

계층적 신경망을 이용한 다중크기의 다중활자체 한글문자인식

Multi-font/multi-size Hangul Character Recognition with Hierarchical Neural Networks

권 재욱, 조 성배, 김 진형

인공지능연구실, 한국과학기술원 전산학과, 서울 청량우체국 사서함 150호

요약

본 논문에서는 인쇄체 한글문자를 실용적으로 인식하기 위하여 고안된 계층적 신경망을 소개하고, 이를 다중활자체의 한글문자를 인식하는 문제에 적용하였다. 이 신경망은 입력된 문자영상을 6가지의 유형으로 분류한 후, 해당 유형을 처리하는 신경망에서 실제 문자를 인식하도록 구성되었다. 또한 각 신경망을 모든 입력영상의 모든 출력노드에 대해 고르게 학습시키기 위하여 Backpropagation 알고리즘을 개선한 Descending Epsilon 알고리즘을 도입하였다. 그 결과 사용빈도수가 높은 한글 520자에 대해 94.4 - 98.4%의 인식률을 얻음으로써 본 논문에서 제안한 시스템이 다양한 활자체로 이루어진 실제 문서인식시스템의 문자인식부에 효과적으로 사용될 수 있음을 제시하였다.

I. 서론

문서자동입력 시스템의 필요성이 증가함에 따라서 그 핵심기술인 한글문자인식에 관한 연구가 활발히 진행되어 많은 시스템이 개발되고 있다. 그러나 이러한 인쇄체 한글 문자인식에 관한 대부분의 연구가 어떤 정해진 활자체와 크기에 대해 행하여진 반면에 실제의 문서는 여러 크기의 다양한 활자체로 구성되므로 이러한 연구결과를 실제 문서인식 시스템에 적용하는 것은 부적절하다. 따라서 실제 문서인식 시스템에 사용하기 위해서는 여러가지 크기의 다양한 활자체로 이루어진 한글문자를 인식하는 다중활자체 한글 문자인식 시스템을 개발할 필요가 있다. 본 논문에서는 인쇄체 한글문자를 인식하기 위하여 제안되었던 계층적 신경망 [6]을 기본으로 하여 신경망을 구성하고, 이를 다중활자체 인식에 적합하도록 Descending Epsilon 알고리즘 [11]등을 사용하여 학습시킨 다중활자체 한글인식시스템을 구현하였다.

본 논문의 II장에서는 다중활자체 한글의 특성과 이에따른 인식의 어려움 그리고 다중활자체 인식에 관한 기존의 연구에 대해 알아보고, III장에서는 다중활자체 한글문자를 인식하기 위한 계층적 신경망의 구조와, 신경망을 학습시키기 위해 사용한 알고리즘에 대해 설명한다. IV장에서는 III장에서 설명한 시스템을 이용하여 다중활자체 한글문자를 인식한 결과를 보이고, 어떤 경우에 오인식이 많이 발생하는가를 분석한다. 마지막으로 V장에서는 앞으로 계속 연구되어야 할 방향을 제시하는 것을 결론으로하여 끝을 맺는다.

II. 다중활자체 문자의 인식

문자인식에 있어서 가장 중요한 문제는 같은 문자에 대한 변형을 처리하는 방법이다. 단일활자체 문자인식의 경우 그 변형은 실제 문서 상태의 차이에서 발생하는 변형과 스캐너로 입력받을 때 나타나는 변형으로 제한되어 있다. 그러나 다중활자체 문자인식의 경우는 같은 문자에 대해서도 활자체 별로 형태와 크기가 서로 다르므로 단일활자체 문자인식과는 비교할 수 없을 정도로 많은 변형이 발생한다. 특히 한글의 경우에는 형태와 크기뿐만 아니라 한 문자내에서의 자소의 위치도 다르므로 더욱 많은 변형이 발생한다<그림 2.1>. 따라서 다중활자체 문자인식이 제대로 이루어지기 위해서는 이러한 변형을 모두 흡수하여 처리할 수 있어야 한다.

다중활자체 문자인식을 위하여 어떠한 방법이 연구되었는가를 알아보면, 영문자 인식의 경우 Pavlidis[9]의 연구와 Shlien[10]의 연구가 가장 널리 알려져 있다. 두 방법은 모두 입력영상으로부터 기하학적인 특성 및 위상학적인 특성을 추출한 후 이것으로 모델을 만들어 인식한 방법이다. 한자의 경우는 자획을 추출하여 길이에 따라 순서를 부여하고 전후 두 자획간의 위치관계에 의해 모델을 만들어 인식한 Wen-Hsing Hsu[8]의 연구 등 많은 연구가 있었으며, 다중활자체 한글인식의 경우에는 입력문자영상의 윤곽선을 분석하여 모음을 검출하고 검출된 모음에 따라 한글을 6가지 유형으로 분류한 후 입력된 문자를 자소별로 분리하여 인식한 이광호의 연구가 자소분리율 95%, 분리된 자소에 대한 문자인식률 96%를 얻었다.



<그림 2.1> 실제 입력된 입력문자의 예

III. 다중활자체 한글인식을 위한 신경망

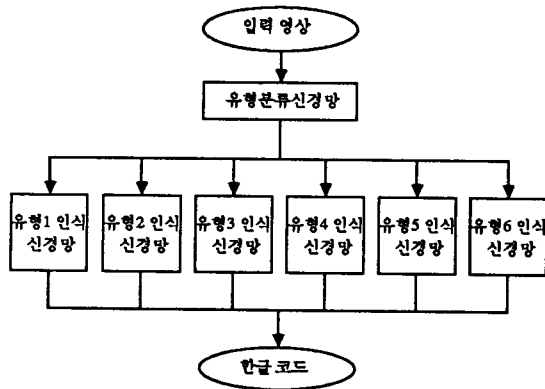
3.1 신경망의 구조

문자인식에 널리 쓰이고 있는 Backpropagation 신경망은 출력노드의 수가 많아짐에 따라 인식률이 저하될 뿐만 아니라, 신경망을 학습시키기 위해 더 많은 중간 노드를 필요로 하므로 학습시간이 많이 소요된다. 또한 하나의 출력노드에 여러 형태의 입력이 대응되면 여러 입력이 공통적으로 가지는 특성을 학습하기가 어렵다. 따라서 다중활자체 한글을 인식하기 위한 신경망은 구조적이고 계층적으로 구성되어야 한다.

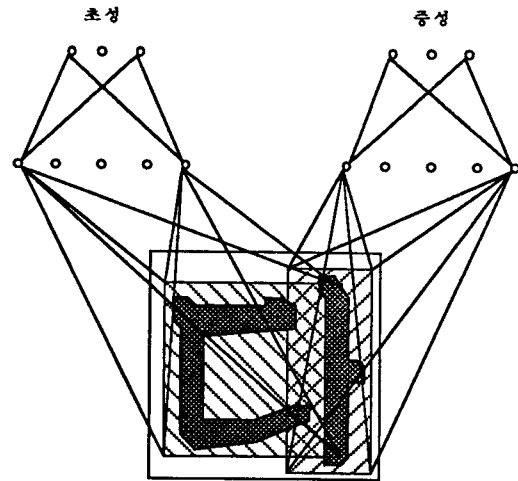
본 논문에서는 조성배 등[6]이 제안한 계층적 신경망을 기본으로 하여 신경망을 구성하였다<그림 3.1>. 이 방법은 신경망을 두 단계로 나누어, 먼저 유형분류 신경망이 입력된 문자를 6가지 유형 중의 하나로 분류하고 이에따라 유형별문자인식 신경망이 문자를 자소별로 인식하는 것이다. 유형분류 신경망은 한글의 구조적 특성인 모음의 형태와 받침의 존재 유무에 따라 한글의 유형을 분류한다. 한글의 유형을 분류하여 신경망에 적용한 이전의 연구에는 6가지로 분류하는 방법[2,5,6]과 15가지로 분류하는 방법[1]이 있었는데 유형분류 단계의 오분류는 후에 전혀 복구될 수 없으므로 본 논문에서는 유형분류율이 높은 6가지 유형으로 분류하는 방법을 선택하였다. 이와같이 유형을 분류하면 같은 출력노드에는 같은 유형의 입력 문자만 대응되므로 하나의 출력노드에 대응하는 입력의 형태를 감소시켜 신경망의 학습속도를 높일 수 있으며, 같은 유형의 문자에서는 각 자소가 유사한 위치에 놓이게 되므로 불필요한 입력을 제거하여 연결강도의 수를 줄일 수 있다.

유형분류신경망에서 분류된 문자는 그에 해당하는 유형별 문자인식신경망에서 자소별로 인식된다. 이 신경망에서는 각 유형별로 자소가 공통적으로 위치하는 영역을 결정하여 그 영역만을 신경망의 중간층과 연결시키는데<그림 3.2>, 본 연구에서는 각 활자체의 모든 경우를 포함할 수 있도록 충분히 큰 영역을 설정하여 중간층노드와 연결시켰다. 물론 이러한 방법으로 자소의 영역을 결정하면 원하지 않는 자소의 일부가 입력에 포함될 수 있지만 신경망은 단순히 결합극복능력을 가진 원형정합방법이 아니라 자체의 일반화 능력을 가지고 있으므로 이러한 문제를 해결할 수 있다.

신경망의 입력으로는 일반적으로 이진영상을 그대로 이용하거나 전처리에 의해 특징을 추출하여 사용한다. 그러나 전자의 경우 입력의 크기가 너무 커져 학습시간이 많이 소요되며 약간의 위치 변화가 인식과정에서 큰 영향을 미칠 수 있다는 단점을 가지고 있고, 후자의 경우에는 신경망전체의 성능이 특징추출 결과에 크게 의존하므로 신경망의 장점을 최대한 활용할 수 없다는 단점을 갖는다. 이와 같은 두 가지 면을 고려하여 본 논문에서는 입력의 수를 줄이면서 약간의 위치 변화를 흡수할 수 있고 또한 신경망의 성능이 입력의 표현에 의해 결정되지 않도록 하는 mesh 벡터를 입력으로 사용하였다.



<그림 3.1> 전체 시스템의 구성



<그림 3.2> 유형 1 문자인식신경망의 구성

3.2 학습 알고리즘

신경망을 학습시키기 위한 알고리즘으로는 기본적으로 Backpropagation 알고리즘을 사용하였다. Backpropagation 알고리즘은 각 출력노드에서의 실제 출력값과 원하는 출력값의 차가 최소화되도록 연결강도의 값을 조정하는 방법으로 현재 패턴인식 분야에서 가장 일반적으로 사용되고 있는 방법이다. 그러나 순수한 Backpropagation에 의해 신경망을 학습시키면 전체 입력문자와 학습 방향이 다른 소수의 입력이 제대로 학습되지 못하여 전체 신경망의 인식률을 저하시키는 원인으로 작용한다. 또한 <그림 3.3>와 같이 비록 출력노드 전체의 오차는 적은 값을 갖도록 학습되었다더라도 일부 출력노드의 오차는 큰 상태로 학습이 끝나는 수가 있다.

다중활자체 한글의 경우는 같은 문자에 대해서도 활자체 별로 각 자소의 형태, 크기, 그리고 문자내에서의 위치가 다르고, 스캐너로 문서를 읽을 때 발생하는 잡음 및 크기 정규화시의 변형 등 많은 변형을 가지고 있다. 따라서 하나의 출력노드에 대응될 입력이 상호 큰 차이를 보이는 경우 유사한 다른 자소를 해당 자소의 변형으로부터 확실히 구분할 수 있어야 한다. 또한 이러한 많은 변형 중의 일부가 학습되지 않은 입력으로 존재할 가능성이 높으므로 이를 처리할 수 있는 방법이 고안되어야 한다. 이러한 문제의 해결을 위하여 본 논문에서는 Descending Epsilon 방법[11]을 사용하였다. 이 방법의 핵심은 활성화되는 출력노드의 값이 다른 노드의 값보다 확실히 크게 하기 위하여 학습되지 않은 출력노드의 오차만으로 신경망을 학습시키는 것인데 신경망이 학습된 정도를 나타내는 척도로

$$E = \sum_{p=1}^P \sum_{i=1}^N (t_{p_i} - o_{p_i})^2$$

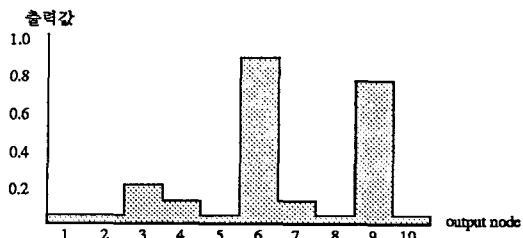
의 식으로 계산되는 오차를 사용하는 대신

$$E = \sum_{p=1}^P \sum_{i=1}^N (\text{number of } (|t_{p_i} - o_{p_i}| > \epsilon))$$

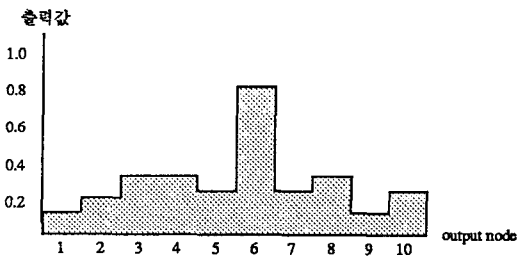
을 사용한다. 즉 실제 출력값을 이진 출력값으로 바꾸었을때, 원하는 출력값과 같은 값을 갖는 출력노드의 수를 사용한다.

Epsilon Descending 방법의 전체 알고리즘은 <그림 3.4>와 같다. 즉 어떤 epsilon값을 설정하여 각 출력노드에서의 오차가 이 값보다 작을 경우 오차를 0으로 만들어 이 출력노드의 오차가 학습에 반영되지 않도록 한다. 이 방법에 의해 신경망을 학습시키면 모든 활성화 출력노드와 비활성화 출력노드가 일정 범위내의 값을 가지므로 비활성화 되어야할 어떤 출력노드가 높은 출력값을 내는 것을 방지한다.

또한 각 입력에 대해 모든 출력노드에서의 오차가 현재 epsilon에 대해 0이 되면 그 입력은 학습에 참여시키지 않는다. 따라서 오차가 큰 입력에 대해 집중적인 학습이 이루어지므로 학습되지 않은 소수의 입력이 전체 인식률을 저하시키는 문제를 해결할 수 있다. 특히 이 경우 변화된 연결강도로 인하여 다음 epoch에서는 학습에 참가하는 입력문자의 집합이 변화하므로 선택적 재학습[6]에서 발생하는 학습된 입력문자와 학습되지 않은 입력문자의 혼합에 따른 문제를 자동으로 해결할 수 있다. 현재의 epsilon에 대해 모든 입력문자가 학습되면 epsilon을 감소시켜 다시 학습을 시키며 epsilon이 원하는 값까지 감소하면 학습을 마친다. 이 방법은 학습되지 않은 소수의 입력문자가 신경망의 인식률을 저하시키는 문제를 해결했고, 학습된 입력문자는 학습에 참여시키지 않으므로 학습시간을 줄일수 있다. 또한 활성화되는 출력노드와 비활성화되는 출력노드 간의 출력값의 차가 크도록 학습되므로 인식과정에서 다소의 변형을 흡수할 수 있고 비교적 높은 오차에서도 높은 인식률을 얻을 수 있다.

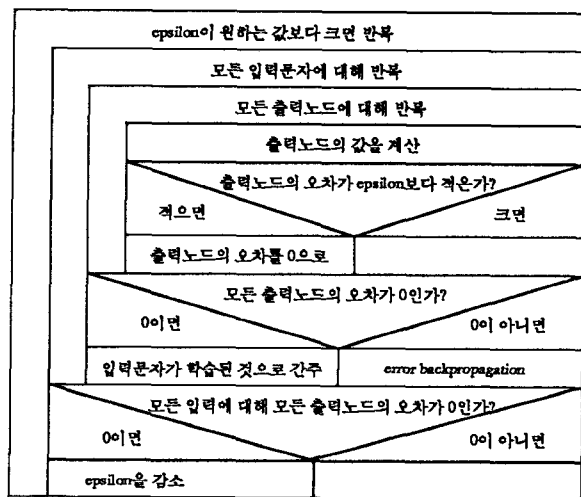


가> Backpropagation



나> Descending Epsilon

<그림 3.3> 신경망의 출력노드가 학습된 형태 (한 예)



<그림 3.4> Descending Epsilon Algorithm

IV. 실험 및 결과분석

4.1 실험 데이터

실험에 사용한 한글 문자는 한글 기계화 연구소의 조사에 의한 한글 짓기 순위별 1500자 [7] 중에서 상위 520자이다. 사용된 활자체는 고딕체, 명조체, 궁서체, 공작체, 디나루체이고 이중 고딕체와 명조체는 Qnix 레이저빔 프린터의 H4 크기와 H6 크기도 함께 사용했다. 실험은 300 DPI의 해상도를 갖는 광학문서 입력장치로 입력받은 영상을 60x60으로 정규화한 후 다시 12x12의 mesh로 만들어 사용하였다. 따라서 신경망의 입력값은 0 또는 1이 아니라 0에서 1 사이의 값을 가지게 된다.

4.2 실험결과와 분석

4.2.1 각 자소별 인식률

아래의 모든 활자체를 입력으로 하여 신경망을 학습시킨 후 각 자소별로 인식한 결과 <표 4.1>과 같은 인식률을 얻었다. 이 결과를 분석해 보면 초성과 중성의 인식률은 99%이상인데 비해 중성의 인식률이 상대적으로 낮아서 전체 인식률 저하의 가장 큰 요인이라 할 수 있다. 실제로 잘못 인식된 문자의 50.8%가 모음의 오인식으로 인한 것이었다. 특히 모음의 오인식 중 89.1%가 유형 2나 유형 5의 모음, 즉 'ㄴ', 'ㅇ', 'ㅏ', 'ㅑ', 'ㅓ'에서 발생했고 그 중 98% 이상이 'ㄴ', 'ㅏ', 'ㅓ'의 혼동에서 발생한 것이었다. 이러한 결과는 신경망의 학습과 인식과정에서 자소를 분리하지 않고 문자의 일정 영역을 입력으로 하여 처리하므로, '그'와 '오'에서와 같이 'ㄴ'과 'ㅇ'이 문자내에서는 유사한 형태를 가지고 있기 때문에 발생하는 것이다. 이러한 문제가 본 논문과 유사한 방식으로 한글을 인식할 경우 가장 시급히 해결해야 할 문제라고 생각하며 참고로 'ㄴ', 'ㅏ', 'ㅓ'의 오인식을 제외하면 전체의 인식률이 97 - 99%로 3 - 4%정도 높아진다.

이 밖에 입력으로 12x12의 mesh를 선택하므로써 중성이나, 유형 5 또는 유형 6의 초성에서 'ㄹ', 'ㅁ', 'ㅂ', 'ㅇ', 'ㅌ', 'ㅍ', 'ㅎ'의 구분이 어려워 진다는 문제점을 가지고 있다. 실제로 입력을 이진화하여 사람이 판단하더라도 구분이 어려울 정도이므로 이러한 문제점과 입력의 크기에 따른 학습 및 인식 속도, 그리고 위치 이동에 따른 변화 흡수 등을 고려하여 적절한 입력의 크기를 결정하는 것이 중요하다고 하겠다. 이러한 오인식이 중성 오인식의 90% 정도를 차지한다. 4.2.2 각 유형별 인식률

앞 절의 결과를 유형별로 분류해 보면 <표 4.2>와 같다. 실험에 사용한 520자를 유형별로 나누면 유형 1이 93자, 유형 2가 50자, 유형 3이 21자, 유형 4가 218자, 유형 5가 123자, 유형 6이 15자이다. 따라서 인식률에 큰 영향을 미치는 것은 유형 1과 유형 4, 그리고 유형 5인데 유형 1과 유형 4는 활자체별로 고른 인식률을 보이므로 결국 유형 5의 인식률이 전체 인식률을 결정한다고 볼 수 있다.

실제로 인식률이 낮은 공작체와 궁서체의 경우 유형 5의 인식률이 83.61%와 90.61%에

머물러 있어 다른 유형과 많은 차이를 나타냄을 알 수 있다. 이것은 앞절에서 유형 5의 모음 'ㄴ', 'ㄷ', 'ㄹ'의 혼동이 전체 인식률에 큰 영향을 미친다는 것을 더욱 잘 나타내주고 있다. 4.2.3 활자체별 인식률

앞절의 결과에서 볼 수 있듯이 공작체와 궁서체의 인식률이 다른 활자체에 비해 매우 낮다. 그 원인은 고딕체와 명조체에 대해서는 여러 크기의 입력문자를 학습시켰으므로 신경망이 이 두 활자체의 입력문자를 주로 학습하였기 때문일 것이다. 즉 학습기회가 적었던 디나루체, 공작체, 궁서체 중 문자의 형태와 자소의 위치가 고딕체나 명조체와 상이한<그림 2.1> 공작체와 궁서체의 인식률이 상대적으로 낮아진 것이다. 따라서 보다 효율적으로 다중활자체 한글을 인식하는 신경망을 학습시키기 위해서 각 활자체에 대해 입력의 빈도수를 고르게 유지하는 것이 필요하다.

	유형	초성	중성	종성	전체
디나루체	99.14	99.23	98.27	99.02	97.78
고딕체 (중)	99.71	99.62	98.36	99.58	97.98
고딕체 (대)	99.76	99.37	98.80	98.87	97.74
공작체	99.52	99.23	96.15	98.16	94.61
궁서체	99.42	98.65	96.82	97.04	94.41
명조체 (소)	99.33	98.27	97.50	98.04	95.76
명조체 (중)	99.81	99.27	98.94	99.16	98.36
명조체 (대)	99.66	99.42	98.27	99.09	97.64

<표 4.1> 자소별 인식률

	유형 1	유형 2	유형 3	유형 4	유형 5	유형 6	전체
디나루체	98.91	100	90.70	98.40	96.73	93.75	97.78
고딕체 (중)	98.93	99.00	100	97.95	96.31	100	97.98
고딕체 (대)	99.46	99.00	100	97.59	95.50	100	97.74
공작체	99.46	97.03	90.91	98.15	83.61	100	94.61
궁서체	95.63	96.04	86.36	96.09	90.61	100	94.41
명조체 (소)	97.80	97.96	95.35	95.20	94.33	96.77	95.76
명조체 (중)	99.47	100	100	97.93	97.14	100	98.36
명조체 (대)	98.92	99.50	97.59	97.49	95.91	100	97.64

<표 4.2> 유형별 인식률

IV. 결 론

본 논문에서는 인쇄체 한글문자를 인식하는 계층적 신경망을 실제 문서에 나타나는 여러 크기의 다양한 활자체 문자 인식에 적용하여 94% 이상의 인식률을 얻었다. 신경망의 학습 능력, 일반화 능력, 그리고 결함 극복 능력을 이용하여 다중활자체 한글의 많은 변형을 흡수할 수 있었으며, 특히 모든 입력영상의 모든 출력노드를 고르게 학습시키는 Descending Epsilon 알고리즘을 신경망의 학습에 사용함으로써 학습되지 않은 소수의 입력에 의한 학습 속도 및 인식률의 저하라는 문제를 해결하였다. 'ㄴ', 'ㄷ', 'ㄹ'의 혼동에서 발생하는 오차를 감소시킬 수 있는 방법을 개발하고, 실제 문서로부터 문자를 추출하는 전처리부와 한글사전 및 문법적 지식을 이용하여 오인식된 문자를 교정하는 후처리부를 설계하여, 본 논문의 문자인식 시스템과 결합한다면 기존의 연구보다는 한단계 더 실용적인 한글 문서인식 시스템에 접근할 수 있을 것이다.

앞으로 계속 연구해야 할 과제는 위에서 언급한 'ㄴ', 'ㄷ', 'ㄹ'의 혼동을 감소시킬 수 있는 방법의 개발과 신경망의 일반화 능력을 높일 수 있는 학습방법의 개발이다. 보다 실용

적인 문서인식 시스템의 개발을 위해서는 현재의 학습되지 않은 활자체의 입력문자에 대한 인식률 80 - 85%를 90% 이상으로 높일 수 있는 신경망의 일반화 방법에 대한 연구가 필수적이다.

참고 문헌

- [1] 김 병기, 김 항준, "신경망 모델을 이용한 한글 문자의 형태 분류와 인식," 한국정보과학회 추계 학술발표 논문집, 서울, pp. 303 - 306, 1989.
- [2] 이 광호, "다중 활자체 한글 인식을 위한 자모의 분리," 한국과학기술원 전산학과 석사 학위논문, 1988.
- [3] 이 동현, 조 원규, "신경망 모델을 사용한 한글 인쇄체 문자의 형태 분류," 한국정보과학회 춘계 학술발표 논문집, 부산, pp. 215 - 218, 1989.
- [4] 이 승호, 조 창제, 김 일영, 조 성배, 양 현승, 김 진형, "한글 문서 인식 시스템 SIL NOON의 개발," 한국정보과학회 춘계 학술발표 논문집, 부산, pp. 211 - 214, 1989.
- [5] 이 주근, "한글 문자 인식에 관한 연구(IV)," 전자공학회지, 제 9 권 제 4 호, pp. 25 - 32, 1972.
- [6] 조 성배, 김 진형, "인쇄체 한글문자의 인식을 위한 계층적 신경망," 정보과학회논문지, Vol. 17, No. 2, pp. 306 - 316, 1990.
- [7] 한글 기계화 연구소 : 한글 기계화 연구, 1975.
- [8] P.N. Chen, Y.S. Chen and W.H. Hsu, "Stroke Relation Coding - A New Approach to the Recognition of Multi-font Printed Chinese Characters," Int. Jour. of PRAI, Vol. 1, No. 1, pp. 149 - 160, 1988.
- [9] S. Kahan, T. Pavlidis and H.S. Baird, "On the Recognition of Printed Characters of Any Font and Any Size," IEEE Trans. on PAMI, Vol. PAMI-9, No. 2, pp. 274 - 288, 1987.
- [10] S. Shlien, "Multifont Character Recognition For Typeset Documents," Int. Jour. of PRAI, Vol. 2, No. 4, pp. 603 - 620, 1988.
- [11] Y.H. Yu, R.F. Simmons, "Descending Epsilon in Back-Propagation : A Technique for Better Generalization," in Proceedings of the IEEE IJCNN, Vol. 3, pp. 167 - 172, 1990.