

論文91-28B-11-1

계층구조 신경망을 이용한 한글 인식

(Hangul Recognition Using a Hierarchical Neural Network)

崔 東 赫*, 柳 盛 元** 姜 賢 哲*** 朴 圭 泰*

(Dong Hyuk Choi, Seong Won Ryu, Hyun Chul Kang, and Kyu Tae Park)

要 約

본 논문에서는 한글 인식을 위한 적응적 계층구조 분류기를 제안하였다.

이 분류기는 USACL (unsupervised adaptive classifier) 과 SACL (supervised adaptive classifier) 두 개의 신경망으로 형성되어 있다. USACL은 입력단과 출력단으로 구성되며, 두 단 사이는 완전 연결되어 있다. 출력단의 노드는 unsupervised, nearest neighbor 학습 규칙에 의해 학습시 생성된다. SACL은 입력단 내부단, 출력단으로 구성되며, 입력단과 내부단 사이는 완전 연결되며 내부단과 출력단은 부분 연결되어 있다. SACL 내부단과 출력단은 부분 연결되어 있다. SACL 내부단과 출력단의 노드는 supervised, nearest neighbor 학습 규칙에 의해 학습시 형성된다. USACL의 역할은 pre-attentive 효과로 SACL 분류시 full-search 대신에 partial search로 처리속도 향상을 가져온다. USACL과 SACL의 입력은 방향성 에지에 방향성 수용장(receptive field)을 적용시킨 특징값이 된다.

계층구조 분류기의 성능을 테스트하기 위해, 다양한 한글 인쇄체로 학습하고, 테스트하였으며, 기존 nearest neighbor 학습 방식인 LVQ (learning vector quantizer)와 분류율 및 처리속도를 비교하였다.

Abstract

An adaptive hierarchical classifier(AHCL) for Korean character recognition using a neural net is designed.

This classifier has two neural nets: USACL (Unsupervised Adaptive Classifier) and SACL (Supervised Adaptive Classifier). USACL has the input layer and the output layer. The input layer and the output layer are fully connected. The nodes in the output layer are generated by the unsupervised and nearest neighbor learning rule during learning. SACL has the input layer, the hidden layer and the output layer. The input layer and the hidden layer are fully connected, and the hidden layer and the output layer are partially connected. The nodes in the SACL are generated by the supervised and nearest neighbor learning rule during learning. USACL has pre-attentive effect, which perform partial search instead of full search during SACL classification to enhance processing speed. The input of USACL and SACL is a directional edge feature with a directional receptive field.

In order to test the performance of the AHCL, various multi-font printed Hangul characters are used in learning and testing, and its processing speed and classification rate are compared with the conventional LVQ(Learning Vector Quantizer) which has the nearest neighbor learning rule.

*正會員, **準會員, 延世大學校 電子工學科
(Dept. of Elec. Eng., Yonsei Univ.)

***正會員, 仁川大學校 情報通信工學科
(Dept. of Info. & Telecomm. Eng., Incheon Univ.)

接受日字: 1991年 7月 2日

I. 서 론

본 연구에서는 복합 문자체 (multi fonts), 복합크기 (multi-size), 고속 처리를 대상으로 삼고, 알고리즘의 단순성, 병렬처리, 적응성, 학습 가능, 계층적 정보처리 능력을 갖는 모델을 연구하였다.

알고리즘의 단순성은 특징 추출시 mask에 의한 correlation을 사용하고, 패턴간 유사도 기준을 correlation이나 유클리디안 거리 (euclidian distance) 를 사용하므로써 얻어진다. 다음 신경망 이론의 적응성을 통계적 방식에 도입하므로써 병렬처리 학습이 가능하다. 계층적 정보처리는 인간의 시각 정보처리를 따르는 것으로 전분류 혹은 preattentive, 분류, 주의 (attention)의 계층으로 구성되며, 이는 입력 패턴의 변형을 부분적으로 흡수하고, 처리 정보량이 감소하며, 인식률의 증가를 가져온다.

본 논문에서는 적응적 노드 생성이 가능한 신경망 분류기를 계층적으로 형성하는 학습 알고리즘을 제안하여 계층 구조를 갖지 않는 supervised nearest neighbor 분류기인 LVQ (learning vector quantizer)¹⁾와 적응적 노드 생성은 가능하나 역시 계층 구조를 갖지 않는 논문²⁾의 AEDC (adaptive euclidian classifier) - 본 논문에서는 SACL (supervised adaptive classifier)로 명명함 - 와 분류율과 계산에 참여하는 노드수를 비교하였다.

II. 본 론

1. AHCL (adaptive hierarchical classifier)

Nearest neighbor 학습 방식은 supervisor, unsupervisor 두가지 모드로 구분할 수가 있으며, supervised nearest neighbor 학습 방법에는 여러가지가 있겠으나 본 논문에서는 두가지 LVQ, SACL (supervised adaptive classifier)만 고려한다. LVQ는 주어진 N개의 노드와 각 노드에 할당된 label에 따라 경쟁 학습에 의해 형성되는 분류기이다.³⁾ SACL는 LVQ에서 초기에 노드 갯수 N을 고정하는 방법 대신에, ART (artificial resonance theory)의 vigilance 개념을 도입하여 학습시 변형이 심한 패턴에 대해 적응적으로 노드를 형성하며, 경쟁 학습에 의해 형성되는 분류기이다.⁴⁾

LVQ, SACL 두 분류기는 분류시 모든 노드의 학습된 weight와 거리계산을 해야하며, 특히 한글의 경우와 같이 class의 갯수가 방대한 경우 실시간 처리에 제약이 된다. 따라서 위 두 신경망 분류기와 분류율은 비슷하면서도 거리 계산에 참여하는 노드수를 줄여야 하고, 이를 위해 전분류 혹은 pre-attention

기능을 가지는 unsupervised nearest neighbor 신경망 분류기인 USACL (unsupervised adaptive classifier)를 LVQ 혹은 SACL과 계층화시킨 적응 계층 분류기가 필요하다.

USACL은 ART1을 단순화 시킨 모델이다.⁵⁾ ART 1은 binary input을 bottom up correlation후 출력 노드를 선정한 다음, top down correlation값으로 vigilance parameter와 비교하여 새로운 노드 생성, 혹은 학습을 한다. USACL은 continuous 입력에 대하여 bottom up 유클리디안 거리로 매칭 후, top down 처리는 생략되며, 최소거리로 vigilance parameter와 비교하여 새로운 노드생성, 혹은 학습을 한다.⁶⁾

본 논문에서는 USACL를 LVQ 혹은 SACL과 결합해야 하는데 LVQ는 학습전 노드 갯수가 결정되어 학습중 노드가 생성되는 USACL 분류기와는 결합이 곤란하므로 학습중 노드가 형성되는 지도학습 분류기인 SACL로 USACL과 결합하는 AHCL (adaptive hierarchical classification)를 제안하였다.

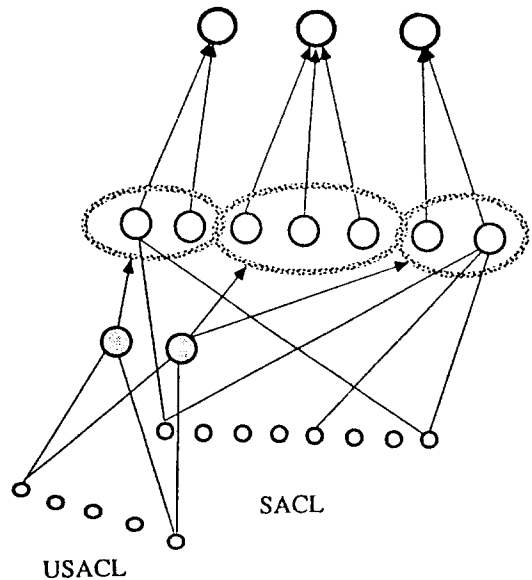


그림 1. AHCL의 구조
Fig. 1. Structure of AHCL.

그림1에서 보면 이 모델은 SACL옆에 USACL이 있으며 이는 분류시 USACL 출력노드가 선정되고 이와 연결된 SACL 내부 노드들만이 분류에 참여하게 된다.

이 논문에서 사용되는 표현의 정의는 다음과 같다.
 X_n : USACL의 입력특징, UDFH, UDFV의 일차원 확장

X : SACL의 입력특징, DFH, DFV의 일차원 확장

U : USACL의 입력단과 출력단간의 weight vector

M : SACL의 입력단과 내부단간의 weight vector

USACL-VP : USACL 노드확장의 기준값인 vigilance parameter

SACL-VP : SACL 내부노드확장의 기준값인 vigilance parameter

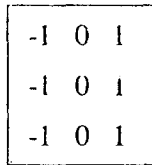
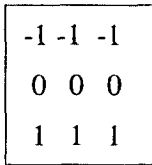
Nu : USACL출력 노드의 갯수

Ns : SACL 내부노드의 갯수

Tk : 노드k에서 학습된 특징 벡터 갯수

2. 특징 추출

USACL, SACL의 입력으로서 32×32 크기로 정규화된 문자 영상에 대해 다음과 같이 추출된 특징을 사용한다. 특징 추출방법은 방향성 edge 연산자와 방향성 receptive field의 적용이다. 본 연구에서는 2개의 분류기 즉, USACL과 SACL가 있다. USACL은 SACL의 내부노드를 선정하는 것으로 세부적 문자특성 보다는 전체적인 문자패턴 모양을 반영할 수 있는 특징이 바람직하다. 이러한 특징 추출은 receptive field의 크기를 확대함으로써 가능하며, 특징의 크기는 작아지게 된다. USACL의 특징을 다음과 같이 추출하였다.



수평 edge 연산자 H(i, j)

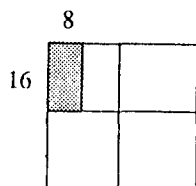
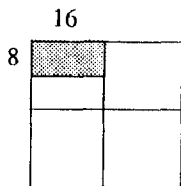
수직 edge 연산자 V(i, j)

$$F_H(x, y) = \left| \sum_{i=-1}^1 \sum_{j=-1}^1 H(i, j) * F(x-i, y-j) \right| \quad (1)$$

$$F_V(x, y) = \left| \sum_{i=-1}^1 \sum_{j=-1}^1 V(i, j) * F(x-i, y-j) \right| \quad (2)$$

$$UDFH_{ij} = \sum_{x=8+i}^{8+i+8} \sum_{y=16+j}^{16+j+16} F_H(x, y) \quad i=0, \dots, 4, j=0, \dots, 2 \quad (3)$$

$$UDFV_{ij} = \sum_{x=16+i}^{16+i+8} \sum_{y=8+j}^{8+j+8} F_V(x, y) \quad i=0, \dots, 2, j=0, \dots, 4 \quad (4)$$

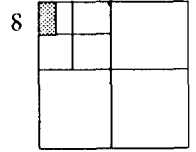
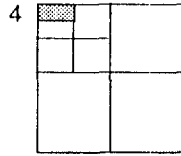


8*16수평방향 receptive field 16*8 수직방향 receptive field

SACL에서는 최종적으로 문자를 분류하므로, 보다 세분화된 특징을 추출한다.

$$DFH_{ij} = \sum_{x=4+i}^{4+i+4} \sum_{y=8+j}^{8+j+8} F_H(x, y) \quad i=0, \dots, 8, j=0, \dots, 4 \quad (5)$$

$$DFV_{ij} = \sum_{x=8+i}^{8+i+8} \sum_{y=4+j}^{4+j+4} F_V(x, y) \quad i=0, \dots, 4, j=0, \dots, 8 \quad (6)$$



4*8 수평방향 receptive field 8*4 수직방향 receptive field

USACL 입력 패턴벡터 **X_u**는 UDFHij의 일차원 확장이고, SACL 입력 패턴벡터 **X**는 DFHij, DFVij의 일차원 확장이다.

3. 학습과 테스트

AHCL의 학습은 USACL과 SACL을 결합한 것이며, STEP1-5까지의 USACL 학습과 STEP6-9까지의 SACL학습으로 구성되며, 주요사항은 STEP6에서 SACL 학습시 고려되는 SACL 내부노드는 USACL에 의해 선정된다는 것이다. 한글과 같이 클래스가 방대한 경우는 학습시간도 문제가 되므로 본 논문에서의 학습방식은 한번의 loop로 학습이 되므로 빠른 시간내에 학습이 완료되며, 알고리즘이 간단 명료하므로 구현이 용이하다!!

(AHCL의 학습)

초기화 : Nu=N_s=0

SACL출력노드 갯수=학습시킬 class갯수

Step1. USACL 입력 특징벡터. **X_u** 입력과 거리 계산

$$Dist_j^2 = \sum_{i=0}^M (x_{ui} - u_{ji})^2 \quad j=1, \dots, N_u \quad (7)$$

M=16, Nu은 흰 USACL출력 노드의 갯수

Step2. 최소거리 출력노드를 결정

$$Dist_k^2 = \min_j \{Dist_j^2\}, \quad j=1, \dots, N_u \quad (8)$$

Step3.

If $Dist_k^2 > USACL-VP$, step4. Else step5.

(9)

Step4. 새로운 USCAL 출력 노드 생성 (Nu증가), Step8으로 간다.

$$U_{Nu} = X_u \quad (10)$$

N_u 는 현재 USACL 출력 노드 갯수

Step5. 대표벡터 U 갱신. Step6으로 간다.

$$\Delta U_k = \frac{1}{T_k} (X_u - U_k) \quad (11)$$

Step6. 특징벡터 X 를 SACL에 입력, 거리계산

$$Dist_j^2 = \sum_{i=0}^N (x_i - m_{ji})^2 \quad (12)$$

, 여기서 j 는 USACL 출력노드에 의해 선정된 SACL 내부노드,

, $N=64$

Step7.

$$Dist_k^2 = \min_j \{Dist_j^2\} \quad (13)$$

If $Dist_k^2 > SACL-VP$, step8. Else step9. (14)

Step8. SACL의 내부노드의 생성 (N_s 증가) 및 출력노드와 연결, 현 USACL 출력노드와 연결, Step 1로 간다.

$$M_{Ns} = X \quad (15)$$

Step9. 대표벡터 M 갱신. Step1로 간다.

$$\Delta M_k = \frac{1}{T_k} (X - M_k) \quad (11)$$

<AHCL의 테스트>

Step1. X_u 를 USACL에 입력, 최소거리 출력노드 선정.

$$Dist_j^2 = \sum_{i=0}^M (x_{ui} - u_{ji})^2 \quad j=1, \dots, Nu \quad (17)$$

$$Disk_k^2 = \min_j \{Dist_j^2\}, \quad j=1, \dots, Nu \quad (18)$$

Step2. 선정된 USACL의 출력노드와 연결된 SACL의 내부 노드들을 선정

Step3. 입력 패턴 벡터 X 와 거리계산

$$Dist_j^2 = \sum_{i=0}^N (x_i - m_{ji})^2 \quad (19)$$

여기서 j 는 USACL 출력노드에 의해 선정된 SACL 내부노드

Step4. 최소거리 내부노드 선정, 이와 연결된 출력노드가 분류결과

III. 실험 및 결과

본 연구에서의 실험 데이터는 PostScript를 구현

한 Super Write 레이저 프린터 인쇄결과를 이용하였다. 이 데이터의 특성은 첫째, 한글 찾기순 900자로 구성되어 있으며, 둘째, 가로확대 세로확대의 크기 변화는 없다. 이유는 실제 한글문자에서 가로확대 세로확대는 자주 나타나지 않기 때문이다. 셋째, 3가지 글자폭 변화가 있다. 넷째, 10, 12, 14 point의 문자 크기 변화가 있다. 다섯째, 두가지 font가 있다. 여섯째, 3(글자폭 변화)*2(font 변화)*3(문자 크기 변화)=18, 해서 모두 18종의 데이터로 구성된다. 다음 이 중 6종으로 학습하고 12종으로 테스트 하였다. 300dpi Microtek-300A 스캐너로부터 이진 문서영상을 취하고, 문자를 추출한다. 추출된 문자를 32×32 크기로 정규화한 후 II.1에서 기술한 특징을 추출한다. 이 실험에서는 여러가지 USACL-VP, SACL-VP값에 따른 분류율을 얻었다.

1. LVQ, SACL 분류기 실험

먼저 AHCL의 성능을 LVQ, SACL과 비교하기 위해 한글 찾기순 200자에 대해 LVQ, SACL의 거리순 누적분류율을 구했다.

학습 데이터 (6개)	테스팅 데이터 (12개)	
M12.2. IMG	M10.1. IMG	M14.1. IMG
G12.2. IMG	G10.1. IMG	G14.1. IMG
M12.1. IMG	M10.2. IMG	M14.2. IMG
G12.1. IMG	G10.2. IMG	G14.2. IMG
M12.3. IMG	M10.3. IMG	M14.3. IMG
G12.3. IMG	G10.3. IMG	G14.3. IMG

데이터 당 문자갯수 : 200개

M: 명조체, G: 고딕체, 10

10, 12, 14: 문자크기.

1, 2, 3: 문자폭

그림1은 LVQ 노드수 400에서 LVQ 학습 반복 횟수에 따른 분류율이다. 반복횟수 30 이상에서는 수렴이 거의 완료되어 분류율에 변화가 없음을 알 수 있다. 표1에서는 LVQ와 SACL의 분류율을 노드수 증가에 따라 비교하였다. LVQ는 학습전 노드수를 결정해야 하므로 노드수를 클래스 갯수의 정수배인 400, 800으로 하였다. 반면 SACL은 SACL-VP값에 따라 생성되는 노드수가 틀리므로 400, 800에 비슷한 노드수를 발생시켜 비교하였다. 두개 분류기 동시 노드수 증가에 따라 분류율이 증가하며, 노드 수가 비슷한 경우 분류율은 비슷하며 후보 갯수가 늘어날 경우 SACL의 분류율은 보다 높다. LVQ는 30회 반복 학습을 실행한 반면, SACL은 한번의 학습으로 학습하였으므로 학습시간은 단축되고 분류율은 비슷

하다. 따라서 이 후의 실험은 모두 SACL과 비교한다.

SACL-VP는 SACL의 내부 노드 갯수와 관련이 있는데, SACL-VP의 값이 작으면 내부 노드의 수가 많아지고 크면 내부 노드의 수가 적어진다. 계산량은 노드의 갯수에 비례하고 이는 LVQ 또한 마찬가지이며, 위 실험의 경우 분류율과 계산 시간은 trade off이다.

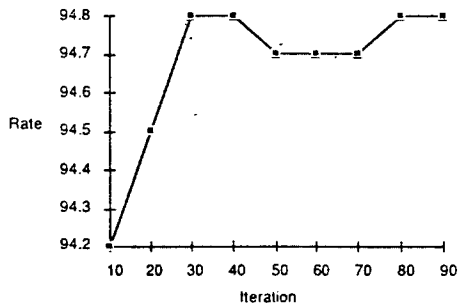


그림 2. 학습 반복 횟수 증가에 따른 LVQ분류율
Fig. 2. Classification rate of LVQ for the iteration increasing.

표 1. 12개 테스트 데이터에 대한 LVQ, SACL의 분류율

Table 1. Classification rate of LVQ and SACL for the 12 testing data.

	SACL-VP	Candidate Number				Nodes
		1	2	3	4	
SACL	100	97.4	99.7	99.8	99.9	973
	200	93.8	98.6	99.2	99.6	381

	Iteration	Candidate Number				Nodes
		1	2	3	4	
LVQ	30	94.8	96.7	97.8	98.4	400
	30	97.2	98.8	99.5	99.7	800

2. USACL의 실험

USACL-VP의 값을 50, 55, 60, 65, 70, 으로 변화시키며 실험하였다. USCAL에 의한 전분류는 분류해야할 클래스의 갯수가 적기때문에 분류율이 비교적 높고, 후보 갯수 2개에 대해서 누적 분류율을 구하였다. 학습 데이터 및 테스트 데이터는 III.1과 같다. 표2에서 USACL-VP가 작을수록 발생하는 출력 노드수는 많아진다. 이는 곧 USACL의 출력 노

표 2. USACL의 분류율
Table 2. Classification rate of USACL.

USACL-VP	Candidate Number		Nodes
	1	2	
50	94.6	98.9	140
55	95.0	98.9	92
60	95.4	99.5	67
65	93.5	99.0	45
70	93.5	98.6	35

드수가 많아 질수록 이후 SACL에 의한 최종 분류에서 계산에 참여하는 SACL 내부노드 수는 작아지고 계산량도 감소하게 된다.

3. AHCL의 실험

III.2의 실험에서 USACL-VP의 값은 60이 적절하다. 이 절의 실험은 USACL-VP=60, SACL-VP=100, 200, 300에서 수행하였고, 결과는 표3에서 보던 거리 계산에 참여하는 노드수가 대폭 줄었고 분류율은 낮아졌다. 학습 데이터 및 테스트 데이터는 III.1과 같다.

다음 표4는 SACL과 AHCL을 분류율, 내부 노드수에 대해서 비교하였다. 후보수의 의미는 그림3에 있다. 이 비교에서는 SACL의 후보갯수 1에 대해서는 AHCL 보다는 성능이 좋으며, 후보갯수 2이상에서는

표 3. AHCL의 분류율
Table 3. Classification rate of AHCL.

SACL-VP	Candidate Number				거리계산에 참여하는 노드 수
	1	2	3	4	
100	92.0	96.2	98.8	99.0	15
200	90.4	94.6	98.3	98.5	8
300	90.5	94.4	98.3	98.5	7

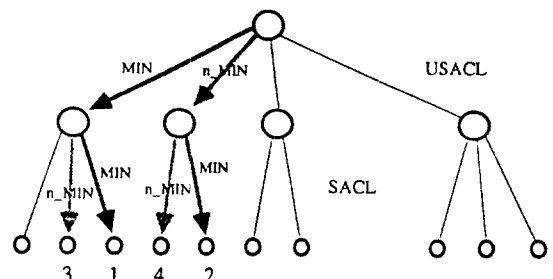


그림 3. 후보 갯수
Fig. 3. Number of candidates.

표 4. SACL과 AHCL의 비교
Table 4. Comparison of SACL and AHCL.

분류방법	VP	Candidate Number				거리계산에 참여하는 노드 수
		1	2	3	4	
SACL	100	97.4	99.7	99.8	99.9	973
AHCL	100	92.0	96.2	98.8	99.0	15
SACL	200	93.8	98.6	99.2	99.6	381
AHCL	200	90.4	94.6	98.3	98.5	8
SACL	300	90.7	97.0	98.3	98.7	202
AHCL	300	90.5	94.6	98.3	98.5	7

성능 차이가 적고, 계산량에 직접 비례하는 내부노드 갯수가 AHCL이 월등히 적다.

4. 학습 방법의 개정후 AHCL의 향상된 분류율
이 실험에서는 학습 방법을 다음과 같이 분류하였다. 이전 학습 방법은 USACL의 최소 거리를 가지는 출력 노드 한개만이 SACL의 내부 노드를 선정하나, 여기서는 최소 거리의 출력 노드 그리고 최소 거리 다음 출력 노드 두개가 SACL의 내부 노드를 선정해서 SACL를 학습한다. 이 방법은 SACL내부 노드들의 대표 벡터들의 중복을 어느 정도 허용해서 분류율을 높이고자 하는 것이며, 계산량은 중복된 대표 벡터만큼 증가하게 된다. 다음 표5는 SACL와 AHCL(1)(USACL 출력 노드 1개에 의한 학습), AHCL(2)(USACL 출력노드 2개에 의한 학습) 분류율, 내부 노드수에 대해서 비교하였다. 이 비교에서는

1) 계산량에 직접 비례하는 내부 노드 갯수가 AHCL(1)이 AHCL(2)의 약 반이 된다.

표 5. SACL, AHCL(1), AHCL(2)의 비교
Table 5. Comparison of SACL, AHCL(1) and AHCL(2).

분류방법	SACL-VP	Candidate Number				Nodes
		1	2	3	4	
SACL	100	97.4	99.7	99.8	99.9	973
AHCL(1)	100	92.0	96.2	98.8	99.0	15
AHCL(2)	100	95.8	97.1	99.5	99.7	30
SACL	200	93.8	98.6	99.2	99.6	381
AHCL(1)	200	90.4	94.6	98.3	98.5	8
AHCL(2)	200	93.5	94.9	99.3	99.4	17
SACL	300	90.7	97.0	98.3	98.7	202
AHCL(1)	300	90.5	94.6	98.3	98.5	7
AHCL(2)	300	93.7	95.0	99.0	99.2	15

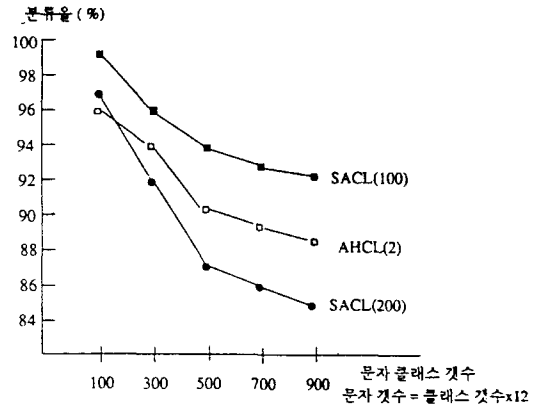


그림 4. 문자 클래스 갯수 증가에 따른 분류율
Fig. 4. Classification rate for the character class increase.

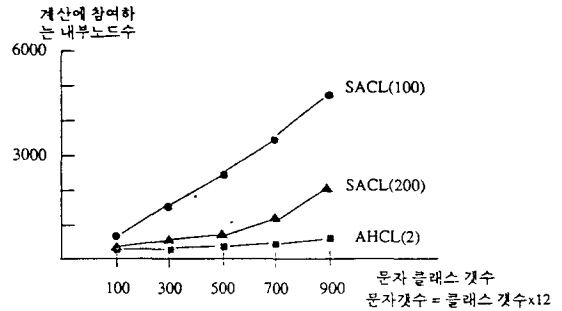


그림 5. 문자 클래스 갯수 증가에 따른 내부 노드수
Fig. 5. Number of nodes in the hidden layer for the character class increase.

2) 후보 갯수가 1일때 AHCL(2)가 AHCL(1) 보다 3-4% 높다.

5. 문자 클래스 갯수가 100, 300, 500, 700, 900인 경우

여기서는 III. 1의 데이터에 대해, 데이터 당 문자수를 100-900까지 증가시켜 가며, SACL, AHCL(2)의 분류율을 구해 보았다. SACL는 SACL-VP 값을 200, 300으로 실험하였으며, AHCL(2)는 USACL-VP = 65, SACL-VP = 250에서 실험하였다.

IV. 결 론

본 연구에서는 적응적 계층구조의 분류기 (AHCL : adaptive hierarchical classifier)를 고안했다. 6종류의 데이터로 학습하고 12종류의 데이터로부터 추출된 방향성 예지 특징으로 SACL와 AHCL의 성능을

테스트하였다. 또한 데이터를 구성하는 문자수를 100-900으로 증가시켜 실험하였다. 인식율은 900자인 경우 88%-92%이며, AHCL이 SACL 보다 2-3% 정도 떨어진다. 계산량은 AHCL이 SACL 계산량의 1/10-1/40 정도이고, SACL는 계산량이 문자수에 비례해서 증가하나 AHCL은 거의 일정하다. 이후 연구목표는 주의 집중 영역을 선정하여 feed-back에 의해 보다 세부적인 분류를 통한 인식률의 향상이다.

參 考 文 獻

[1] 최원호, 최동혁, 이병래, 박규태, "한글인식을 위한 신경망 분류기의 응용," 대한전자공학회 논문지, 제27호 제 8 호, 1990, pp. 93-103.
 [2] S. Grossberg, *Neural Networks and Natural Intelligence*, MIT press, 1988.
 [3] T. Kohonen, *Self-organization and Associative Memory*, Springer-Verlag, 1984.
 [4] T. Kohonen, "An introduction to neural computing," *Neural Networks*, vol. 1, pp. 3-16, 1988.

[5] P. K. Simpson, *Artificial Neural System* Pergammon Press, 1990.
 [6] Y.H. Pao, *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, 1989.
 [7] R.O. Lippmann, "An introduction to computing with Neural Nets," *IEEE ASSP Magazine*, pp. 4-22, April 1987.
 [8] H. Frohn, H. Geiger, and W. Singer, "A Self-Organizing Neural Network Sharing Features of Mammalian Visual System," *Biol. Cybern.*, pp. 333-343, 1987.
 [9] Y. Mori, K. Yokosawa, "Neural Networks that learn to discriminate similar Kanji characters," in *Advances in Neural Information Processing Systems I.D.* Touretzky ed., Morgan Kaufmann Pub., pp. 332-339, 1989.
 [10] J.T. Tou and R.C. Gonzalez, *Pattern Recognition Principles*, Addison-Wesley, 1974.
 [11] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, Wiley-Interscience, 1973.

著 者 紹 介

崔 東 赫 (正會員) 第26卷 第12號 參照
 현재 연세대학교 대학원 전자공학과 박사과정

姜 賢 哲 (正會員) 第26卷 第12號 參照
 현재 인천대학 정보통신공학과 조교수



柳 盛 元 (準會員)
 1968年 8月 20日生. 1991年 2月 연세대학교 전자공학과 졸업. 현재 연세대학교 전자공학과 공학석사. 주관심분야는 패턴인식, 영상처리 등임.

朴 圭 泰 (正會員) 第26卷 第12號 參照
 현재 연세대학교 전자공학과 교수