 제안한 방식이 기존 LVQ 네트워크의 초기 기준 벡터 설정 방식인 입력 패턴 방식, 랜덤 방식, 평균값 방식보다 학습 횟수가 감소되었음을 확인하였으며, 분류성능도 우수함을 확인하였다.

부 록

표 1. Fisher의 Iris Data Set(mm)
Table 1. Fisher's Iris Data Set(mm).

Setosa				Versicolor				Virginic			
Sepal Length	Sepal Width	Petal Length	Petal Width	Sepal Length	Sepal Width	Petal Length	Petal Width	Sepal Length	Sepal Width	Petal Length	Petal Width
51	35	14	02	70	32	47	14	63	33	60	25
49	30	14	02	64	32	45	15	58	27	51	19
47	32	13	02	69	31	49	15	71	30	59	21
46	31	15	02	55	23	40	13	63	29	56	18
50	36	14	02	65	28	46	15	65	30	58	22
54	39	17	04	57	28	45	13	76	30	66	21

Setosa				Versicolor				Virginic			
Sepal Length	Sepal Width	Petal Length	Petal Width	Sepal Length	Sepal Width	Petal Length	Petal Width	Sepal Length	Sepal Width	Petal Length	Petal Width
46	34	14	03	63	33	47	16	49	25	45	17
50	34	15	02	49	24	33	10	73	29	63	18
44	29	14	02	66	29	46	13	67	25	58	18
49	31	15	01	52	27	39	14	72	36	61	25
54	37	15	02	50	20	35	10	65	32	51	20
48	34	16	02	59	30	42	15	64	27	53	19
43	30	14	01	60	22	40	10	68	30	55	21
48	30	11	01	61	29	47	14	57	25	50	20
58	40	12	02	56	29	36	13	58	28	51	24
57	44	15	04	67	31	44	14	64	32	53	23
54	39	13	04	56	30	45	15	65	30	55	18
51	35	14	03	58	27	41	10	77	38	67	22
57	38	17	03	62	22	45	15	77	26	69	23
51	38	15	03	56	25	39	11	60	22	50	15
54	34	17	02	59	32	48	18	69	32	57	23
51	37	15	04	61	28	40	13	56	28	49	20
46	36	10	02	63	25	49	15	77	28	67	20
51	33	17	05	61	28	47	12	63	27	49	18
48	34	19	02	64	29	43	13	67	33	57	21
50	30	16	02	66	30	44	14	72	32	60	18
50	34	16	04	68	28	48	14	62	28	48	18
52	35	15	02	67	30	50	17	61	30	49	18
52	34	14	02	60	29	45	15	64	28	56	21
47	32	16	02	57	26	35	10	72	30	58	16
48	31	16	02	55	24	38	11	74	28	61	19
54	34	15	04	55	24	37	10	79	38	64	20
52	41	15	01	58	27	39	12	64	28	56	22
55	42	14	02	60	27	51	16	63	28	51	15
49	31	15	02	54	30	45	15	61	26	56	14
50	32	12	02	60	34	45	16	77	30	61	23
55	35	13	02	67	31	47	15	63	34	56	24
49	36	14	01	63	23	44	13	64	31	55	18
44	30	13	02	56	30	41	13	60	30	48	18
51	34	15	02	55	25	40	13	69	31	54	21
50	35	13	03	55	26	44	12	67	31	56	24
45	23	13	03	61	30	46	14	69	31	51	23
44	32	13	02	58	26	40	12	58	27	51	19
50	35	16	06	50	23	33	10	68	32	59	23
51	38	19	04	56	27	42	13	67	33	57	25
48	30	14	03	57	30	42	12	67	30	52	23
51	38	16	02	57	29	42	13	63	25	50	19
46	32	14	02	62	29	43	13	65	30	52	20
53	37	15	02	51	25	30	11	62	34	54	23
50	33	14	02	57	28	41	13	59	30	51	18

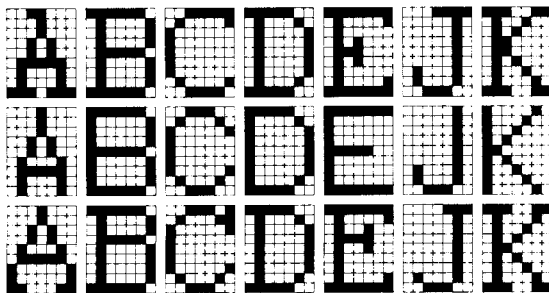


그림 1. 훈련 데이터
Fig. 1. Training dat.

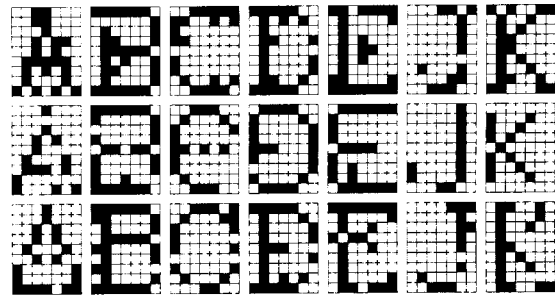


그림 2. 시험 데이터
Fig. 2. Test dat.

참 고 문 헌

[1] T. Kohonen, "The self-organizing map," Proceedings of the IEEE, vol. 78, no. 9, pp. 1464-1480, 1990.

[2] T. Kohonen, "The self-organizing map," Neuro-computing, vol. 21, no. 1-3, pp. 1-6, 1998.

[3] T. Kohonen, "Generalizations Of The Self-organizing map," Proceedings of International Joint Conference on Neural Networks, vol. 1, pp. 457-462, 1993.

[4] T. Kohonen, E. Oja, O. Simula, A. Visa and J. Kangas, "Engineering applications of the self-organizing map," Proceedings of the IEEE, vol. 84, no. 10, pp. 1358-1384, 1996.

[5] A. Baraldi, P. Blonda, F. Parmiggiani, G. Pasquariello and G. Satalino, "Model transitions in descending FLVQ," IEEE Transactions on Neural Networks, vol. 9, no. 5, pp. 724-738, 1998.

[6] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," IEEE Transactions on Neural Networks, vol. 11, no. 3, pp. 586-600, 2000.

[7] N.B. Karayiannis, J.C. Bezdek, N.R. Pal, R.J. Hathaway and Pin-I Pai, "Repairs to GLVQ: a new family of competitive learning schemes," IEEE Transactions on Neural Networks, vol. 7, no. 5, pp. 1062-1071, 1996.

[8] Laurene Fausett, Fundamentals of Neural Networks, Prentice Hall, 1994.

[9] Martin T. Hagan, Neural Network Design,

Thomson Learning, 1996.

- [10] Allen Gersho and Robert M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Pub., 1992.
- [11] S. P. Luttrell, "Derivation of a class of training algorithms," IEEE Transactions on Neural Networks, vol. 1, no. 2, pp. 229-232, 1990.
- [12] S. P. Luttrell, "Code vector density in topographic mappings: Scalar case," IEEE Transactions on Neural Networks, vol. 2, no. 4, pp. 427-436, 1991.
- [13] H. Ritter, "Asymptotic level density for a class of vector quantization processes," IEEE Transactions on Neural Networks, vol. 2, no. 1, pp. 173-175, 1991.
- [14] Tanaka, T. Saito, M., "Quantitative Properties of Kohonen's Self-Organizing Maps as Adaptive Vector Quantizers (Japanese)," IEICE transactions on information and systems, D-II, Vol. J75-D-II, No. 6, pp. 1085-1092, 1992.
- [15] N. Kitajima, "A new method for initializing reference vectors in LVQ," IEEE Int. Conf. on Neural Networks, vol. 5, pp. 2775-2779, Nov/Dec. 1995.

저 자 소 개

鄭慶權(正會員) 第39卷 CI編 第4號 參照

嚴基煥(正會員) 第38卷 SC編 第4號 參照



李鎔求(正會員)

1987년 : 동국대학교 전기공학과 졸업(공학사). 1993년 : 동국대학교 대학원 전기공학과 졸업(공학석사). 1997년 2월 : 동국대학교 전자공학과 졸업(공학박사). 1998년 3월 ~ 현재 : 한림정보산업대학 전자정보

통신과 조교수. <주관심분야 : 지도 및 비지도학습 신경 회로망, 퍼지논리시스템, 패턴인식, 시스템식별 및 제어>