

# 통합 사용자 인터페이스에 관한 연구 : 인공 신경망 모델을 이용한 한글 필기체 On-line 인식

최 정훈<sup>✉</sup>, 권 희용, 황 희영  
서울 대학교 컴퓨터 공학과

## A Study on the Intelligent Man-Machine Interface System : On-Line Recognition of Hand-writing Hangul using Artificial Neural Net Models.

Choi Jeonghoon, Kwon Heeyong, Hwang Heeyeung  
Dept. of Computer Engineering, Seoul National University

### 요 약

본 논문에서는 Error Back Propagation 학습을 이용해 한글 문자를 On-Line 인식 하는 시스템을 제안한다. Pointing device의 궤적을 추적해 입력 패턴의 특징(feature)을 추출해 신경 회로망 입력으로 준다. 이때 사용하는 특징은 기본 획(stroke)의 종류 및 획간의 상대적 위치 관계이다.

학습과정에서는 자소의 정의를 읽어 초성, 중성, 종성에 대해 각 획수마다 정의된 신경회로망의 weight를 조정한다. 인식 과정에서는 초성, 중성, 종성의 순으로 에러가 최소인 획수의 신경 회로망 출력을 택하여 2 바이트 조합형 코드로 완성한다. 이로써 Intelligent Man-Machine Interface 시스템중 위치 및 크기에 무관한 전필 입력 시스템을 구현한다.

### I. 서론

최근 보다 자연스런 Man-Machine 인터페이스에 대한 요구로 문자 및 음성 입력에 대한 연구가 한글에 대해서도 폭넓게 행해졌다. 한글 문자 인식의 연구 방향은 OFF-LINE 인식과 ON-LINE 인식으로 크게 구분할 수 있다.

OFF-LINE 인식은 카메라, scanner, OCR(Optical Character Reader) 등으로 digitize된 영상에 잡음(noise) 제거, 세선화(thinning), 윤곽선(contour) 추적, edge detection 등의 전처리 과정이 선행된 후에야 비로소 인식과정에 들어 간다. [1,2,3,4,5]

이에 비해 ON-LINE 인식은 이미지 프로세싱에 의한 전처리 과정이 필요없이 light pen, tablet, 마우스 등의 pointing device 에서 입력되는 좌표의 변화를 추적해 실시간 안에 인식결과를 보여야 한다. [5,7,8,9,10,11,12]

기존의 인식 방법은 표준 패턴과의 직접 비교(template matching), 입력 패턴의 통계적 분류(stochastic method), 문법에 의한 구분 분석(syntactic method) [1,2,3,6,7,8], 사전(dictionary) 및

table 탐색 [4,9,10], Dynamic Programming [11] 등의 방법이 사용되어 왔다. 최근 신경 회로망(Neural Network)이 패턴 분류에 적합하다는 인식하에 이를 이용한 인식이 시도되고 있다. [5,12]

입력된 좌표들에서 특징점만을 추출해 기본획(stroke)을 결정하고, 위치 정보를 이용해 기본획의 배열만으로는 구분할 수 없는 애매성을 해결한 예 [3,9] 가 있으나 이는 table 탐색이나 문법에 의한 인식 방법을 채택하고 있다. 신경회로망을 이용한 인식의 예로는 마우스 입력에 대해 동물의 시신경 구조를 본따 획 분석, 자소 분리, 문자 인식의 전과정에 걸쳐 신경회로망만을 이용하거나 [12] 스캐너 입력에 대해 6가지 한글 문자 기본 형태로 분류하는데 그친 예가 있다. [5]

본 논문은 IMMI(Intelligent Man-Machine Interface) 프로젝트의 일환으로서, Error Back Propagation 알고리즘을 통해 학습된 신경 회로망을 이용하여 pointing device를 통해 입력된 한글 필기체를 on-line 인식하는 시스템의 구현을 보이고 있다. 실제 실험에서 사용된 입력 장치는 마우스이다. 기본 획 종류와 획간의 상대적 시작위치를 이용해 각 자소의 모양을 정의하고 Generalized Delta Rule을 이용해 학습시킨다. 획 분석 과정에선 기존의 방법을 이용해 학습에 사

용된 중간 코드 형태를 생성한다. 이 중간 코드는 자소 분석에 필요한 입력 패턴의 특징을 기본획 종류와 획간 상대 위치 정보로 표현한다. 인식과정에서는 이 중간 코드를 입력으로 받아 학습 과정에서 조정된 weight 화일을 읽어 인식을 행한다.

우리의 획 분석 방법은 기존의 논문 [6,7,10,11]에서도 지적했듯이 한글 문자 인식을 위해서는 획의 종류 뿐아니라 위치 정보가 있어야 완전할 수 있다는 점에 착안했다. 그러나 기존 방식에서는 이들 모두를 수용할 경우, 문제의 크기 및 complexity의 과도한 증가로 부분적인 수용으로 해결하여 왔다. 그러나 우리는 신경회로망을 사용함으로써 위치 정보와 획의 종류에 관한 정보를 완전히 수용하여 인식에 완전을 기할 수 있게 되었다. 따라서 본 시스템은 다음과 같은 장점을 가진다.

1) 문법에 의한 syntactic 방법의 생성 규칙(production rule)이 복잡하고 수정이 어려운데 비해 규칙적으로 자소를 정의 함으로써 수정이 용이하다. 또한

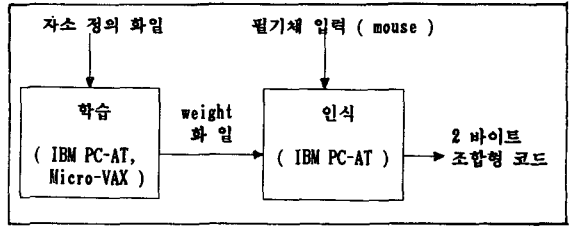
2) syntactic 방법에서는 생성규칙에 정의 되지 않은 입력은 결코 인식할 수 없지만 신경 회로망의 특징상 정의되지 않은 변형도 일부는 인식할 수 있다.

또한 순수한 신경 회로망으로 획분석까지 수행하는데 비교하면, 획 분석 과정에서 비교적 단순한 알고리즘을 사용해 어느 정도 위치나 크기의 변화를 허용한다는 장점을 들 수 있다.

## II. 신경 회로망에 의한 학습과 인식

### 1. 시스템 구성

본 시스템은 학습부와 인식부의 2 단계로 구성된다. 학습부의 입력은 기본 stroke의 종류와 상대적 시작 위치로 표현된 자소 정의 화일로 에러가 한계값보다 작아질 때까지 weight를 조정해 화일로써 출력한다. 인식부는 마우스의 궤적을 추적해 각 stroke마다 시작점, 중간점, 끝점을 통해 기본 획의 종류를 결정한다. 이진 획(stroke)과의 상대 위치와 결정된 stroke 종류를 학습에 사용된 자소 정의 형태의 중간 코드(특징)로 변환해 신경 회로망의 입력으로 준다. 학습된 weight 화일을 읽어 초성, 중성, 종성의 순으로 각각 에러가 최소인 자소를 선택해 2 바이트 조합형 한글 코드를 완성한다.



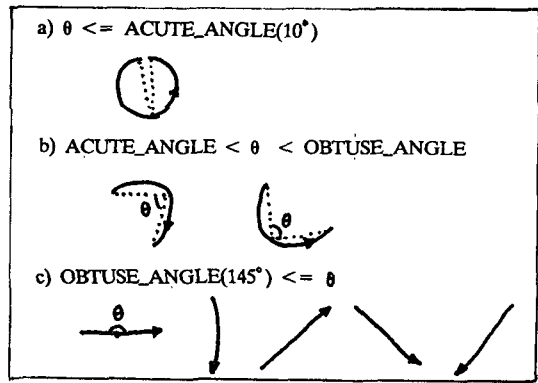
<그림 1> 시스템 구성

### 2. 자소 형태 정의

학습 과정의 입력은 각 자소의 모양을 정의한 화일이다. 자소의 모양은 8가지 기본 stroke(획) 모양과 8가지 상대적 시작 위치로 표현된다.

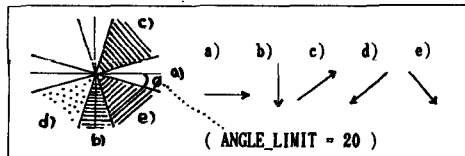
#### 1) 기본획

시작점( $X_1, Y_1$ ), 중간점( $X_2, Y_2$ ), 끝점( $X_3, Y_3$ ) 사이의 각도로 크게 3 부류로 분류된다.



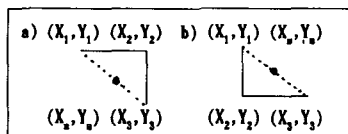
<그림 2> 중심각에 따른 기본획의 3종류

다시 끝점, 시작점간의 기울기를 통해 c)의 5패턴을 분류한다.



<그림 3> 기울기에 의한 직선획의 분류

b)의 2패턴은 중간점에 대한 시작점, 끝점의 중점( $X_m, Y_m$ ) 위치로 분류된다.



<그림 4> 중간점에 대한 중점의 상대 위치에 의한 곡선획의 분류

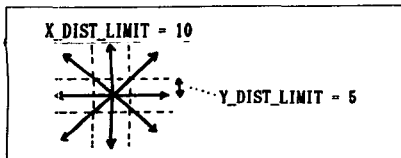
이들 8 개의 기본획에 대해 <표 1>처럼 기호 이름과 이진 숫자를 배정한다.

기본획	기호 이름	이진 숫자
0	OO	0100
ㄱ	GG	1110
ㄴ	NN	0111
→	RR	1000
↓	DD	0010
↗	RU	1101
↘	LD	0001
↖	RD	1011

<표 1> 기본획의 분류

2) 상대적 시작 위치

이전 stroke의 끝점을 ( $X_{old}, Y_{old}$ )로 유지해 현재 stroke의 시작점 ( $X_1, Y_1$ )과 비교해 상대 위치를 판정한다. 일정 pixel갯수를 초과하지 않아야 수직, 수평에 포함된다. 화면의 X, Y 축의 해상도가 다르므로 X, Y 축에 대해 서로 다른 한계값을 준다. 초성의 시작위치는 편의상 오른쪽 위로 하는 데 이는 연속 필기의 경우 오른쪽 위부터 새로운 글자가 시작되기 때문이다.



<그림 5> 상대 위치 분류 전략

이들 상대 위치에 대해 <표 2>처럼 기호 이름과 이진 숫자를 배정한다.

기본획	기호 이름	이진 숫자
↑	UU	1110
↗	RU	1101
→	RR	1000
↘	RD	1011
↓	DD	0010
↙	LD	0001
←	LL	0100
↖	LU	0111

<표 2> 상대 위치의 분류

3) 기본 자소의 분류

학습은 초성, 중성, 종성을 각 획수별로 각각 수행한다. 이는 인식 과정에서 초성, 중성, 종성의 순서로 가장 최소 에러를 갖는 획수를 찾아 해당 신경 회로망의 출력을 얻기 위함이다. 이에 따라 부분적인 수정시 해당 신경 회로망만 학습시키면 된다. 19 초성, 21 중성, 27 종성에 대해 획수로 분류한 예가 <표 3>에 있다. 총 17 개의 신

경 회로망이 사용된다. 이들의 기본적인 구조는 모두 같다. 각 획마다 종류 및 위치 정보를 표현 하는데 8비트를 이용하고 최대 획수는 8획('ㅁ')이므로  $8 \times 8 = 64$  개의 입력 노드를 가진다.

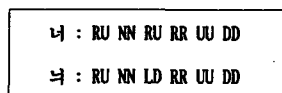
	초성	중성	종성
1	ㄱ ㄴ ㄷ	ㄱ ㄴ ㄷ	ㄱ ㄴ ㄷ
2	ㄷ ㅌ ㅍ	ㅌ ㅍ ㅊ	ㅌ ㅍ ㅊ
3	ㄷ ㅌ ㅍ	ㅌ ㅍ ㅊ	ㅌ ㅍ ㅊ
4	ㅂ ㅅ ㅈ ㅊ	ㅅ ㅈ ㅊ	ㅂ ㅅ ㅈ ㅊ
5		ㅈ ㅊ ㅋ	ㅂ ㅅ ㅈ ㅊ
6		ㅈ ㅊ ㅋ	ㅂ ㅅ ㅈ ㅊ
7		ㅈ ㅊ ㅋ	ㅂ ㅅ ㅈ ㅊ
8	ㅁ		ㅂ ㅅ ㅈ ㅊ

<표 3> 획수 및 초중성에 따른 기본 자소 분류

다른 기본 자소 분류의 예는 중성을 평모음, 종모음, 이들의 복합모음으로 나눈 경우 [5], 평모음, 복모음으로만 나눈 경우 [10], 복모음, 복자음 전부를 기본 자소에서 제외시킨 경우 [9], 평모음과 종모음으로 결합된 복합 모음만 기본 자소에서 제외시킨 경우 [7,11] 등을 들 수 있다. 이 기본 자소의 분류는 인식 대상이 되는 기본 자소의 범위 및 인식 순서에 영향을 준다. 즉 복모음, 복자음을 기본 자소에서 제외시킨 경우는 인식된 기본 자소를 결합해 복합 자소로 인식할 수 있다. 이처럼 초중중성으로 기본 자소를 정의함으로써 인식과정의 초성, 중성, 종성 인식의 3단계로 단순화된다.

4) stroke 조합에 의한 문자 표현과 인식

이와 같이 stroke마다 종류와 이전 stroke 끝점과의 상대적 시작 위치로 한 문자의 모든 stroke를 표현한 예를 <그림 6>에 보인다. 이 예는 애매(ambiguous)할 때만 그 획에 한해 위치 정보를 고려한 기법에서 해결하지 못한 애매성을 우리의 표현법으로 해결한 예이다. [7] 이러한 표현에서 초성, 중성, 종성의 순으로 최소 에러를 갖는 획수의 신경 회로망 출력을 취한다. 이때 복합 자소의 경우를 위해 가능한 한 길이가 긴 쪽으로 인식이 이루어지도록 한계값(UPDATE\_LIMIT)을 주었다. 에러가 이 한계값보다 작은 것중 가장 길이가 긴 것을 선택함으로써 복합 자소를 인식한다.



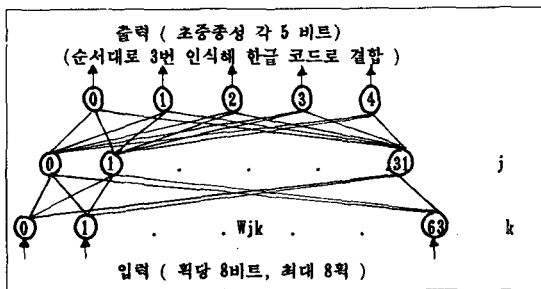
<그림 6> stroke 종류와 상대 시작 위치에 의한 문자 표현

다른 표현법의 예로는 흘림체를 인식하기 위해 곡선의 기본획을 삭제한 경우 [11], 한 자소를 흘려 쓰는 변형을 고려해 여러 개의 기본획을 첨가한 경우 [6,11], 필요할 때만 여러가지 많은 위치 정보를 고려하는 경우 [7] 등을 들 수 있다.

<표 1>에서 정의한 기호 이름은 자소 정의 및 마우스 입력 변 환의 출력으로 사용된다. 이를 배정한 이진 숫자로 치환해 학습 및 인 식에 사용한다. 이진 숫자의 배정은 각각 3 비트로 가능하지만 가능한 한 패턴간 비트 차이를 많이 줌으로써 independency를 증가시켜 분 류율을 향상 시킬 목적으로 4비트씩 배정했다. 이들의 비트 차이는 모 두 2비트 이상이다. 이진 숫자의 배정이 인식률에 미치는 영향을 테스 트해 보다 적합한 조합을 구할수 있을지도 조사할 예정이다.

### 3. 학습 알고리즘

우리의 학습 방식은 Generalized Delta Rule에 근거한 Error Back Propagation 방식이다 [13]. 이는 출력 노드에서 에러를 계산해 에러를 줄이는 방향으로 weight를 조정하는 Supervised Learning으로 학습시 target vector,  $d_j$ 를 주어 에러를 계산할 때 사용한다. 본 시스템 에서 사용한 신경회로망은 기본 구조가 3-layer Perceptron인 17개로 그중 최대 크기인 구조가 <그림 7>에 나와 있다. 이보다 확수가 작 은 자소의 신경회로망은 <그림 7>의 구조에서 좌측의 일부를 사용 한다. 각 픽마다 모양과 시작 위치를 각각 4비트씩 배정했으므로 픽당 8개의 입력 노드가 필요하다. 최대 확수는 8픽이므로  $8 \times 8 = 64$ 개가 입력 노드의 최대수가 된다. 출력 노드의 경우는 한글 조합형 코드가 초중중성에 각각 5비트씩 필요하므로 5개로 한다. 중간 노드의 갯수는 적당히 결정한다.



<그림 7> 3-layer Perceptron : 64 x 32 x 5

### 4. 인식 알고리즘

마우스 제적을 추적해 학습에 사용한 자소 정의 형태의 중간 코 드를 생성한다. 최소한 1픽의 중성을 포함하므로 (전체 픽수 - 1)의 길이까지 가능한 초성의 패턴당 에러(pss)를 계산한다. 가장 에러가 적은 픽수인 초성의 신경회로망 출력을 취한다. 인식할 다음 시작 위 치를 초성의 길이만큼 빼 중성에 대해 같은 방식으로 인식한다. 남은 길이가 0이 아니면 중성이 존재하는 것으로 남은 길이에 해당하는 중 성 신경회로망의 출력을 취한다.

1. 초기화  
입력의 작은 weight 값으로 초기화 - Symmetry Breaking
2. 학습 :  
- 패턴마다 각 노드의 출력을 계산 (forward computing)  
입력 노드 :  
 $X_{pj} = \text{INPUT}_{pj} \quad (p : \text{pattern}, 0 \leq j \leq 63)$   
기타 노드 :  
 $\text{NET}_{pj} = \sum W_{jk} X_{pk}$   
$$X_{pj} = \frac{\text{NET}_{pj}}{1 + e^{-\text{NET}_{pj} \text{BIAS}}}$$
  
- error back propagate  
출력 노드 :  
 $\delta_{pj} = (d_{pj} - X_{pj}) X_{pj} (1 - X_{pj})$   
 $\text{tss} = \sum (d_{pj} - X_{pj})^2$   
중간 노드 :  
 $\delta_{pj} = X_{pj} (1 - X_{pj}) \sum \delta_{pk} W_{kj}$   
 $\Delta W_{jk} = \eta \delta_{pj} X_{pk}$
3. 학습 결과 저장 - 출력 화일  
초성 : f\_1\_wt, f\_2\_wt, f\_3\_wt, f\_4\_wt, f\_8\_wt  
중성 : m\_1\_wt, m\_2\_wt, m\_3\_wt, m\_4\_wt, m\_5\_wt  
중성 : l\_1\_wt, l\_2\_wt, l\_3\_wt, l\_4\_wt, l\_5\_wt, l\_6\_wt, l\_7\_wt

<그림 8> Error Back Propagation에 의한 학습 알고리즘

1. 입력  
pointing device(마우스) 입력을 중간 코드로 변환  
--> 버퍼 : in, 길이 : len  
화면에 제적 표시
2. 초성 인식  
e\_first ( in, len-1, o\_buf[0], &offset )
3. 중성 인식  
e\_middle ( in, len-offset/8, o\_buf[1], &offset )
4. 종성 인식  
e\_last ( in, len-offset/8, o\_buf[2], &offset )
5. 출력  
o\_buf[0], o\_buf[1], o\_buf[2] 를 THRESHOLD 값에 의해 0/1로 바꾸어 2바이트 조합형 한글 코드로 출력한다.

<그림 9> 인식 알고리즘

인식에 신경 회로망 모델을 사용해 인식 절차가 명료하고 단순해 졌다. 또한 초성에서 len-1 픽수로 인식을 시도해 '근'을 '르'로 인식하 는 애매성을 제거할 수 있다. 또한 약간 입력이 학습 패턴과 다르더라도 일부는 정인식하는 fault-tolerant(robust)한 특성이 있다. 이에 비 해 syntactic 방법에서는 모든 경우의 생성 규칙을 기술해야 하고, 기 술되지 않은 패턴은 결코 인식할 수 없다.

## III. 인식 실험 및 개선 방안

학습 과정은 IBM PC-AT, Micro-VAX 에서 구현했고 인식 과 정은 마우스를 이용해 AT에서 Turbo-C 2.0 으로 구현했다. 마우스 입력시 화면 모드는 640 x 200 pixel 의 해상도를 갖는 CGA 흑백 그래픽 모드로 했다. 이를 위해 Hercules 카드의 경우는 simcga를 번 지 실행시킨다.

이 회승서의 "언어 표현과 어감" 에서 255 자를 실험한 결과 182 자를 인식해 71 % 의 인식률을 보였다. 오인식의 이유는 (1) 변형

(variation)을 고려하지 않고 각 자소의 대표 패턴만을 학습한 점과  
(2) 현재의 coding 방식으로 해결하지 못한 애매성(ambiguity)으로 이  
들의 해결 방안을 제시한다.

### 1. 변형(variation) 의 해결

변형의 원인은 획수, 획순에 따른 변형과 한글의 6가지 기본형태  
에 따른 자소간 상대 위치의 다양성이다 <그림 10,11,12>. 이를 해  
결하기 위해서는 뒤에 2비트의 변형 비트를 더 두어 여러 변형을 표  
현하면 된다. 즉 출력 노드의 갯수를 7개로 늘리고 자소 정의의 target  
vector를 7비트로 표현하되 여러 변형에 대해 앞의 5비트만 유효한 코  
드값을 갖게 한다. 인식시에는 이 5비트만 출력으로 취한다. 실험 결  
과를 분석한 결과 이 부분만 수정해도 90 % 이상의 인식이 예상된  
다.

변형 문제를 해결하는 다른 기법들은 이런 많은 변형 가능성을  
문법으로 복잡하게 표현하거나, OFF-LINE 인식에서 글자의 정규화  
를 수행하는 등의 예가 있다.

ㅅ : RU GG RU RD (현재 학습 패턴)  
ㅆ : RU RR LL LD RU RD

<그림 10> 초성 'ㅅ'의 획수에 따른 변형

ㅅ : RU DD UU RR UU DD LL RR (현재 학습 패턴)  
ㅆ : RU DD RU DD LU RR LD RR

<그림 11> 초성 'ㅅ'의 획순에 따른 변형

크 : RU GG LU DD LL RR  
오 : RU OO DD DD LL RR  
노 : RU NN LL DD LL RR  
코 : RU GG LD DD LL RR (현재 학습 패턴)

<그림 12> 모음 '으'의 상대적 시작 위치에 따른 변형

### 2. 애매성(ambiguity) 의 해결

애매성의 원인은 다른 모음인데도 동일한 중간 코드를 갖는 경우  
와 중성이 초성의 일부로 인식되는 경우, 중성이 중성의 일부로 인식  
되는 경우등을 들수 있다. 첫번째는 현재의 coding 방법으로는 당연한  
결과이고 나머지는 복합 자소를 위해 최소 에러가 아니더라도 일정한  
한계(UPDATE\_LIMIT = 0.05)이하의 에러를 가지면 길이가 긴 자  
소쪽으로 인식하려는 시도 때문이다.

유 : RU OO LD RR LL DD RU DD  
위 : RU OO LD RR LL DD RU DD

<그림 13> 모음 'ㅠ'와 '위' : 동일한 중간 코드 가능

더 : RU RR LL NN LU RR RU DD  
티 : RU RR LL NN LU RR RU DD

<그림 14> 변형에 의해 중성 일부가 초성에 붙어 인식된 경우

갈 : RU GG LU DD UU RR LD RR LL NN  
간 : RU GG LU DD UU RR LD RR LD NN

<그림 15> 중성 일부가 중성에 붙어 인식된 경우

이의 해결안으로 이전 획의 끝점과의 상대 위치 뿐 아니라 시작  
점과의 상대 위치도 고려함으로써 획의 범위를 크게 벗어 나는지를  
체크한다. LD,LU,RU 패턴의 경우 이런 애매성이 일어났으므로 이  
를 Far\_LD, Near\_LD, Far\_LU, Near\_LU, Far\_RU, Near\_RU 로  
세분한다.

LD의 경우 시작점간의 상대 위치가 LD 이면 Far\_LD, DD 이  
면 Near\_LD 이다. LU의 경우는 상대 위치가 LU 이면 Far\_LU, UU  
이면 Near\_LU 이다. RU의 경우는 이것이 RR 이면 Near\_RU, RU  
이면 Far\_RU 로 한다. 기호 이름을 편의상 FD, LD, FL, NL, FR,  
NR 의 두 글자로 표현해 애매성이 제거된 예가 <그림 16>에 있다.  
이 방법은 비록 애매성을 해결하지만 변형을 증가 시키는 단점이 있  
다.

유 : RU OO FD RR LL DD NR DD  
위 : RU OO FD RR LL DD FR DD  
더 : RU RR LL NN FL RR FR DD  
티 : RU RR LL NN NL RR FR DD  
갈 : RU GG FR DD UU RR FD RR LL NN  
간 : RU GG FR DD UU RR NL RR FD NN

<그림 16> FD/ND, FL/NL, FR/NR 로 해결된 모호성

다른 방법으로는 항상 시작점간의 상대 위치를 표현할 수도 있다.  
즉 Far\_LD는 LD LD, Near\_LD는 DD LD, Far\_RU는 RU RU,  
Near\_RU는 RR RU로 2 상대 위치를 모두 표현하는 것이다.

모호성의 해결 방법으로 이전에 제안 된 것들은 길이가 길고 사  
용 빈도가 많은 것 [1], 시작점에 대한 이전 획 시작점, 끝점과의 상  
대 위치 및 획의 범위 [7], 획간 중심점의 상대 위치 [6] 등을 고려했  
다. [7]에서는 애매성이 발생할 경우에만 많은 위치 정보를 고려하고  
있고 우리는 항상 최소한의 위치 정보를 가지게 된다.

### 3. 기타 문제점과 해결 방안

1) 신경 회로망의 특성상 학습 시간이 매우 길다. 이를 위해 hidden node의 갯수를 약간 줄이고 ERR\_LIMIT를 약간 크게 해 시간과 인식률의 최적화한다.

2) 현재 학습시간 및 인식률에 영향을 주는 상수들을 변화 시켜 볼 필요가 있다. 이들은 학습 비율( $\eta$ ), ERR\_LIMIT, THRESHOLD, BIAS, UPDATE\_LIMIT등이다. 마우스 입력의 픽 분석에서도 ACUTE\_ANGLE, OBTUSE\_ANGLE, ANGLE\_LIMIT, X\_DIST\_LIMIT, Y\_DIST\_LIMIT 등이 있다.

3) 회전된 글자의 인식을 위해서는 특징점의 좌표를 회전각만큼 재배치한 후 인식한다.

4) 흘러 쓸 경우의 기본픽을 첨가한다.

5) 문자 입력 끝에서 인식을 시작하지 않고 새로운 픽이 입력될 때마다 인식을 시도해 일종의 한글 automata를 내장하면 연속 필기가 가능하다.

## IV. 결 론

본 시스템은 마우스의 제적을 추적해 각 픽마다 픽간의 상대위치 및 픽의 종류를 추출하여 신경회로망의 입력패턴 feature로 사용하였다. 신경회로망은 Error back propagation에 의한 Supervised learning된 3 layer perceptron 구조로 구성하였다.

이로써 문자인식에 있어서 문제로 되어왔던 크기 및 위치에 무관한 인식을 가능케 하였으며, 자소별 학습 방식을 취함으로써 학습량을 최소화하고 학습이 안된 변형된 입력에 대해서도 어느정도 인식이 가능하게 되었다.

본 시스템은 현재 여러가지 문제를 가지고 있지만 시스템이 개선됨에 따라 통합 사용자 인터페이스 시스템(Intelligent Man-Machine Interface System)의 일환으로서 점필(pointing device)에 의한 입력 시스템의 가능성을 보이고 있다.

## 참 고 문 헌

- [1] 김태균, Agui Takeshi, "Syntactic 범에 의한 한글의 패턴인식에 관한 연구", 전자공학회지 제 14권 5호. pp15-21. 1977.12
- [2] 유승필, 김태균, "속성문법을 이용한 필기체 한글 문서내의 자모인식", 전자공학회지 제 26권 3호. pp85-94. 1989.3
- [3] 이은주, 권오석, 김태균, "필기체 한글에서 자모분리와 인식

", 한국정보과학회논문지 Vol.15 No.6 pp526-534 1988.12

[4] 박종욱, 이주근, "Shape Pattern에 의한 필기체의 한글인식", 전자공학회지 제 22권 5호. pp1-9. 1985.9

[5] 이동현, 조원규, 양원승, 김진형, "신경망 모델을 사용한 한글 문자의 형태분류", 한국정보과학회 학술발표 논문집 Vol.16 No.1 pp215-218 1989

[6] 홍성민, 김명원, 박종국, 조원경, "구조해석에 의한 필기체 한글의 온라인 인식", 전자공학회지 제 23권 1호. pp114-119. 1986.1

[7] 이회동, 김태균, "보강문맥자유문법을 이용한 필기체 한글 온라인 인식", 전자공학회지 제 24권 5호. pp37-44. 1987.9

[8] 김태균, Agui Takeshi, Masayuki Nakajima, "Stroke 조합에 의한 필기체 한글의 표현과 인식", 전자공학회지 제 25권 1호. pp18-26. 1988.1

[9] 이은주, 김태균, "한글의 on-line 인식에 관한 연구", 한국정보과학회 가을 학술발표 논문집 pp278-284 1985

[10] 이은주, 김태균, "한글에 적합한 픽해석에 의한 연속 필기체 한글의 on-line 인식에 관한 연구", 한국정보과학회 논문지 Vol 15, No. 3 pp171-181 1988.6

[11] 이회동, 김태균, Agui Takeshi, Masayuki Nakajima, "확장된 DP 매칭법에 의한 흘림체 한글 온라인 인식", 전자공학회지 제 26권 1호. pp29-37. 1989.1

[12] 이일병, 정아연, "신경회로망 KoRecNet의 설계와 중간결과", 신경회로망 워크샵, pp77-85 1989

[13] James L. McClelland, David E. Rumelhart, "Parallel Distributed Processing", pp318-364, The MIT Press, 1986

[14] Richard P. Lippmann, "An Introduction to computing with Neural Nets", IEEE ASSP Magazine, April, 1987

[15] 김명원, 이광노, "Pattern 인식을 위한 Neural Network", 전자통신 제 11권 1호 pp41-56 1989.4