

# 신경회로망을 이용한 인쇄체 한글 문자의 인식

## (The Recognition of Printed Korean Characters by a Neural Network)

金 相 宇\*, 全 允 豪\*, 崔 悰 鎬\*

(Sang Woo Kim, Yun Ho Jeon, and Chong Ho Choi)

### 要 約

이 논문에서는 인쇄체 한글문자 인식에 있어서 신경회로망의 적용가능성을 알아 보았다. 한글 문자 수의 과다와 그들 사이의 유사성, 많은 입력 영상 데이터 등으로 인하여 신경회로망을 한글인식에 적용시키는데는 많은 난점이 따른다. 한글 문자의 이진영상은 신경회로망의 입력으로 사용하기에는 그 데이터 수가 너무 많으므로 입력 영상으로부터 DC 성분을 추출하여 이것을 신경회로망의 입력으로 사용하기 위한 전처리과정을 두었다. 출력층은 한글의 특성에 맞도록 구성하였다. 한글인식에 도입된 신경회로망은 다층인식자이고, 적용된 훈련방법은 BEP 알고리즘을 한글인식에 적절하도록 변형시킨 형태이다. 이 방법을 통하여 정위치에 있는 2,300개 이상의 문자를 인식할 수 있었다. 이 결과로부터 신경회로망을 이용한 인쇄체 한글문자 인식은 적절한 방법임을 알 수 있다.

### Abstract

The potential of neural networks for the recognition of the printed Korean characters is examined. In spite of good classification capability of neural networks, it is difficult to train a neural network to recognize Korean characters. The difficulty is due to a large number of Korean characters, the similarities among the characters, and the large number of data from the character images. To reduce the input image data, DC components are extracted from each input images. These preprocessed data are used as input to the neural network. The output nodes are composed to represent the characteristics of Korean characters. A MLP (multilayer perceptron) with one hidden layer was trained with a modified BEP algorithm. This method gives good recognition rate for the standard positioned characters of more than 2,300. The result shows that neural networks are well suited for the recognition of printed Korean characters.

### I. 서 론

신경회로망은 '병렬분산 정보처리체제'로 요약할 수 있는데, 간단한 역할을 수행하는 노드들의 병

렬구성체라는 구조의 단순성에 비하여 그것이 해결할 수 있는 일들이 매우 다양하다는 데에 그 매력에 있다. 특히 순차적 프로그램에 의하여 일이 수행되는 기존의 Von Neumann type 컴퓨터가 그 기능을 제대로 발휘하지 못하는 분야에서 신경회로망은 새로운 돌파구로 기대를 모으고 있다. 그러나 이제까지 고안된 신경회로망은 그 성능평가나 응용에 있어

---

\*正會員, 서울대학교 制御計測工學科  
(Dept. of Cont. Inst. Eng., Seoul Nat'l Univ.)  
接受日字 : 1989年 9月 28日

서 그 수준이 대부분 단순작업을 넘지 못하고 있는데 이는 신경회로망을 실제적으로 구현하는 과정에서 현재의 환경에서 해결할 수 없는 몇가지 문제점이 발생하기 때문이다. 본 논문에서는 신경회로망이 이제까지 응용되어 온 것 중에서 가장 큰 장점을 보이고 있는 분류(classification)기능을 복잡한 분류작용으로 볼 수 있는 인쇄체 한글의 인식에 이용하여 봄으로써 한글인식 과정에 대한 신경회로망의 응용 가능성을 알아 보고 아울러서 기존의 인공 지능 분야에서 추구하는 방법과는 다른 형태의, 신경회로망을 이용한 효과적인 한글인식 알고리즘을 제시하려고 한다.

한글은 초성 19개, 중성 21개, 종성 27개의 기본 자모로 구성되어 있지만, 이들이 서로 결합하여 조합문자를 이룬다는 특징을 지닌다. 따라서 가능한 조합문자의 갯수는 총 11,172자에 이르는데<sup>[1]</sup>이중에 실제로 널리 쓰이는 것은 1,500-2,000자를 넘지 못하는 것으로 알려져 있다. 그럼에도 불구하고 단순 자소의 어울림이 이같은 조합문자를 만들기 때문에 많은 수의 유사 형태의 문자를 생성하여 한글문자인식을 어렵게 하는 요인으로 작용한다. 따라서 기존의 한글인식 알고리즘에서는 대개 전처리과정(preprocessing)을 두어서 문제의 해결을 시도하고 있다. 전처리과정으로 많이 이용된 것은 획(stroke)에 대한 정보를 추출하는 방법<sup>[2,3,4]</sup> 및 FFT에 의한 입력자료의 변환<sup>[5]</sup> 등으로서, 이 경우도 이러한 난점을 완전히 해결하고 있지는 못하다.

최근 들어서 신경회로망을 한글문자 인식에 이용하려는 움직임이 활발해지고 있다.<sup>[7,8]</sup> 그러나 다층인식자와 BEP 알고리즘의 단순 응용만으로는, 분류대상이 수천에 이르는 점, 그들 문자 사이의 유사성, 또 실제로 입력되는 문자 영상의 크기 등으로 인하여 훈련이 잘 되지 않으며 인식 가능한 문자의 갯수도 한계를 보이고 있다. 따라서 신경회로망의 이러한 분류능력상 한계를 극복하기 위한 방법들이 모색되고 있다.

본 논문에서 사용한 문자 체계는 24×24화소의 이진영상(binary image)으로 구성된 한글 프린터에 사용되는 글자로서 그 갯수는 모두 2,350이다.<sup>[6]</sup> 인식을 위하여 사용된 신경회로망의 구조는 다층인식자(multilayer perceptron)이며 이를 위하여 쓰인 학습 방법은 BEP(back-error propagation) 알고리즘의 변형된 형태이다. 원래의 한글 이진영상을 그대로 신경회로망의 입력으로 사용하지 않고, 전처리 과정을 두어서 한글 이진영상을 적절한 다른 형태로 변형시킨 것을 신경회로망의 입력으로 사용하였다. 출력층

노드는 초·중·종성으로 이루어진 한글의 특성에 알맞도록 구성하였다. 그리고 신경회로망의 구조는 중간층(hidden layer)을 1층을 가지도록 했으며, 입력층과 출력층을 이어주는 중간층의 각 노드들은 입력층의 모든 노드들과 연결되도록 하였다.

입력 영상이 표준 위치에 있을 경우에 대하여 먼저 학습을 시킨 후, 문서로부터 스캐너를 통한 문자 추출을 가상하여 입력영상의 좌우이동(left-right shift)에 대해서도 학습을 시도하여 보았다. 또한 중간층 노드의 갯수가 한글인식 과정에 어떤 영향을 주는가에 대하여 알아보았으며 다층인식자의 학습속도를 개선시키는 방법에 대해서도 알아보았다.

본 논문에서는 제안된 신경회로망의 구조와 그 동작 특성, 적용된 학습 알고리즘과 전체 훈련과정, 그리고 실제 데이터를 이용하여 인식 실험한 결과를 보이기로 한다. 그럼으로써 한글인식에 있어서 신경회로망의 적용가능성을 알아 보고자 한다.

## II. 신경회로망 및 전처리과정

### 1. 신경회로망

인식에 사용한 신경회로망은 다층인식자로서 한층의 중간층 및 입력층과 출력층으로 이루어져 있다.

각층 사이의 노드의 연결은 전방향으로만 이루어지는데 입력층을 제외한 중간층과 출력층의 각 노드  $j$ 의 실제 출력값(actual output)은

$$f(x_j) = \frac{1}{1 + \exp\{- (x_j + \theta_j)\}} \quad (1)$$

$$x_j = \sum_i \omega_{ij} o_i$$

$o_i$  :  $x_j$ 에 연결된 이전 층의 실제 출력값

$\omega_{ij}$  :  $o_i$ 와  $x_j$ 를 연결하는 가중치값

$\theta_j$  :  $x_j$ 의 bias 값

형태의 sigmoid 함수로 주어지며 따라서 실제 출력값의 범위는 0.0에서 1.0사이에 위치하게 된다.

출력층의 노드의 갯수는 49개로서 그림1에 주어져 있는데 그 내용은 초성 19개, 중성 14개, 종성 16개로 구성되어 있다. 원래 초·중·종성을 합한 갯수는 67개이지만, 이 중에서 중성의 가로모음과 세로모음이 복합된 경우는 이를 분리했고, 중성에서도 서로 다른 두 자음이 결합된 것은 역시 분리했다. 이렇게 해도 중성의 경우는 모음이 결합하는 순서 때문에, 중성의 경우는 서로 다른 두 자음의 결합이 바뀌어서 다른 중성을 형성하는 일은 없기 때문에 인식 결과가 애매성을 보이지 않는다.

출력을 표시할 수 있는 또다른 방법은 초·중·종성

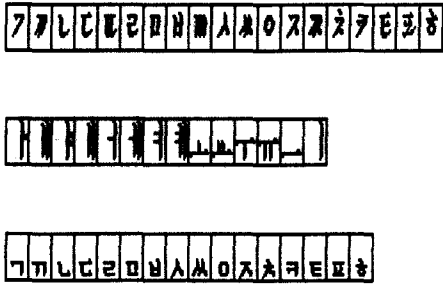


그림 1. 49개의 출력노드  
Fig. 1. 49 output nodes.

에 대한 고려없이 인식에 사용되는 글자 각각을 이진코드(binary codes)로 나타내는 것으로서<sup>[7]</sup> 이 경우 출력노드의 갯수는 감소한다. 그러나 혼란시켜야 할 글자 세트의 방대함으로 인하여, 그들에 관한 정보가 분산되어 저장되는데는 이러한 연결선의 감소가 바람직하다고만 볼 수 없으며 오히려 문자 인식 및 훈련에 역효과를 줄 수 있다. 그림1에서와 같이 출력력을 넓게 퍼뜨려 놓음으로써 가중치의 갯수는 증대하지만, 한글의 특성에 따른 이러한 출력층의 구성은 타당하다고 생각한다.

중간층 노드의 갯수는 우선 100개를 설정하였다. 중간층 노드의 갯수에 관한 정확한 기준은 알려진 것이 없기 때문에 처음에는 출력층 노드 갯수의 이배수 정도를 중간층 노드의 갯수로 삼았으며 이것이 변화할 경우 전체 인식 과정에 어떤 영향을 주는지에 대해서도 알아 보았다.

2. 전처리과정

신경회로망의 입력층의 노드의 갯수는 모두 96개로서 이는 입력영상인 24×24화소의 이진 데이터를 바로 입력으로 사용하지 않고 적절한 전처리과정을 두었기 때문이다. 이렇게 한 이유는 크게 두가지를 들 수 있는데, 먼저 24×24 화소를 그대로 입력으로 사용하는 경우에는 입력 노드의 갯수가 모두 576개로서 이는 중간층 노드와 어울릴 경우에 입력층과 중간층 사이의 가중치 갯수가 수만개에 이르게 되어 결과적으로 막대한 훈련 시간을 요구하게 되는데 본 논문에서 도입한 전처리과정은 입력의 갯수를 96개로 줄임으로써 가중치의 갯수를 감소시키고 이것은 결국 훈련 시간을 줄일 수 있도록 하여 준다. 다음으로는 전처리과정을 거친 후의 입력자료가 원래의 한글 화소 영상보다 한글 자체가 지니는 특징을 나타내는데 좀더 효과적일 수도 있다는 점을 들 수 있다.

즉 576개의 이진 화소는 인식에 열쇠가 되는 주요 특징(essential features)외에 많은 양의 중요도가 떨어지는 정보를 포함할 수 있는데 전처리과정을 통하여 이런 정보들을 어느 정도 걸러낼 수 있을 것으로 생각된다.

본 논문에서 시도한 전처리과정은 그림2에 나타나 있다.

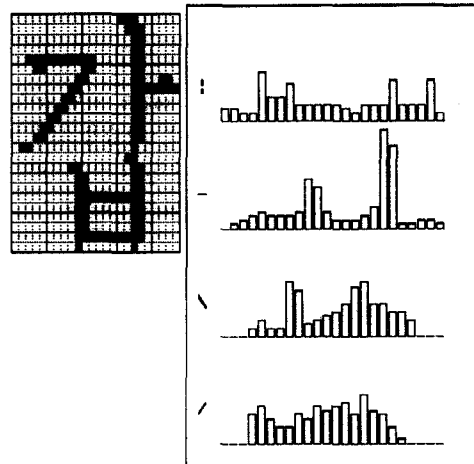


그림 2. 전처리과정 : 96개의 DC 성분  
Fig. 2. Preprocessing : 96 DC components.

그림2는 ‘갑’자로부터 얻은 96개의 DC 성분으로서 처음 24개의 데이터 각각은 가로방향의 24입력화소 더한 값이고, 다음 24개의 데이터 각각은 세로방향의 24입력화소를 더한 값이다. 세번째 24개의 데이터 각각은 +45° 방향으로 두줄씩 화소를 더한 것을 2로 나눈 값이고 마지막 24개의 데이터 각각은 +135° 방향으로 두줄씩 화소를 더한 것을 2로 나눈 값이다.

이상의 96개의 데이터는 0°, 45°, 90°, 135°방향의 DC 성분을 끄집어 낸 셈으로서 실제로 한글을 2차원 FFT하여 보면 DC 성분이 그 글자의 주요 특징을 이루고 있음을 확인할 수 있다.<sup>[8]</sup> 이렇게 구한 DC 값들은 0에서 24까지의 정수값들로서 이것을 24로 나누어서 0.0에서 1.0까지의 값으로 정규화시킨 것(normalized value)을 노드의 실제 입력값으로 삼았다.

III. 훈련 과정

신경회로망을 훈련시키기 위하여 사용된 학습 알고리즘은 Rumelhart 등이 제안한 BEP 알고리즘<sup>[9]</sup>의

변형으로서 이는 지도학습이론(supervised learning)의 대표적인 방법이다.

학습을 시작하기 전의 가중치들은 영(zero) 근처의 임의의 값으로 초기화시켜 놓게 되며 학습이 진행됨에 따른 BEP 알고리즘의 가중치를 변형시키는 식은 아래층(lower layer)에 속한 노드  $i$ 와 위층(upper layer)에 속한 노드  $j$ 를 연결하는 가중치  $\omega_{ij}$  및 패턴(pattern)  $p$ 에 대하여

$$\Delta \omega_{ij}(n+1) = \eta \delta_{pj} \omega_{ij} + \alpha \Delta \omega_{ij}(n) \quad (2)$$

로 주어진다.

$\Delta \omega_{ij}$ 는 가중치의 변화량이고,  $\omega_{ij}$ 는 패턴  $p$ 에 대한 노드  $i$ 의 출력값이며, 상수  $\eta$ 는 학습률로서 이 값이 클수록 가중치의 변화량이 커지게 된다. 이 값이 커지게 되면 학습속도가 증가한다는 장점은 있지만 수렴하지 못하고 진동할 우려가 있으며 너무 작으면 진동은 막아지지만 학습 속도가 느려지는 단점이 생긴다. 다음 상수  $\alpha$ 는 관성항(momentum)으로서 전번 단계의 계산에서 구해진 가중치의 변화 경향을 다음 단계에 반영시키는 역할을 하는데 이렇게 함으로써 수렴속도를 증가시킨다.<sup>(9)</sup>

본 논문에서는 학습률은 0.5, 관성항은 0.8을 사용하였는데 관성항은 BEP 알고리즘에서 많이 쓰이는 값을 택하였고, 학습률은 시뮬레이션 과정을 통하여 경험적으로 얻은 값으로서 진동이 생기지 않는 범위내에서 가능한한 큰 값을 택하였다.

$\delta_{pj}$ 는 패턴  $p$ 에 대하여 역방향으로 전파된 오차의 분담값으로서 출력층에 있는 노드  $j$ 에 대해서는

$$\delta_{pj} = (t_{pj} - o_{pj}) o_{pj} (1 - o_{pj}), \quad (3)$$

$t_{pj}$ : 패턴  $p$ 에 대한 노드  $j$ 의 목표 출력값(target output)

$o_{pj}$ : 패턴  $p$ 에 대한 노드  $j$ 의 실제 출력값(actual output)

으로 정의되고, 중간층에 있는 노드  $j$ 에 대해서는

$$\delta_{pj} = -o_{pj} (1 - o_{pj}) \sum_k \left( \frac{\delta_{pk} \omega_{jk}}{\Omega_k} \right) \quad (4)$$

로 정의되는데, 여기서  $k$ 는 노드  $j$ 의 윗층에 속한 노드를 나타내는 첨자이다.  $\Omega_k$ 는 Rumelhart 등이 제안한 BEP 알고리즘에는<sup>(9)</sup> 없는 요소로서 노드  $k$ 에 연결되는 가중치  $\omega_{jk}$ 에 대하여

$$\Omega_k = \min \{ 0.2 \times \sum_j |\omega_{jk}|, 1.0 \} \quad (5)$$

로 정의된다. 이 요소는 윗층에서 계산된 오차분담 값  $\delta_{pk}$ 가 아래층으로 전달됨에 따라 그 크기가 지나

치게 작아지는 것을 방지하여 수렴속도를 증가시키기 위한 것이다. 위의 (4)에서  $\Omega_k$ 의 역할은 학습이 진행됨에 따라 점점 작아지게 되는데 이것은 처음에 영 부근의 임의의 값으로 정해 놓았던  $\omega_{jk}$  값들이 훈련이 진행됨에 따라 영으로부터 멀어지고 따라서 (5)에 의하여  $\Omega_k$ 가 1이 될 가능성이 높아지기 때문이다. 결국 (5)는 훈련이 진행됨에 따라 Rumelhart 등이 제안한 알고리즘과 일치하는 방향으로 작용하게 된다.

학습에 쓰이는 모든 패턴이 한번씩 훈련에 이용된 동안을 sweep이라 하는데 학습 과정중에 이루어지는 위와 같은 가중치의 변형은, 매 패턴마다 이루어질 수도 있고 또는 매 패턴에서 발생하는 오차의 분담값을 한 sweep 동안 더했다가 매 sweep마다 이루어질 수도 있다. 패턴의 개수가 적은 경우라면 두 방법은 차이가 별로 없지만 훈련시킬 패턴의 수가 많은 경우에는 그 패턴이 지나는 국부적인(local) 특징을 전체에 반영시킨다는 문제가 있음에도 불구하고 매 패턴마다 가중치를 변형시키는 것이 더 좋다. 이러한 사실은 실제 한글인식 시뮬레이션 과정에서 확인할 수 있었는데 매 sweep마다 가중치의 변형이 이루어지는 경우 과도한 패턴의 수로 인하여 거의 학습이 되지 않는 현상을 볼 수 있었다.

또 하나 고려된 것은 훈련과정에서 매 패턴마다 발생하는 오차를 조사하여 그것이 미리 정하여 놓은 최소오차(minimum error)수준이상일 경우에만 가중치를 변화시키도록 한 것이다. 이것은 Sawai 등이 제안한 방법인데<sup>(10)</sup> 특히 한글처럼 훈련시킬 패턴의 수가 많아서 전체 훈련시간이 길어질 우려가 있을 경우에는 유용한 방법으로 생각된다.

#### IV. 결과 및 분석

신경회로망의 구현 및 시뮬레이션은 PC-386(20 MHz) 상에서 이루어졌다. 신경회로망의 규모때문에 많은 시간이 소모되기는 했지만 몇가지 경우에 있어서 바람직한 결과를 얻을 수 있었다. 전처리과정을 거치기 전의 한글 영상 화소는 명조체 24×24의 이진영상으로서 실험에 사용된 전체 글자수는 2,350자였다.

시뮬레이션과정을 통하여 조사된 것은 현재 sweep에서 각 입력문자에 대하여 그 입력문자가 출력된 후의 글자오차, 매 sweep 당 전체오차, 매 sweep이 끝났을 때까지 인식된 문자의 개수 등이다. 글자  $k$ 에 대한 글자오차(character error)는

$$\text{character error}(k) := [\sum_i (t_i(k) - a_i(k))^2] / 49 \quad (6)$$

$t_i(k)$  : 목표 출력값(target output)

$a_i(k)$  : 실제 출력값(actual output)

$i$  : 출력 노드의 index( $i=1, 2, \dots, 49$ )

로 주어지는데 목표출력값은 0.9 및 0.1을 기준으로 삼았다. 전체오차(total error)는 훈련에 쓰인 문자들 각각의 글자오차를 산술평균한 값으로서

$$\text{total error} := \frac{1}{M} \sum_{i=1}^M \text{character error}(i), M: \text{전체글자수} \quad (7)$$

으로 정의된다. 그리고 인식의 성공여부는 목표에 해당하는 출력노드의 실제 출력값이 0.5를 넘는가를 판단기준으로 삼았다.

처음 시도한 것은 2,350자 전체에 대한 학습으로서, 한 sweep 훈련에 걸린 시간은 1시간 정도이고, 모두 50 sweep 동안 학습이 이루어졌다. 훈련 결과로서 모두 2,337자 정도가 학습됨을 확인할 수 있었는데 그림3은 훈련과정을 나타낸 것으로서 sweep이 진행됨에 따른 인식문자수의 증가현상을 보여주고 있다. 이는 전처리과정이 없이 입력 영상을 바로 유형별 신경회로망의 입력으로 사용한 기존의 방법<sup>17)</sup>에 비하여 전처리과정을 도입한 본 논문의 인식과정이 더 효과적임을 보여주고 있다.

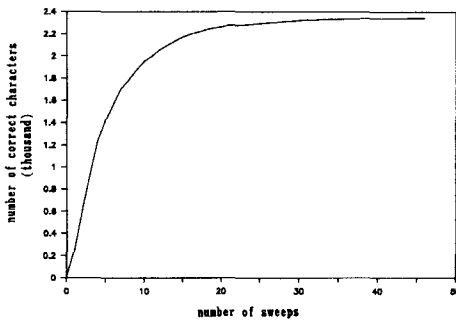


그림 3. 다층인식자의 한글인식훈련 (2,350자)  
Fig. 3. Training of multilayer perceptron for 2,350 characters.

다음은 2,350자의 25%에 해당하는 588자를 임의로 선택한 후, 이것을 훈련 세트토 사용하여 신경회로망을 훈련시킨 후, 이렇게 훈련된 가중치 집합을 이용하여 전체 글자 집합에 대하여 어떻게 반응하는가를 살펴 보았는데 1,998자를 인식하였다. 이는 전

체의 25%를 훈련에 이용하여 전체의 85% 정도를 인식해낸 것으로서 훈련된 신경회로망이 입력범위에 대하여 어느 정도 일반화(generalization)를 하고 있음을 알 수 있다.

588자의 문자 세트에 대해서는 중간층 노드의 갯수를 변화시켜서 학습을 시도하여 보았는데 100 sweep이 지난 후의 그 인식 및 전체 글자세트에 대한 일반화 정도는 아래표와 같다.

중간층 노드의 갯수	인식정도	일반화 정도	(학습시간)/sweep
100	583	1,998/2,350	15 min
80	582	1,917/2,350	12 min
50	564	1,710/2,350	10 min
30	555	1,663/2,350	7 min

표에서 알 수 있듯이 중간층 노드의 갯수가 적어지면 신경회로망이 포함할 수 있는 정보량이 적어지고 따라서 전체 문자 세트에 대한 일반화도 제대로 이루어지고 있지 못함을 볼 수 있다. 다음으로 고려할 것은 중간층 노드의 갯수와 학습속도와의 관계로서 그림 4, 그림 5에 보여지고 있는데 인식문자의 수에 있어서 중간층 노드의 갯수가 적어지면 대체로 그 성능이 저하됨을 알 수 있다. 그림 4, 그림 5에서 주목할 점은 중간층 노드의 갯수가 100개일 때와 80개일 때로서, 40 sweep 정도에 전체훈련 세트를 거의 인식하고 있고 그 성능면에서도 별로 차이를 보이고 있지 않다. 이것은 주어진 문자 세트를 훈련시킬 경우 중간층 노드의 갯수가 줄어들 여지가 있음

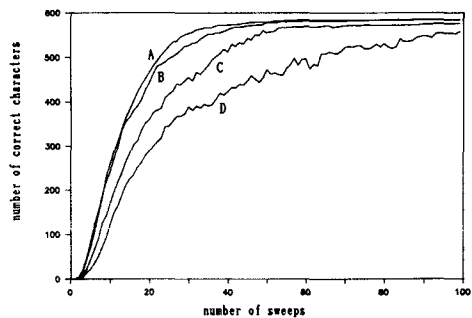


그림 4. 학습에 따른 인식글자수 (588자)  
중간층 노드의 갯수 (A : 100, B : 80, C : 50, D : 30)  
Fig. 4. The number of correctly recognized characters. (588 characters)  
The number of hidden nodes (A : 100, B : 80, C : 50, D : 30).

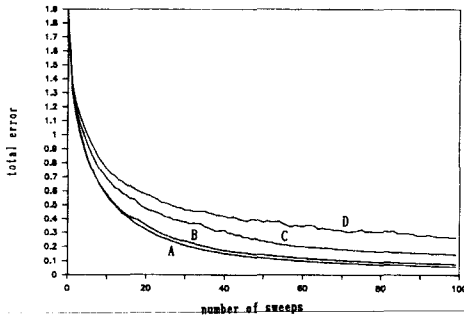


그림 5. 학습에 따른 전체오차 (588자)  
 Fig. 5. Total error. (588 characters)  
 The number of hidden nodes  
 (A: 100, B: 80, C: 50, D: 30).

을 의미하며, 따라서 2,350자 전체에 대한 훈련에서도 100개보다 적은 갯수의 중간층 노드로도 학습이 가능할 것으로 생각된다.

이상에서는 입력이 매우 이상적인 형태라고 가정하였으나 궁극 목표인 스캐너에 의한 문자 인식을 가상하기 위하여 좌우로 이동된 문자에 대해서 인식정도를 실험하여 보았다. 훈련에 쓰인 것은 588자 및 그것들이 좌우로 각각 한 비트씩 이동된 후의 1,4764(=588×3)자로서 훈련과정에서의 인식정도는 1,740자 정도였다.

좀더 악화된 환경을 고려하기 위하여 기본자체가 좌우로 각각 두 비트씩 이동된 것까지 포함한 2,940(=588×5)자에 대하여 훈련시켜 보았는데 그 훈련과정은 그림6과 같다. 2,940자 중 인식과정에서 인

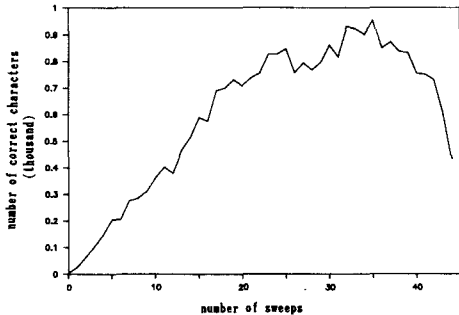


그림 6. 두 비트 좌우 이동에 대한 다층인식자의 훈련 (2,940자)  
 Fig. 6. Training of multilayer perceptron for the two bit shifted 2,940 characters.

식한 최대의 글자수는 952자이고 그것을 고비로 하여 점차 인식정도가 떨어짐을 알 수 있는데, 이는 변환된 글자 세트 전체를 수용하기에는 신경회로망의 규모가 작기 때문인 것으로 생각되며 인식정도가 떨어지는 현상은 앞으로 좀 더 규명하여야 할 것이다. 그리고 이러한 변환에 무관(invariant) 하기 위해서는 다른 형태의 신경회로망을 고려해야 할 것으로 생각된다.

이제까지는 (2)의 BEP 알고리즘을 이용하여 학습시킨 경우들이고 다음에 고려된 것은 Sawai의 방법을<sup>[10]</sup> 도입하여 훈련시킨 경우이다. 그림7은 588자의 문자 세트에 대하여, 최소오차 수준이상의 글자오차가 발생했을 경우에만 가중치 변화를 시도한 것과 그렇지 않은 것을 같이 실어 놓은 것이다. 가중치 변화를 위한 최소오차는 글자오차가 0.1인 경우를 삼았다. 그림에서는 매 sweep 횟수의 증가에 따른 인식 글자수의 증가에 있어 두 방법 모두 별 차이가 없는 것처럼 보이지만, 최소오차를 도입한 경우 실제로 소요된 훈련시간에 있어서 25% 정도를 줄일수 있었으며, 훈련 과정에서 전체 문자세트에 대한 100% 인식이 이루어짐을 확인할 수 있었다. 따라서 비슷한 성능을 지니는 신경회로망을 훈련시킬 때 최소오차를 도입해도 무방함을 알 수 있었으며 특히 보통의 방법으로 훈련이 잘 되지 않는 몇몇 특이한 입력패턴의 학습에 매우 유용한 방법임을 확인할 수 있었다.

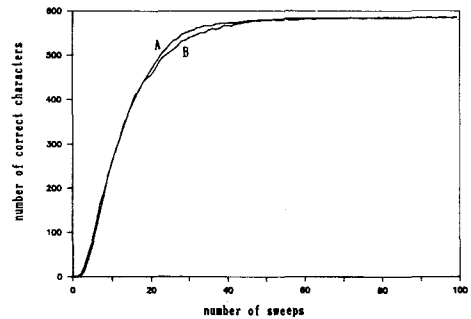


그림 7. BEP 알고리즘과 Sawai의 방법에 의한 다층인식자의 훈련  
 (A: BEP 알고리즘 B: Sawai의 방법)  
 Fig. 7. Training of multilayer perceptron by BEP algorithm and by Sawai's method.  
 (A: BEP algorithm B: Sawai's method)

## V. 토의 및 결론

본 논문에서는 다층인식자의 인쇄체 한글 인식 능력 및 한계에 대하여 살펴 보았으며 그 실용가능성을 알아 보았다. 입력으로는 전처리과정에서 한글자체가 지나는 여러 방향의 DC 성분을 추출하여 사용하였으며 출력노드는 한글의 특성에 알맞도록 구성하였다. 이러한 신경회로망의 구성은 신경회로망의 훈련을 용이하게 만들었다.

정위치에 있는 문자에 대하여 신경회로망의 한글 인식정도, 중간층 노드의 개수가 전체 인식과정에 미치는 영향, 최소오차 도입을 통한 학습시간의 감소법 등에 관하여 알아 보았다. 한글 2,350자 중에서 2,337자까지 인식에 성공하였으며 훈련을 계속한다면 인식 글자수는 더 증가할 것이다. 입력문자에 좌우 변환이 있는 경우에 대하여도 인식을 시도하여 보았는데 변환이 심한 경우에는 그다지 만족스러운 결과를 얻지 못하였다. 그에 대한 대책으로는 우선 중간층 노드 개수의 증가를 통한 신경회로망의 크기 확대를 생각할 수 있지만 이는 훈련시간의 연장이라는 문제를 불러 일으킨다. 또 여기서는 거의 변형이 없는 다층인식자 구조를 사용하였는데 이것에 약간의 구조 변형을 가한다면 인식률을 증가시키는데 도움을 줄 수 있을 것으로 생각된다. 그밖의 다른 방법으로는 한글 자체가 지나는 위상적 특성을 인식에 이용하는 것으로서<sup>[7]</sup> 전체 글자 세트를 몇개로 분할함으로써 훈련시간의 감소를 가져 올 수 있고, 인식률의 향상을 추구할 수 있다는 면에서 유용한 방법으로 생각되는데 이 과정을 신경회로망의 구조에 포함시키는 방법을 연구중이다.

다층인식자는 입력이 24×24인 한글 문자에 대해서 전처리과정 및 본 논문에서 제시한 구조를 가질 때 좋은 인식 능력을 가지고 있으므로 한글문자 인식에 유력한 신경회로망 구조라 할 수 있으며 앞에서 제기된 문제점들은 본 논문에서 제시한 다층인식자를 기초로 하여 좀더 연구하면 다 해결될 수 있을 것이다.

## 參 考 文 獻

- [1] 이주근, "한글 문자의 인식에 관한 연구(IV)," 전자공학회지, 제9권 제4호, pp. 25-32, 1972년 9월.
- [2] T. Agui, A. Nakajima, Tae K. Kim, and E.T. Takahash, "A method of recognition of Korean characters by tree grammars," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, pp. 245-251, July 1972.
- [3] 김태균, 安居院猛, 中嶋正之, "Stroke 조합에 의한 필기체 한글의 표현과 인식," 전자공학회지, 제25권 제1호, pp. 18-26, 1988년 1월.
- [4] 이의동, 김태균, 安居院猛, 中嶋正之, "확장된 DP matching법에 의한 한글 온라인 인식," 전자공학회지, 제26권 제1호, pp. 26-36, 1989년 1월.
- [5] 오길남, 신승호, 진용욱, "2차원 디지털필터링에 의한 한글 자모의 인식 알고리즘," 전자공학회지, 제21권 제3호, pp. 55-59, 1984년 5월.
- [6] LQ 1550-850 User's Manual, Tri-gem, 1987.
- [7] 고병기, 김진형, 양현승, "경쟁하는 신경망 조직을 이용한 인쇄체 한글 문자의 인식," 제1회 영상처리 및 이해에 관한 워크샵 발표 논문집 pp. 127-134, 1989.
- [8] 김상우, 전운호, 최종호, "신경회로망을 이용한 한글문자의 인식," 학계학술대회논문집, 대한전기학회, pp. 166-169, 1989.
- [9] D.E. Rumelhart and J.L. McClelland, *Parallel distributed processing*, vol. 1, vol. 2, MIT Press, 1986.
- [10] H. Sawai, A. Waibel, P. Haffner, M. Miyatake and K. Shikano, "Parallelism, hierarchy, scaling in time-delay neural networks for spotting Japanese phonemes/CV-syllables," *IEEE 3rd Int'l. Conf. on Neural Network*, vol. 2, pp. 81-88, 1989.

---

 著 者 紹 介
 

---



金 相 宇 (準會員)

1965年 8月 7日生. 1988年 2月  
 서울대학교 제어계측공학과 학사  
 학위 취득. 현재 서울대학교 제어계  
 측공학과 석사과정. 주관심분야는  
 신경회로망 등임.



全 允 豪 (準會員)

1966年 11月 14日生. 1989年 2月  
 서울대학교 제어계측공학과 학사  
 학위 취득. 현재 서울대학교 제어  
 계측공학과 석사과정. 주관심분야  
 는 Vision 및 신경회로망 등임.

崔 悰 鎬 (正會員) 第25卷 第2號 參照

현재 서울대학교 제어계측공  
 학과 교수