

## 수정된 자기 구조화 특징 지도를 이용한 한국어 음소 인식

최두일 · 이수진 · 박상희  
연세대학교 전기공학과

Korean Phoneme Recognition using Modified Self Organizing Feature Map

Doo-Il Choi    Su-Jin Lee    Sang-Hui Park  
Dept. of Electrical Eng., Yonsei Univ.

**Abstract**

In order to cluster the input pattern neatly, some neural network modified from Kohonen's self organizing feature map is introduced and Korean phoneme recognition experiments are performed using the modified self organizing feature map(MSOFM) and the auditory model.

**1. 서론**

음성신호 인식기로서 가장 우수한 성능을 가진 인간의 메카니즘에 관한 연구는 전처리단으로서의 청각계통에 관한 연구와 인식단으로서의 뇌에 관한 연구도 나눌 수 있다.

인간의 청각계통에 관한 연구는 Bekesy[1]가 코클리어 메카닉스에 관하여 연구한 이후로 지금까지 많은 생리학적 연구가 진행되어 왔다.

뇌의 기능을 모델화한 신경회로망에 관한 연구는 McCulloch와 Pitts가 신경세포의 모델에 관하여 연구한 이후로 꾸준히 진행되어 많은 모델이 제안되었다. 음성인식을 위한 신경회로망 모델은 back propagation learning rule을 이용한 다층 퍼셉트론과 self organizing rule을 이용한 특징 지도등이 널리 이용되고 있다.

본 연구에서는 self organizing feature map의 문제점을 실험적으로 지적하고 수정된 self organizing feature map을 제안한다. 한편, 음성신호의 전처리를 위하여, 말초 청각 계통의 생리학적 특성을 반영한 계산 모델을 구성하여 한국어 음성신호에 대한 특징 추출부로 이용하고 제안된 신경회로망을 인식부로 이용하여 음소 인식 실험을 수행하고자 한다.

**2. 말초청각계통모델**

말초청각계통의 전체 모델은 그림 1과같이 외이, 중이, 내이 및 헤어셀 모델로 구성하였다.

식 (1)과 식 (2)는 본 연구에서 사용한 외이 및 중이모델 [2]의 전달함수를 나타내고 있다.

$$H_E(s) = \frac{P(s)}{S(s)} = \frac{1}{(s + 1000)^2 + (7875\pi)^2} \quad (1)$$

$$H_M(s) = \frac{X(s)}{P(s)} = \frac{1}{s + 6000\pi} \quad (2)$$

식 (3)은 본 연구에서 사용한 내이 모델[3]의 전달함수를 나타내고 있다.

$$H_I(s) = \begin{cases} \frac{(\beta_1/3)s}{s^2 + (\beta_1/3)s + \beta_1^2} & , f_1 \leq 300\text{Hz} \\ C_1 \cdot \beta_1^4 \left( \frac{2000\pi}{\beta_1 + 2000\pi} \right)^{0.8} \cdot \left[ \frac{s}{s + \beta_1} \right] \cdot \left[ \frac{1}{(s + \alpha_1)^2 + \beta_1} \right]^2 & , f_1 > 300\text{Hz} \end{cases} \quad (3)$$

여기서,  $f_1$  : 기저막의 각 위치에서의 중심 주파수

$$\beta_1 = 2\alpha_1 = 2\pi f_1$$

$$C_1 \cdot \beta_1^4 \left( \frac{2000\pi}{\beta_1 + 2000\pi} \right)^{0.8} : \text{이득 요소}$$

식 (4)-(7)은 헤어셀 모델[4]의 기본식을 나타내고 있다.

$$p(t) = P_0 \{ \frac{1}{2} \cdot y_1 + [ \frac{1}{4} y_1^2(t) + 1 ]^{1/2} \} \quad (4)$$

$$r = \overline{n(t) \cdot p(t)} + \overline{n(t) \cdot g} \quad (5)$$

$$f(t) = n(t) \cdot p(t) \text{ or } n(t) \cdot p(t) \cdot \rho(t-t_0) \quad (6)$$

$$\frac{dn}{dt} = r - n(t) [ p(t) + g ] \quad (7)$$

**3. Kohonen의 자기 구조화 특징 지도**

일반적으로 경쟁학습의 안정성은 초기의 랜덤 연결 강도에 크게 의존한다. 극단적인 경우에는, 어떤 경우에도 학습되지 않는 쓸모없는 출력 노드가 다수 존재할 수 있다. 이러한 문제점을 해결하기 위하여 많은 방법들이 제안되었다.[5]

현재까지 제안된 많은 경쟁학습 알고리즘 가운데, Kohonen[6]이 제안한 자기 구조화 특징 지도(SOFM: Self Organizing Feature Map)는 입력 패턴을 자동적으로 적응 양자화할 수 있을 뿐 아니라, 출력 노드를 가지런히 구조화할 수 있다는 점에서 가장 우수한 경쟁 학습 알고리즘 가운데 하나라고 알려져 있다. 그러나, SOFM은 몇가지 단점을 보이고 있다. 첫째로, 입력 패턴이 그림 2와 같은 순환성 분포를 갖는 경우 출력층의 중심 노드들이 쓸모없이 되어 버린다. 그림 2는 그와 같은 실험 결과를 보이고 있다. 더군다나 심각한 경계 효과(boundary effect)가 보인다. 이러한 경계 효과는 구석에 위치한 노드가 중심에 위치한 노드보다 더 많이 학습되기 때문에 발생하며 경계 효과를 제거하기 위해서는 매우 오랜 학습 시간이 필요하게 된다. 그림 3은 경계 효과가 발생하는 극단적인 경우를 보이고 있다. 이 경우 두 입력 X1과 X2는 상호 독립이며 평균 0, 편차 20/3의 가우시안 분포를 갖고있다. 특징 지도는 입력 분포의 평균 점으로 움츠러들어서 퍼져 나가려면 매우 오랜 시간이 걸리게 된다.

두번째로, 이는 경쟁 학습 알고리즘의 일반적인 문제이지만, 입력 패턴이 복잡한 분포를 갖는 경우에는 회로망이 매우 불안정해진다. SOFM에서는 확률 밀도가 0인 영역에 존재하는 연결 강도 벡터가 주위의 0이 아닌 입력의 영향을 받아서 입력 벡터 영역으로 수렴하지 못하는 경우가 종종 발생된다. 이러한 현상은 이웃(neighbor)이 감소함에 따라 없어지지만 잔여 효과는 그대로 남게 된다.[7] 그림 4는 그와 같은 실험 결과를 나타낸다.

반면에 그림 5의 사각형 분포와 같이 입력 벡터의 분포가 매우 균일하다면 연결 강도 벡터군은 입력 데이터에 산뜻하게 적응된다.

#### 4. 수정된 자기 구조화 특징 지도

출력 노드의 가지런한 배열과 입력 패턴의 안정적인 분할을 위하여 Kohonen은 SOFM을 제안하였다. Kohonen의 생각은 뉴런간의 상호 작용은 감소하는 이웃의 개념을 이용하여 쉽게 구현될 수 있다는 것이다. 그는 세가지 형태의 특징 지도를 소개하였는데 그것은 그림 6부터 그림 8에서 나타내듯이 1 차원 배열, 사각형 배열 및 육각형 배열이다. 여기서 ⊗는 이진 노드를 나타내고 ⊙는 이웃 노드를 나타낸다.

3장에서 언급했듯이 한가지 문제점은 구석 근처에 위치한 출력 노드들이 중심 근처에 위치한 노드들 보다 항상 많이 학습되기 때문에 지도가 입력 분포의 중심으로 움츠러든다. 이러한 현상은 이웃이 없어짐에 따라 멈추지만 어떤 노드들의 연결 강도 벡터들은 한곳에 몰리기도 하고 경계 효과가 발생하기도 하는 잔여 효과가 남는다.

우리의 생각은 만약 출력 지도가 순환성이어서 이웃노드의 갯수가 위치에 상관없이 항상 같다면 모든 노드가 학습

될 확률은 항상 같게 되고 지도는 초기에도 안정성을 갖을 것이라는 것이다. 그림 9와 그림 10은 수정된 SOFM(MSOFM)에서의 이웃의 한 예를 보이고 있다.

그림 11부터 그림 14는 그림 2부터 그림 5의 SOFM과 같은 입력 분포 - 환경, 가우시안, 구조화 및 사각형 분포-에 대하여 MSOFM을 실험한 결과이다. 경계 효과가 발생하거나 0 입력 영역에 연결 강도 벡터가 존재하는 문제점들은 여전히 남아있지만 SOFM에 비하여 감소함으로써 MSOFM에 향상됨을 알 수 있었다. (표 1 참조)

#### 5. 한국어 음소 인식 실험 및 결과

전체 인식 시스템의 블록선도는 그림 15와 같다.

음성신호의 전처리를 위한 청각 모델의 내이 필터의 중심 주파수는 50[Hz] - 6248[Hz]까지의 범위를 선택하였고 그 범위를 1[Bark] 간격으로 20채널을 선택하였다. 중심 주파수의 [Hz] 단위의 변환은 Zwicker와 Terhardt[8]가 제시한 식을 이용하였다.

신경회로망은 입력층과 출력층의 2층으로서, 입력 노드는 20개, 출력 노드는 가로12 × 세로12의 사각형 배열로 구성하였다. 인식에 사용된 음성은 13개 한국어 기본자음 + 모음 /아/로 구성하였다. 청각 모델 출력은 상 길이 12.8[msec]의 Hamming창을 띄워 6.4[msec]씩 이동시키면서 음성의 특징 벡터를 추출하였다. 음성신호의 특징 벡터는 약 60개로 이루어진다. 그림 16은 /가/, /다/, /바/의 청각 모델 출력 특징 벡터를 나타낸다.

신경 회로망의 학습은 랜덤하게 선택된 음성 특징 벡터로 60,000회 실시하였으며, 이웃 범위  $N_c$ 와 학습률  $\alpha(t)$ 는 각각 식(8) 및 식(9)와 같이 시간에 따라 단조 감소하도록 하였다.

$$N_c = \begin{cases} 6 - \frac{t}{800}, & 0 < t < 4000 \\ \frac{36000 - t}{32000}, & 4000 < t < 36000 \\ 0, & 36000 < t < 60000 \end{cases} \quad (8)$$

$$\alpha(t) = \begin{cases} 0.1 - \frac{0.99 \times t}{4000}, & 0 < t < 4000 \\ 0.001 - \frac{0.099 \times (t - 36000)}{32000}, & 4000 < t < 36000 \\ \frac{0.001 \times (60000 - t)}{28000}, & 36000 < t < 60000 \end{cases} \quad (9)$$

본 연구에서는 8개의 유성 자음과 모음 /ㅏ/에 대한 특징 지도와 5개의 무성 자음과 모음 /ㅓ/에 대한 특징 지도를 따로 구성하여 학습 하였으며, 그림 17과 그림 18은 각각 유성 음소와 무성 음소의 최종 특징 지도를 나타낸다.

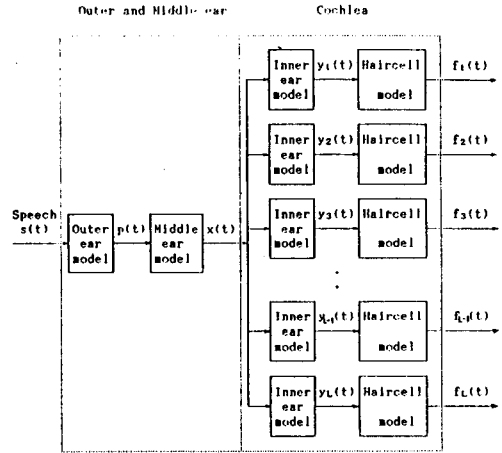
학습에 사용된 신호와 같은 음절을 동일한 화자가 다시 발음한 2개의 비학습 데이터를 그 화자의 음성을 학습한 신경회로망에 입력하여 화자 종속 인식 실험을 수행하였다. 그 결과 각 음절을 이루는 약 70개 정도의 벡터에 관한 인식 결과가 모든 음소에 대해서 화자 1에서는 94.7[%], 화자 2에서는 95.2[%]의 인식률을 나타내었다.

### 6. 결론 및 토의

본 연구에서는 순환성 이웃의 개념을 도입하여 자기 구조화 특징 지도를 구성함으로써 신경 회로망의 성능을 개선시켰으며, 이 신경 회로망과 청각 모델을 조합하여 음성인식 실험을 수행하여 높은 인식률을 얻을 수 있었다. 본 연구의 결과는 향후 단어 인식 및 어의 인식의 기초가 될 수 있으리라고 기대된다.

### 참고 문헌

- [1] G.V. Békésy: Experiment in hearing, Robert E. Krieger Publishing Company, Huntington, New York, 1960.
- [2] D.M. Monro, "Computer modelling of the peripheral mechanical response of the auditory system" in Auditory Investigation : The scientific and technological basis, edited by H.A. Beagley, Clarendon Press, pp 431 - 450, 1979.
- [3] J.L. Flanagan, Speech analysis, synthesis and perception, Springer - Verlag, Berlin, 1984
- [4] 윤태성: "청각 모델을 이용한 한국어 단음의 인식에 관한 연구", 박사학위 논문, 연세대학교 대학원, 1988.
- [5] D. E. Rumelhart, J. L. McClelland, and the PDP research group., Parallel distributed processing, Vol. 1, pp 151-193, MIT Press, 1986
- [6] T. Kohonen, Self Organization and Associative Memory. 2nd edition, Springer-Verlag, ch. 5, pp.119-157, 1988
- [7] J. Kangas, T. Kohonen, J. Laaksonen, "Variants of Self-Organizing Maps" IEEE Trans. on Neural Network, Vol. 1, pp.93-99, 1990
- [8] E. Zwicker and E. Terhardt, " Analytical expressions for critical band rate and critical bandwidth as a function of frequency", J. Acoust. Soc. Am. Vol.68, pp 1523 - 1525, 1980.



$s(t)$  : speech data  
 $p(t)$  : outer ear output  
 $x(t)$  : middle ear output  
 $y_i(t)$  : inner ear output at  $i$ -th channel  
 $f_i(t)$  : haircell output at  $i$ -th channel

그림 1 전체 청각 모델

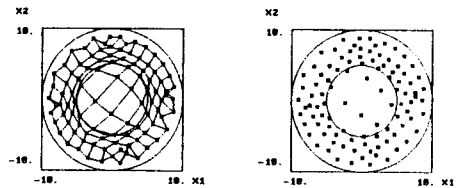


그림 2 환경 입력 분포에 대한 10x10 사각형 배열 출력노드를 갖는 SOFM의 최종 양자화 결과 (a) 이웃 노드간을 선으로 연결한 경우 (b) 이웃 노드간을 선으로 연결하지 않은 경우

\* 시뮬레이션 변수 :  
 $\alpha(t) = 0.02$ , ITER = 10000, NB = Integer{10.1 exp(-15t/ITER)}, 여기서  $\alpha$ 는 학습률,  $t$ 는 시간, ITER는 iteration 수 그리고 NB는 이웃의 범위.

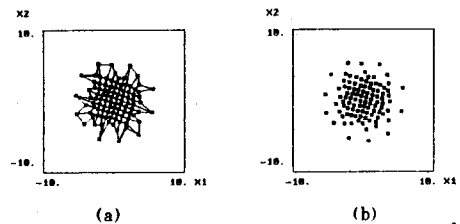


그림 3 가우시안 입력 분포에 대한 10x10 사각형 배열 출력노드를 갖는 SOFM의 최종 양자화 결과 (a) 이웃 노드간을 선으로 연결한 경우 (b) 이웃 노드간을 선으로 연결하지 않은 경우

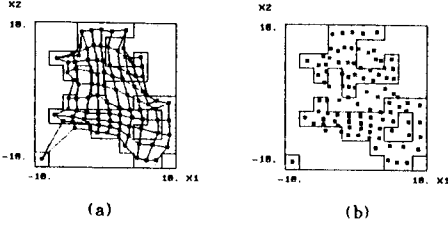


그림 4 구조화 입력 분포에 대한 10×10 사각형 배열 출력 노드를 갖는 SOFM의 최종 양자화 결과 (a) 이웃 노드간을 선으로 연결한 경우 (b) 이웃 노드간을 선으로 연결하지 않은 경우

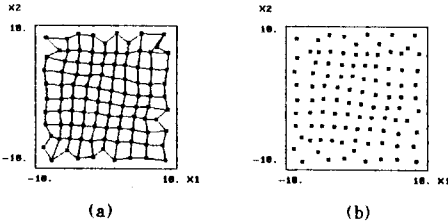


그림 5 사각형 입력 분포에 대한 10×10 사각형 배열 출력 노드를 갖는 SOFM의 최종 양자화 결과 (a) 이웃 노드간을 선으로 연결한 경우 (b) 이웃 노드간을 선으로 연결하지 않은 경우



그림 6 1 차원 배열의 SOFM에서의 이웃의 한 예 (a) 이진 노드가 중앙에 위치한 경우 (b) 이진 노드가 가장 자리에 위치한 경우

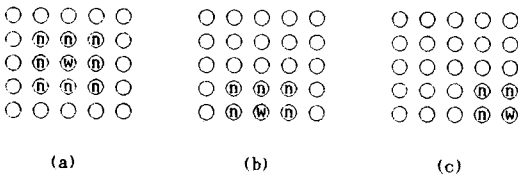


그림 7 사각형 배열의 SOFM에서의 이웃의 한 예 (a) 이진 노드가 중앙에 위치한 경우 (b) 이진 노드가 변에 위치한 경우 (c) 이진 노드가 모서리에 위치한 경우

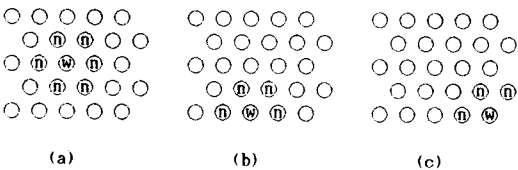


그림 8 육각형 배열의 SOFM에서의 이웃의 한 예 (a) 이진 노드가 중앙에 위치한 경우 (b) 이진 노드가 변에 위치한 경우 (c) 이진 노드가 모서리에 위치한 경우



그림 9 1 차원 배열의 MSOFM에서의 이웃의 한 예 (a) 이진 노드가 중앙에 위치한 경우 (b) 이진 노드가 가장 자리에 위치한 경우

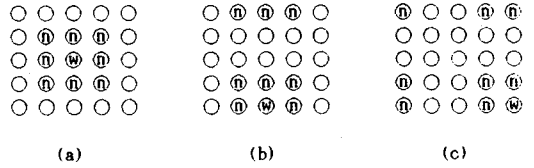


그림 10 사각형 배열의 MSOFM에서의 이웃의 한 예 (a) 이진 노드가 중앙에 위치한 경우 (b) 이진 노드가 변에 위치한 경우 (c) 이진 노드가 모서리에 위치한 경우

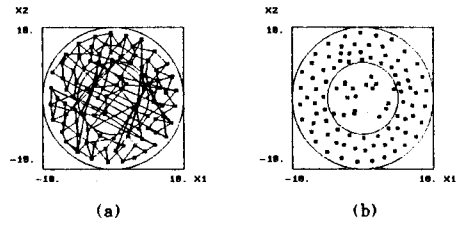


그림 11 환형 입력 분포에 대한 10×10 사각형 배열 출력 노드를 갖는 MSOFM의 최종 양자화 결과 (a) 이웃 노드간을 선으로 연결한 경우 (b) 이웃 노드간을 선으로 연결하지 않은 경우

\* 시뮬레이션 변수 :  
 $\alpha(t) = 0.02$ , ITER = 10000, NB = integer(5.1e xp(-15t/ITER))

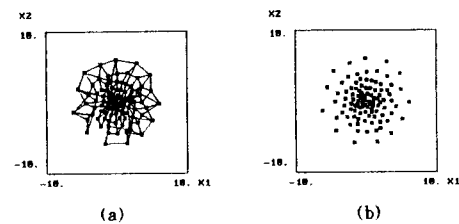


그림 12 가우시안 입력 분포에 대한 10×10 사각형 배열 출력 노드를 갖는 MSOFM의 최종 양자화 결과 (a) 이웃 노드간을 선으로 연결한 경우 (b) 이웃 노드간을 선으로 연결하지 않은 경우

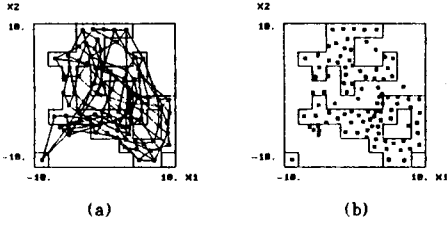


그림 13 구조화 입력 분포에 대한 10x10 사각형 배열 출력 노드를 갖는 MSOFM의 최종 양자화 결과 (a) 이웃 노드간을 선으로 연결한 경우 (b) 이웃 노드간을 선으로 연결하지 않은 경우

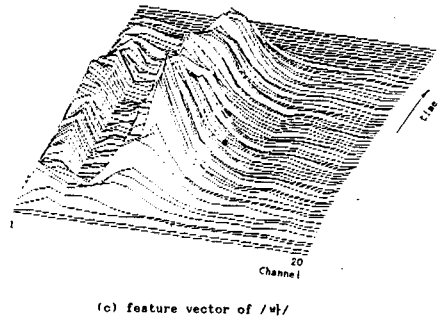


그림 16 청각 모델을 이용한 /가/, /다/, /바/의 특징벡터

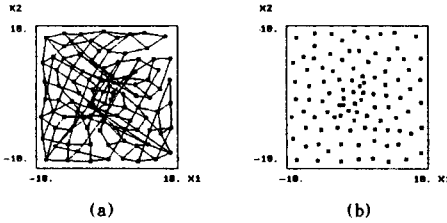


그림 14 사각형 입력 분포에 대한 10x10 사각형 배열 출력 노드를 갖는 MSOFM의 최종 양자화 결과 (a) 이웃 노드간을 선으로 연결한 경우 (b) 이웃 노드간을 선으로 연결하지 않은 경우

ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ
ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ
ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ
ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅕ	ㅗ
ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅗ	ㅛ	ㅜ	ㅠ
ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅛ
ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ
ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ
ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ
ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ

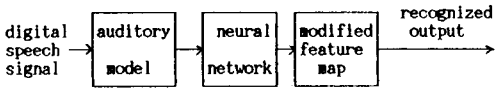
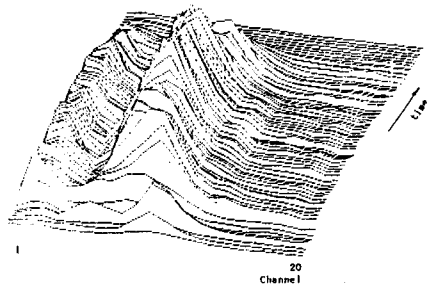
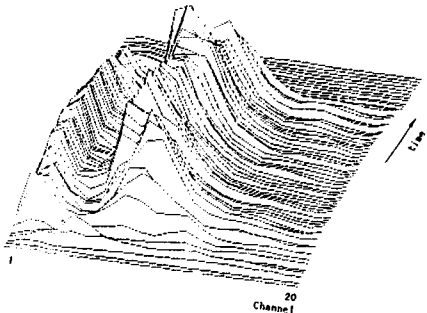


그림 15 인식 시스템 블록 선도

그림 17 유성음소의 특징 지도



(a) feature vector of /가/



(b) feature vector of /다/

ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ																													
ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ																						
ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ																	
ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ														
ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ													
ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ									
ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ									
ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ						
ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ			
ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ	ㅣ	ㅜ	ㅠ	ㅡ	ㅣ

그림 18 무성음소의 특징 지도

표 1. 평균 자승 양자화 오차

(a) 2-D 균일 분포

MSE	type	SOFM	MSOFM
1		0.803322	0.782120
2		0.791167	0.807354
3		0.809663	0.782788
norm. avg.	1		0.986736

\* 시뮬레이션 조건

두 입력  $x_1$  과  $x_2$  가 상호 독립이며  $[-10, 10]$ 에서 균일한 경우

ITER = 10000, 출력 노드 갯수 = 100

SOFM :  $\alpha(t) = 0.02$ , NB = integer{10.1 exp(-15t/ITER)}

MSOFM :  $\alpha(t) = 0.02$ , NB = integer{5.1 exp(-5t / ITER)}

여기서  $\alpha$  는 학습률, t 는 시간, ITER 는 iteration 횟수,

R 은 답음 요소, RR 은 응답 영역, NB 는 이웃의 범위.

(b) 2-D 구조화 분포

MSE	type	SOFM	MSOFM
1		0.410249	0.408091
2		0.438084	0.389172
3		0.414525	0.399563
norm. avg.	1		0.947713

\* 시뮬레이션 조건

두 입력  $x_1$  과  $x_2$  가  $[-10, 10]$ 에서 구조화된 경우

ITER = 10000, 출력 노드 갯수 = 100

SOFM :  $\alpha(t) = 0.02$ , NB = integer{10.1 exp(-15t/ITER)}

MSOFM :  $\alpha(t) = 0.02$ , NB = integer{5.1 exp(-5t / ITER)}

(c) 2-D 가우시안 분포

MSE	type	SOFM	MSOFM
1		0.398097	0.377423
2		0.402458	0.357844
3		0.425296	0.353706
norm. avg.	1		0.888341

\* 시뮬레이션 조건

두 입력  $x_1$  과  $x_2$  가 상호 독립이며  $[-10, 10]$ 에서 가우시안 분포를 갖는 경우

ITER = 10000, 출력 노드 갯수 = 100

SOFM :  $\alpha(t) = 0.02$ , NB = integer{10.1 exp(-15t/ITER)}

MSOFM :  $\alpha(t) = 0.02$ , NB = integer{5.1 exp(-5t / ITER)}

(d) 3-D 균일 분포

MSE	type	SOFM	MSOFM
1		6.010944	5.079556
2		5.911648	5.005223
3		6.077623	5.018165
norm. avg.	1		0.839042

\* 시뮬레이션 조건

세 입력  $x_1, x_2, x_3$  가 상호 독립이며  $[-10, 10]$ 에서 균일한 경우

ITER = 10000, 출력 노드 갯수 = 100

SOFM :  $\alpha(t) = 0.02$ , NB = integer{10.1 exp(-15t/ITER)}

MSOFM :  $\alpha(t) = 0.02$ , NB = integer{5.1 exp(-5t / ITER)}

(e) 3-D 구조화 분포

MSE	type	SOFM	MSOFM
1		2.568118	2.074659
2		2.369979	1.800154
3		2.225391	1.850075
norm. avg.	1		0.799176

\* 시뮬레이션 조건

세 입력  $x_1, x_2, x_3$  가  $[-10, 10]$ 에서 구조화된 경우

ITER = 10000, 출력 노드 갯수 = 100

SOFM :  $\alpha(t) = 0.02$ , NB = integer{10.1 exp(-15t/ITER)}

MSOFM :  $\alpha(t) = 0.02$ , NB = integer{5.1 exp(-5t / ITER)}

(f) 3-D 가우시안 분포

MSE	type	SOFM	MSOFM
1		2.017907	1.672191
2		1.973864	1.716892
3		2.045576	1.669963
norm. avg.	1		0.837958

\* 시뮬레이션 조건

세 입력  $x_1, x_2, x_3$  가 상호 독립이며  $[-10, 10]$ 에서 가우시안 분포를 갖는 경우

ITER = 10000, 출력 노드 갯수 = 100

SOFM :  $\alpha(t) = 0.02$ , NB = integer{10.1 exp(-15t/ITER)}

MSOFM :  $\alpha(t) = 0.02$ , NB = integer{5.1 exp(-5t / ITER)}