

신경 회로망과 퍼지 추론을 이용한 다중 특징 문자 인식 방법

최 원 호
전자공학과

<요 약>

본 논문에서는 문자 인식을 위하여 mesh와 방향성 특징의 두 종류를 사용하여 각각 신경 회로망으로 분류하고, 그 출력을 퍼지 추론 방식으로 종합하여 판정하는 방법을 제시하였다. 이 방법을 사용하면 단일 특징 인식률의 정도에 따라 1 ~ 8% 이상 인식률을 향상시킬 수 있었으며, 인식률이 낮은 패턴에 대해 더욱 효과적으로 동작할 수 있었다. 또한 판정을 위해 퍼지 추론 방법을 사용함에 따라 복잡한 계산을 피했으며, 구조의 복잡성과 분류율을 파라메터로 조절할 수 있는 신경 회로망을 사용하여 특징들을 분류함으로써 실용성과 유연성을 높였다.

A Method of Multifeature Character Recognition Using a Neuro-fuzzy System

Won Ho Choi
Dept. of Electronic Engineering

<Abstract>

A neuro-fuzzy system for the multifeature character recognition is proposed. Two neural network classifiers are used to classify two sets of character features. The outputs of the classifiers are transferred to the fuzzy inference engine to make decision. With this multifeature and fuzzy decision making method, the improved rate of the recognition are varied with the recognition rate of the single feature system. It is shown that the rate of 1 ~ 8% is improved than the single feature recognition system, and that the system can work more effectively when the single system has low recognition rate.

I. 서 론

패턴 인식을 하기 위하여 하나의 특징(feature)을 사용하면, 그 특징이 완벽하지 않는 한, 특징 공간(feature space) 상에 무리(cluster)가 제대로 형성되지 않거나 무리들이 서로 겹쳐지는 부분이 발생하기 마련이다. 특히 문자 인식에서는 클래스 수가 많고 문자의 형태가 다양하므로 문자 모두를 만족시키는 특징을 구하기 어렵다. 따라서 부분적으로 적합한 특징을 사용할 수 밖에 없는데, 이렇게 되면 특징 공간 상에 샘플들이 불규칙적으로 퍼지는 클래스가 있게 된다. 이러한 경우에 아무리 성능이 우수한 분류기를 사용하더라도 분류하는 데에 한계가 있으므로 만족스러운 인식률을 얻기가 어렵다. 이 문제를 해결하기 위하여 근래에 여러 종류의 특징과 분류기를 사용하는 방법이 많이 연구되고 있다[1].

패턴 인식 방법을 크게 두 종류로 구분하면 특징 벡터를 이용하는 방법과 syntactic-and-structural 방법으로 나눌 수 있으며 이를 방법을 위한 수많은 알고리즘들이 개발되어 왔다. 특징 벡터를 분류하기 위한 분류기 만을 보더라도 Bayes 분류기, k-NN 분류기, 여러 가지의 거리(distance) 분류기, 신경망에 근거한 분류기 등 여러 방식이 있다. 또한 문자 인식에 사용 가능한 특징으로서는 화소 밀도, 모멘트, characteristic loci, 그리고 Fourier, Walsh, Hadamard 등의 수학적 변환, skeleton이나 contour로부터 얻은 특징(loop, endpoint, junction, arc, concavity, convexity, stroke 등)과 같은 여러 종류가 있다[3],[4]. 이 특징들 중 서로 보완적인 기능을 갖는 몇 개의 특징을 조합하여 패턴 인식에 적용하면 보다 정확한 인식을 할 수 있겠지만[5], 이 특징들을 분류한 분류기들의 출력을 어떻게 조화시켜서 최종 결정을 할 수 있겠는가 하는 것이 어려운 문제이다.

기존의 방법으로 평균을 취하는 방법, 투표 원리(voting principle)에 의한 방법, Dempster-Shafer formalism에 의한 방법[1]등이 사용되었으나, 하나의 분류기가 후보들의 거리, 무리 내의 샘플 밀도 및 분포와 같은 여러 값을 출력하는 경우에는 위의 방법을 사용하는 것보다는 퍼지 이론을 이용하는 것이 매우 효과적이다.

본 논문에서는 mesh와 방향성 특징[6]의 두 종류를 특징으로 사용하고 노드 확장 신경회로망을 이용하여 신경망 내부의 패턴 샘플의 밀도와 출력단에 나타나는 거리를 구한 다음, 퍼지 추론 방법으로 결정을 내리는 과정을 설명하고자 한다. 구성한 시스템은 그림 1과 같다.

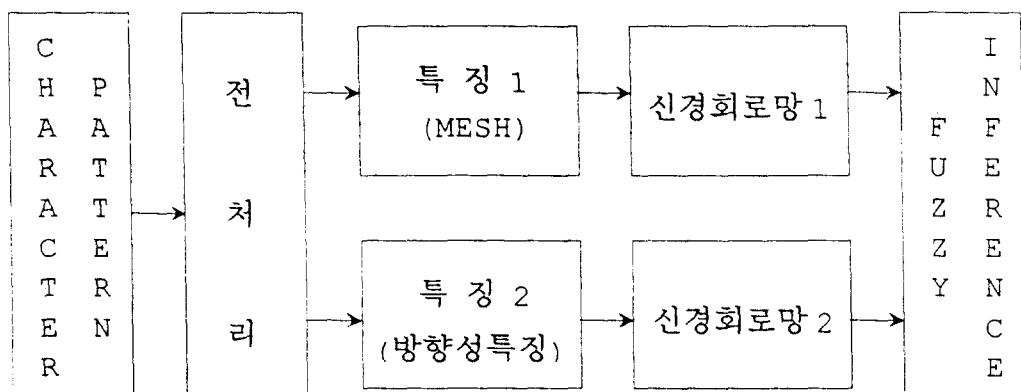


그림 1. 전체 시스템 구성도

Fig. 1 The block diagram of the system

잡음 제거, 크기의 정규화 등의 전처리를 한 후, MESH와 방향성 특징을 추출한다. 이 특징들을 각각의 신경 회로망에 입력시켜 분류를 한다. 사용되는 신경 회로망 1과 2는 동일 종류로서 Adaptive Mahalanobis Distance Classifier(AMDC)[2]를 사용하였다. 신경 회로망으로부터 각 후보들의 거리와 이 거리를 산출하는데 사용된 내부 노드에 소속되어 있는 학습 패턴의 수를 퍼지 추론 시스템이 입력으로 받아서 최종 결정을 내리게 된다.

II. 신경 회로망 분류기

본 연구에서 특징들의 분류기로서 사용된 AMDC는 그림 2와 같이 입력단, 내부단, 출력단의 3단으로 구성되어 있다. 입력단과 내부단의 연결은 완전 연결(full connection)이고, 내부단과 출력단은 부분 연결(partial connection) 상태이다. 입력단의 노드 개수는 입력 패턴의 차원 수와 같으며, 내부단은 학습시 하나씩 발생되므로 내부단의 노드 개수는 샘플의 분포와 학습 조건에 따라 달라진다. 출력단 노드의 개수는 분류하고자 하는 클래스의 개수와 같다.

입력단은 학습과 테스팅 시에 입력되는 입력 패턴을 받아서 그대로 내부단에 전달해 주는 역할을 한다. 하나의 입력 노드는 입력 패턴 벡터의 한 원소(element)를 받아서 모든 내부 노드에 보내 준다.

내부 노드는 입력단에서 받은 입력 패턴과 자신이 갖고 있는 대표점(reference point)과의 거리를 계산하여 출력단으로 보내 준다. 대표점은 내부 노드에 소속된 학습 패턴들의 중심점이다. 거리의 계산은 학습 시에 구성된 편차 벡터(variance vector)를 이용하여 마할라노비스 거리로서 구하게 된다. 이 거리의 계산 시 threshold를 사용하여 입력 패턴이 내부 노드 자신의 주목 영역(attention area) 내에 존재하는가 또는 밖에 있는가를 판단하여 이 정보도 함께 출력단에 전달한다.

주목 영역은 특징 공간 상에서 일정한 원 또는 구(sphere)의 모양을 갖는 것이 아니고 거리 자체가 마할라노비스 거리이므로 하나의 내부 노드에 소속된 학습 패턴의 분포 모양에

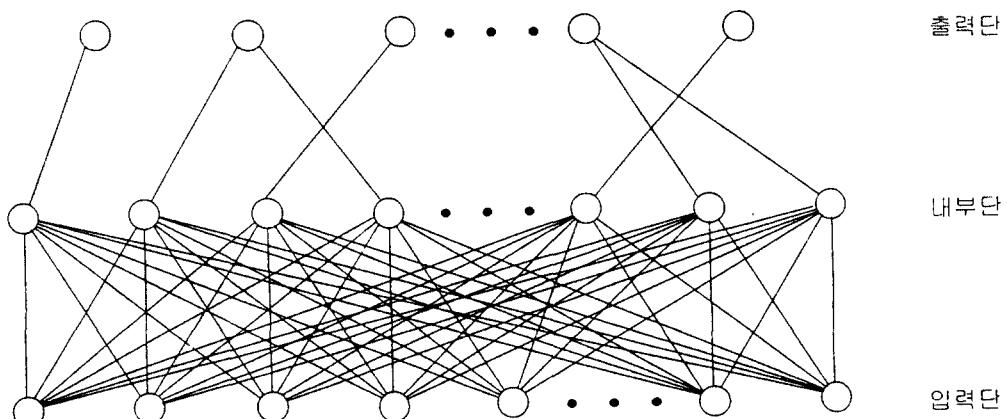


그림 2. AMDC 신경 회로망

Fig. 2 The structure of the AMDC neural network

적용된 주목 영역이 된다. 학습 시에 출력 노드에 의해 활성화된 내부 노드는 입력되고 있는 학습 패턴을 이용하여 대표 벡터(reference vector)와 편차 벡터를 계산한다. 하나의 입력 패턴을 학습할 때에 하나의 내부 노드 만이 대표 벡터와 편차 벡터를 수정하게 되며, 단 한번의 수정으로 학습이 완료된다.

출력 노드는 학습 시에는 지정된 출력 노드 만이 활성화되며 나머지 노드는 비활성 상태에 있지만, 테스팅 시에는 모든 출력 노드가 동작을 한다. 학습 시에 활성화된 출력 노드는 자신과 연결된 내부 노드로부터 마할라노비스 거리와 학습 패턴이 주목 영역 내에 있는가 하는 정보를 전달받아서 어느 내부 노드에 학습 패턴을 귀속시킬 것인가를 결정하여 해당 내부 노드를 활성화시키고 나머지를 비활성화 시킨다. 만약 자신에게 연결된 모든 내부 노드의 주목 영역 내에 학습 패턴이 존재하지 않으면 새로운 내부 노드를 하나 발생시켜 자신과 연결하고 그 내부 노드의 대표 벡터를 학습 패턴과 같게 놓는다. 이 경우 편차 벡터는 초기 편차 벡터(initial variance vector)로 한다.

내부 노드는 자신이 관찰하는 작은 무리가 소속된 클래스에 해당하는 출력 노드로 연결되어 있다. 즉, 하나의 클래스를 작은 클래스들의 집합으로 생각해서 작은 클래스에 해당하는 내부 노드를 소속된 클래스에 해당하는 출력단에 연결하여 준다. 이러한 작은 클래스를 필요로 하는 경우가 문자 인식에서 많이 발생한다.

생성된 내부 노드는 신경망이 입력 패턴을 학습할 때 대표 벡터를 만드는데, 이 벡터는 그 노드에 귀속된 입력 패턴들의 평균값이다. 학습 시에 해당 클래스를 지정하고 입력 패턴을 넣으면 지정된 출력 노드는 자신에게 속한 내부 노드들을 활성화시킨다. 활성화된 내부 노드들은 입력되고 있는 패턴 벡터와 자신이 갖고 있는 대표 벡터와의 거리를 계산해서 주목 영역 (attention area) 내에 입력 패턴이 있는가 또는 밖에 있는가를 판단한다. 만약 주목 영역 안에 있으면 다음 식에 의하여 대표 벡터 m 을 이동시킨다.

$$m_i(t+1) = m_i(t) + \frac{1}{t+1} \{x_i(t+1) - m_i(t)\} \quad (1)$$

$$m_i(1) = x_i(1) \quad (2)$$

여기서 $m_i(t)$ 은 i 번째 내부 노드에 소속된 학습 패턴이 t 개 일 때의 대표 벡터를 뜻한다. 따라서 $m_i(t+1)$ 은 이 노드에 포함되는 학습 패턴이 하나 더 들어왔을 때의 대표 벡터이다. $x_i(t+1)$ 은 i 번째 내부 노드에 $t+1$ 번째로 포함된 입력 패턴 벡터이다. 활성화된 모든 내부 노드의 주목 영역 안에 입력 패턴이 없을 때에는 새로운 내부 노드를 하나 생성시켜 지정된 출력 노드와 연결하고 식 (2)와 같이 그 내부 노드의 대표 벡터는 입력 패턴과 같게 한다.

차기 대표점 $m(t+1)$ 의 계산은 현 대표점 $m(t)$ 에서 학습 입력 벡터 $x(t+1)$ 을 받으면 곧 바로 계산될 수 있으므로, 학습이 완료된 상태에서 추가 학습이 필요한 경우에는 추가되는 학습 패턴을 그대로 넣어 주어도 앞서 학습된 결과에 자동적으로 연결되어 학습될 수 있다.

각 클래스에 속하는 학습 패턴의 분포 모양에 따라 그 클래스가 점유하는 내부 노드, 즉 서브클래스의 개수가 달라진다. 학습 패턴이 널리 분포되어 있어서 하나의 서브클래스의 주목 영역 안에 모두 들어올 수가 없을 때는 새로운 내부 노드를 발생시키게 되고, 또한 여러 개의 그룹으로 나누어진 클래스도 많은 내부 노드를 갖게 된다.

새로 발생된 내부 노드의 초기 주목 영역(initial attention area)은 서브클래스의 개수와

밀접한 관계가 있다. 초기 주목 영역을 넓게 잡으면 서브클래스의 개수가 줄어들고, 좁게 잡으면 서브클래스의 개수가 많아진다. 좁은 영역에 밀접된 패턴을 갖는 클래스는 하나 또는 몇 개만의 내부 노드를 갖게 된다.

AMDC는 마할라노비스 거리를 계산하기 위하여 다음 식과 같은 편차 벡터의 계산을 필요로 한다. 편차 벡터는

$$\sigma_i = [\sigma_{i1} \ \sigma_{i2} \ \sigma_{i3} \ \dots \ \sigma_{in}]^T \quad (3)$$

로 표시된다. 여기서 σ_{ij} 는 i 번 째 내부 노드의 j 축에 대한 편차이다. 편차의 계산은

$$\sigma_{ij} = \frac{1}{t+1} \sum_{r=1}^{t+1} x_j^2(r) - m_{ij}^2(t+1) \quad (4)$$

의 식을 이용하면 된다. 이 식에서 $x_j(r)$ 은 i 번 째 내부 노드로 r 번 째 입력한 j 축 입력의 값, 즉 j 번 째 입력 노드의 입력 값이고 $m_{ij}(t+1)$ 은 i 번째 내부 노드에서 $(t+1)$ 번째 까지 입력된 학습 패턴들의 대표 벡터의 j 번째 원소이다. 실제로 처음 내부 노드를 발생시켰을 때는 대표 값이 입력 패턴과 같고 그 노드의 편차는 0이 된다. 이 경우에 마할라노비스 거리를 계산할 수 없으므로 σ 대신에 d 라는 벡터를 구해서 계산하도록 한다.

$$d_i = [d_{i1} \ d_{i2} \ d_{i3} \ \dots \ d_{in}]^T \quad (5)$$

$$d_{ij} = \begin{cases} \theta & \text{if } \sigma_{ij} \leq \theta \\ \sigma_{ij} & \text{if } \sigma_{ij} > \theta \end{cases} \quad (6)$$

이 식에서 θ 는 σ 의 초기 값으로 생각하면 된다. 다시 말해서 σ_{ij} 값이 θ 보다 작을 때는 θ 로 나누어서 거리를 구하고 θ 보다 클 때는 편차로 나누어서 거리를 구하도록 한다. 따라서 마할라노비스 거리는

$$D_i^2 = \sum_{j=1}^n \frac{(x_{ij} - m_{ij})^2}{d_{ij}^2} \quad (7)$$

로 계산한다.

학습 시에 출력 노드에 의해 활성화된 내부 노드는 각 방향에 대하여 다음 식을 계산한다.

$$g_{ij} = \frac{(x_{ij} - m_{ij})^2}{d_{ij}^2} \quad (8)$$

이 식은 내부 노드의 중심점에서 입력 패턴과의 j 방향에 대한 거리에서 threshold를 뺀 값이다. 모든 방향에 대하여 g_{ij} 가 음수 또는 0이라면, 즉

$$g_{ij} \leq 0 \quad \text{for all } j, \quad j = 1, 2, \dots, n \quad (9)$$

의 조건이 만족되면 입력 패턴은 내부 노드 i 의 주목 영역 안에 들어가게 된다. 이 식에서 α 가 비록 상수이기는 하지만 각 내부 노드별로 크기가 다른 주목 영역을 갖는다. 왜냐하면 식 (8)의 제수 $(d_{ij})^2$ 은 내부 노드에 소속된 샘플의 분포 모양에 따라 그 값이 달라지기 때문이다. 주목 영역과 외부와의 경계선은

$$\frac{(x_{ij} - m_{ij})^2}{d_{ij}^2} = \alpha \quad (10)$$

를 만족하여야 하고 이 식을 고쳐 쓰면

$$(x_{ij} - m_{ij})^2 = \alpha \cdot d_{ij}^2 \quad (11)$$

가 된다. 따라서 유clidean 거리의 개념으로 보았을 때 i 번째 내부 노드의 j 축에 대한 주목 영역은 d_{ij} 가 된다. 그러므로 각 노드마다, 또 각 축 별로 거리가 다른 주목 영역을 형성하게 된다.

내부 노드에서 출력 노드로 전달하는 값은 식 (7)의 D_i 를 그대로 출력 노드로 전달하되, 모든 방향에 대해 주목 영역 안에 입력이 존재하면 음부호를 붙이고 그렇지 않으면 그대로 출력 노드로 전달한다.

$$Z_i = \begin{cases} -D_i^2 & \text{if } x \text{ is in attention area} \\ +D_i^2 & \text{otherwise} \end{cases} \quad (12)$$

출력 노드에서는 Z_i 값을 받아서 부호를 이용하여 어느 노드의 주목 영역 안에 입력 패턴이 존재하는지를 판단하여 Z_i 값으로 어느 노드와 가장 가까운지를 결정할 수 있다.

III. 퍼지 추론 시스템의 설계

3.1 퍼지 추론 시스템 입력 선정

신경 회로망으로부터 퍼지 추론 시스템이 받아들이는 각 후보들의 거리와 이 거리를 산출하는데 사용된 내부 노드에 소속되어 있는 학습 패턴의 수, 즉 밀도는 다음 식과 같다.

$$\text{DISTANCE : } D_{ij} \quad (13)$$

$$\text{DENSITY : } \Delta_{ik} \quad (14)$$

여기서 $i = \{1, 2\}$ 로서 신경 회로망 1과 2의 출력을 의미하고, j 와 k 는 각 분류기에서 거리를 기준으로 설정된 5 개의 후보를 나타내며 $j = k = \{1, 2, 3, 4, 5\}$ 이다.

분류기에서 선정된 5개의 후보 중 두 분류기에 공통으로 포함된 후보들을 그림 3과 같이 퍼지 추론의 최종 결정을 위한 후보 클래스로 선택한다. 그림에서 분류기 1과 2에 공통으로 포함된 후보가 4개 밖에 없으므로 퍼지 추론의 방향성 특징 분류 순위 후보 클래스도 4 개가 된다.

분류기 1과 2에서 사용하는 특징이 서로 다르므로 이들이 출력하는 거리와 밀도의 평균 값도 서로 다르다. 따라서 분류기들의 출력을 같은 비중

방향성특징 분류 순위						Mesh 특징 분류 순위					
순위	1	2	3	4	5	순위	1	2	3	4	5
클래스	전	관	면	선	신	클래스	신	전	선	관	정
후보로 선정된 클래스											
번호	1	2	3	4	5	번호	1	2	3	4	5
클래스	전	관	선	신	X	클래스	전	관	선	신	X

그림 3. 퍼지 추론 후보의 선정

Fig. 3 The selection of candidates for fuzzy inference

으로 취급하기 위해서는 정규화 과정이 필요하다. 정규화된 거리 d_{ij} 와 밀도 δ_{ik} 는

$$d_{ij} = \frac{D_{ij}}{\max_{j \in N} D_{ij}}, \quad (15)$$

$$\delta_{ik} = \frac{\Delta_{ik}}{\max_{k \in N} \Delta_{ik}} \quad (16)$$

이며, 여기서 $N = \{1, 2, 3, 4, 5\}$ 이다. 이 값들은 퍼지 추론 후보로 선정되었을 때 퍼지 추론 후보의 벡터 값으로 그대로 맵핑된다. 하나의 퍼지 추론 후보는 분류기 1과 2의 정규화 거리와 정규화 밀도의 4 개 값을 갖고 있다. 후보 번호를 p 라고 하면 분류기 번호는 $i = \{1, 2\}$ 이므로 p 번 째 후보가 갖는 값은

가 된다.

3.2. 소속 함수(Membership Function)

퍼지 추론을 용이하게 만들기 위하여 거리와 밀도로부터 적절한 소속 정도(membership degree)를 추출하여야 한다. 문자를 인식할 때 밀도보다는 거리의 영향이 크므로 소속 함수도 거리에 대한 변화가 더 커야 한다. 그럼 4에 소속 함수를 그렸다.

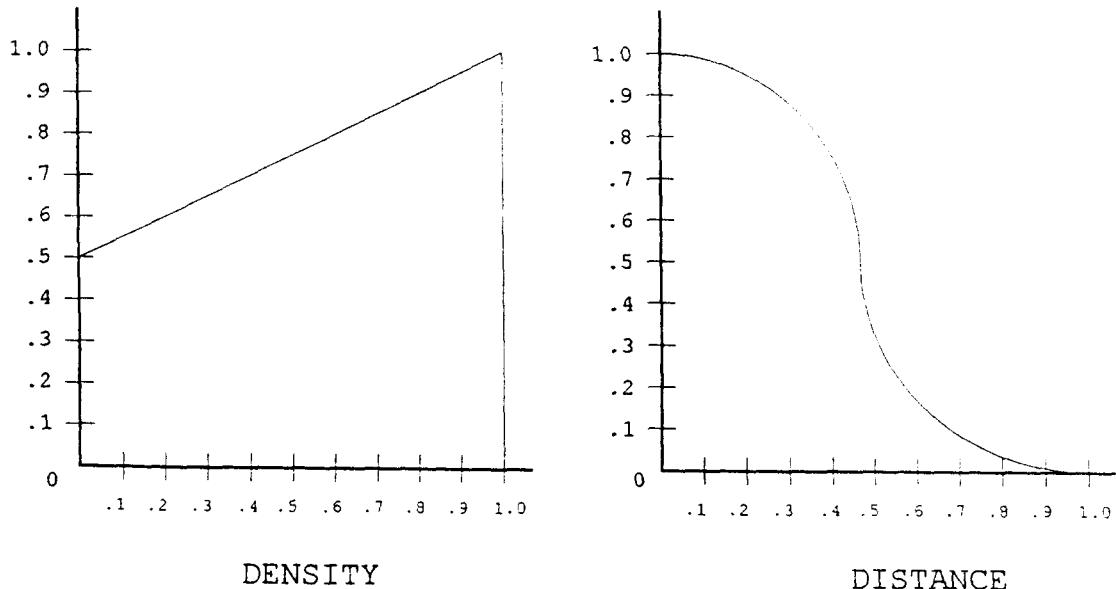


그림 4. 소속 함수
Fig. 4 Membership functions

밀도의 영향을 낮추기 위하여 기울기가 낮은 함수를 밀도에 적용하였으며, 거리는 중간 영역에서 변화가 크도록 sigmoid 함수를 사용하였다. 이들의 식은 다음과 같다.

$$\mu_{\text{den}}(\delta) = T\delta - (T - 1) \quad (18)$$

$$\mu_{\text{dis}}(d) = \frac{1}{1 + \exp\{(d - d_2) * S\}} \quad (19)$$

여기서 T 는 이 함수의 기울기로서 이며, d_2 는 두 번째 후보 클래스가 갖고 있는 거리이다. S 는 sigmoid 함수의 기울기의 정도에 관련되는 파라메터이다. 두 번째 후보 클래스의 거리를 sigmoid 함수의 중앙에 놓음으로써 소속 정도 값의 분포가 넓어지도록 하였다.

소속 정도는 밀도와 거리의 소속 정도의 곱으로 구한다. 따라서 p 번째 후보의 소속 정도는 다음 식과 같다.

$$\mu_{pi} = \mu_{dis}(d_{ip}) * \mu_{den}(\delta_{ip}) \quad (20)$$

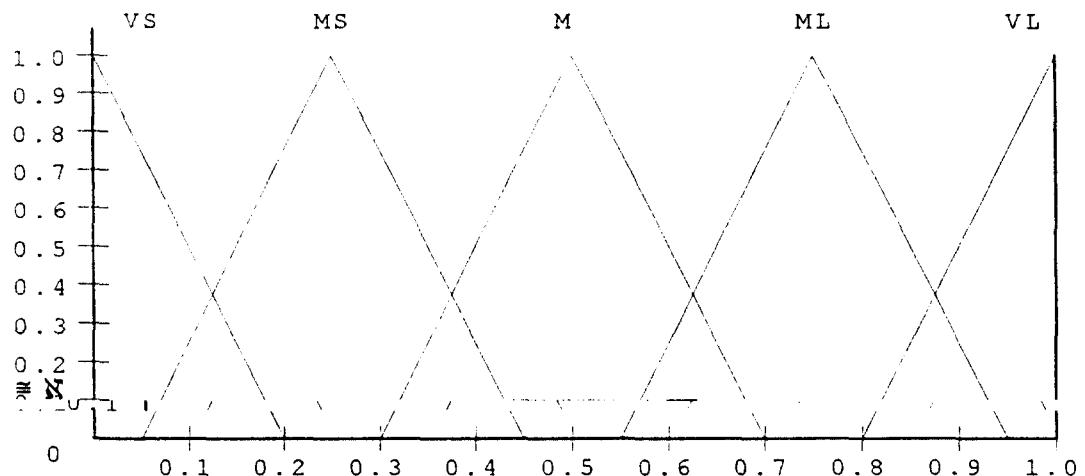


그림 5. 전전부의 소속 함수

Fig. 5 Membership functions for variables CLASSIFIER_I and CLASSIFIER_II

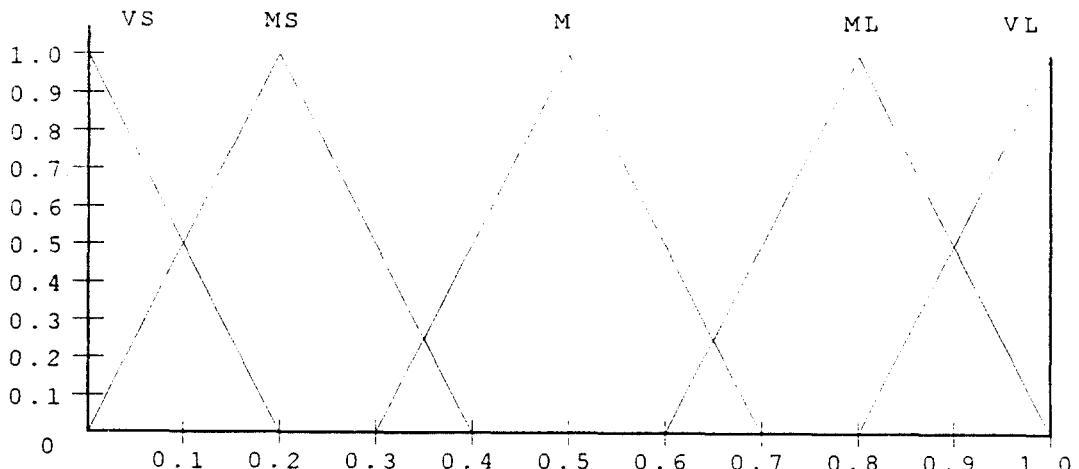


그림 6. 후전부의 소속 함수

Fig. 6 The membership function for the variable OUTPUT

거리 소속 함수가 밀도에 대하여 변화하는 개념으로 볼 수도 있다. 밀도 소속 함수의 기울기와 거리 소속 함수의 지수 값은 실험적으로 구하였다.

퍼지 추론을 위한 전건부와 후건부의 소속 함수는 흔히 사용되는 삼각형의 것을 사용하였다. VL(Very Large), ML(Medium Large), M(Medium), MS(Medium Small), VS(Very Small)의 5 종류가 사용되었으며 전건부의 소속 함수는 그림 5와 같다. VS, MS, M, ML, VL의 중앙은 각각 0, 0.25, 0.5, 0.75, 1.0이고 기울기는 5이다.

후건부의 소속 함수는 그림 6에 그렸으며 0.5를 기준으로 한 출력의 변화를 많이 주기 위하여 MS과 ML의 중심을 0.2와 0.8로 각각 이동시켰다.

3.3 퍼지 룰 및 비퍼지화

두 개의 특징을 각각 분류하였을 때 인식률이 높은 분류기를 분류기I, 낮은 분류기를 분류기II로 지정하여 추론에 사용하는 퍼지 룰을 표 1과 같이 만들었다. 표에서 보는 바와 같이 분류기I의 역할 비중이 분류기II에 비하여 더 크도록 룰을 작성하였다. 예로

If CLASSIFIER_I is LARGE and CLASSIFIER_II is MEDIUM
then OUTPUT is LARGE.

에서 보듯이 분류기I이 분류기II에 우선하는 것을 알 수 있다.

퍼지 추론은 MAX-PRODUCT 방법을 사용하였다. MAX-MIN 방식을 사용하는 경우에는 소속 정도가 작은 분류기의 영향을 주로 받게 되므로 바람직하지 못하다. 따라서 분류기I과 분류기II의 소속 정도를 서로 곱하여 출력을 산출하는 MAX-PRODUCT 방법을 사용하였다.

비퍼지화 방식은 흔히 많이 사용되는 Centroid 방법[7]을 사용하였다.

표 1. 퍼지 룰
Table 1. Fuzzy rules

분류기 I	VS	MS	M	ML	VL
분류기 II					
VS	VS	MS	MS	M	M
MS	VS	MS	M	M	ML
M	MS	MS	M	ML	ML
ML	MS	M	M	ML	VL
VL	M	M	ML	ML	VL

IV. 실험 및 결과

4.1 정규화 및 특징 추출

스캐너를 통해 입력된 임의 크기의 개별 문자를 정규화 과정을 통해 32 X 32의 크기로 통일하였다. 정규화 과정은 식 (21)과 같은 bilinear 보간법을 사용하였다. 임의 크기의 개별 문자 영상을 $f(n_1, n_2)$, $n_1=0, \dots, N_1-1$, $y=1, \dots, N_2-1$ 이라 하면 정규화된 문자 영상 $F(x,y)$, $x=0, \dots, 31$, $y=0, \dots, 31$ 은 다음과 같다.

$$\begin{aligned} F(x,y) &= (1-dx)(1-dy)f(n_1, n_2) + (1-dx)dyf(n_1, n_2+1) \\ &\quad + dx(1-dy)f(n_1+1, n_2) + dx dyf(n_1+1, n_2+1), \\ dx &= \{x/63 - n_1/(N_1-1)\}(N_1 - 1) \\ dy &= \{y/63 - n_2/(N_2-1)\}(N_2 - 1) \end{aligned} \quad (21)$$

정규화된 문자 영상으로부터 수평, 수직 예지(edge)연산자와 수평, 수직 수용 영역(receptive field)[8]을 이용하여 식 (22)-(25)의 특징 벡터 x 를 계산하였다.

$$F_u(i, j) = \sum_{u=-1}^1 \sum_{v=-1}^1 H(u, v) * F(i-u, j-v), \quad i, j = 1, \dots, 31 \quad (22)$$

$$F_v(i, j) = \sum_{u=-1}^1 \sum_{v=-1}^1 V(u, v) * F(i-u, j-v), \quad i, j = 1, \dots, 31 \quad (23)$$

$$X(4i+j) = \sum_{u=4i}^{4i+3} \sum_{v=8j}^{8j+7} F_h(u, v) \quad i = 0, \dots, 7, \quad j = 0, \dots, 3 \quad (24)$$

$$X(i+4j+32) = \sum_{u=4i}^{8i+7} \sum_{v=8j}^{4j+3} F_v(u, v) \quad i = 0, \dots, 3, \quad j = 0, \dots, 7 \quad (25)$$

실험에 사용되는 다른 하나의 특징인 mesh특징은 문자를 일정한 영역으로 나누어서 그 영역 안에 들어 있는 문자의 화소 수를 세는 매우 단순한 특징이다. 이 특징을 위해서 문자 패턴을 64 X 64로 정규화하여 8 X 8 영역으로 나눈다. 이렇게 하면 모두 64 개의 영역이 생긴다. 하나의 영역 안에 있는 문자를 구성하는 화소 수를 세어서 그 영역의 특징으로 삼는다. 따라서 한 문자의 특징은 64 개 영역의 화소 수를 센 값들을 원소로 한 벡터가 된다.

4.2 밀도 소속 함수의 기울기 및 거리 소속 함수의 지수 값 결정 실험

2밀도 소속 함수의 기울기와 거리 소속 함수의 지수 값을 결정하기 위하여 이를 값을 변화시켜 가며 한글 인식 실험을 하였다. 16종 (1600 자)의 인쇄체 한글을 신경 회로망 분류기에 학습시켰으며, 4종 (400 자)의 데이터로 테스팅을 하였다. 그림 7은 밀도 소속 함수의 기울기에 대한 인식률의 변화이고 그림 8은 거리 소속 함수에 대한 인식률의 변화이다.

이 실험 데이터에서 알 수 있듯이 밀도 소속 함수의 기울기는 0.5, 거리 소속 함수인 sigmoid 함수의 지수 값은 9에서 13 사이의 값을 선택하는 것이 좋다. 따라서 본 연구에서는 $T = 0.5$, $S = 10$ 으로 실험하였다.

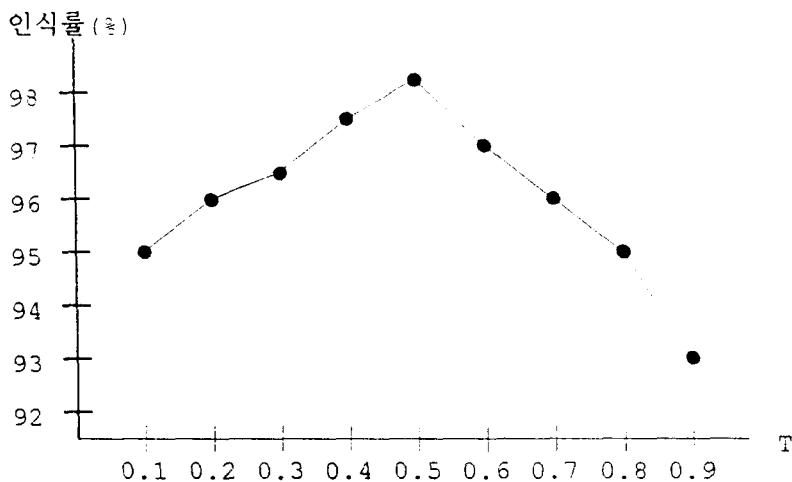


그림 7. 밀도 소속 함수 기울기에 대한 인식률의 변화

Fig. 7 Recognition rates for various values of T .

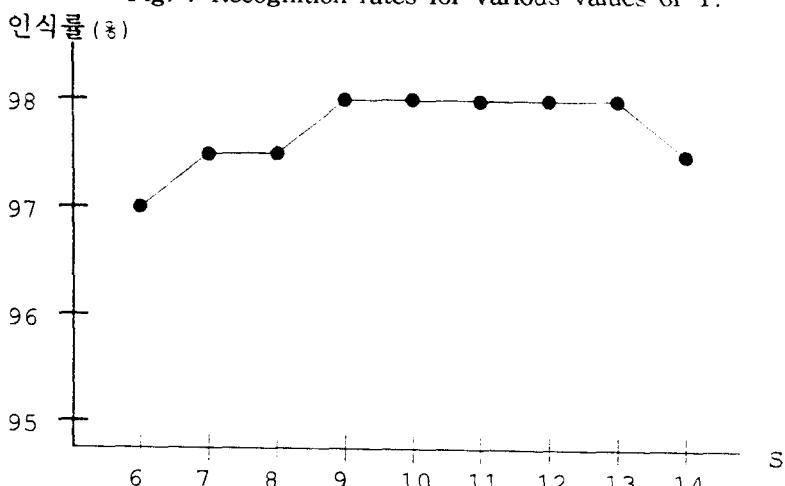


그림 8. sigmoid 소속 함수 지수 값에 대한 인식률의 변화

Fig. 8 Recognition rates for various values of S .

4.3 퍼지 추론을 사용한 다중 특징 시스템

AMDC는 주목 영역 α 가 크면 내부 노드의 개수가 적게 발생하며 α 가 작으면 내부 노드가 많이 발생한다. 내부 노드 수가 작은 경우에는 신경 회로망의 구조가 간단해지는 대신 인식률이 저하되며, 내부 노드 수가 많은 경우에는 그 반대로 동작한다.

Mesh와 방향성 특징을 독자적으로 사용하여 분류한 경우의 인식률과 다중 특징과 퍼지 추론을 사용한 시스템의 인식률을 비교하기 위하여 값을 변화시켜 가며 인식률을 살펴보았다. 그럼 9에서 보듯이 값을 150에서 100까지 변화시키면 mesh의 경우 인식률이 88.5%에서 98.5% 까지 증가되며, 방향성 특징은 91%에서 98.5% 까지의 범위를 갖는다. 두 특징을 모두 사용하면 94.5%에서 99.5% 까지 인식률이 향상되는 것을 알 수 있다. 이 그래프에는 참고로 값에 대한 내부 노드 수를 5의 단위로 인식률과 함께 그려 넣었다.

표 2에는 두 특징의 인식률 평균(R_{av})과 다중 특징 인식률(R_{multi})을 비교하였다. 이 표에서 보면 R_{av} 가 98.25% 일 때는 1.25%가 향상되지만 R_{av} 가 84.25%일 때는 7.75%가 향상된다. 따라서 단일 특징 인식률이 낮은 경우에 다중 특징이 더욱 효과적인 것을 알 수 있다.

표 2. 방향성과 mesh의 평균 인식률과 다중 특징 인식률의 비교

Table 2. Recognition rates of the single feature system and the multifeature system

α	105	115	125	135	145	155	165
R_{av} (%)	98.25	97.0	95.5	93.75	91.5	87.25	84.25
R_{multi} (%)	99.9	99.0	99.0	97.5	96.0	94.5	92.0
$R_{multi} - R_{av}$	1.25	2.0	3.5	3.75	4.5	7.25	7.75

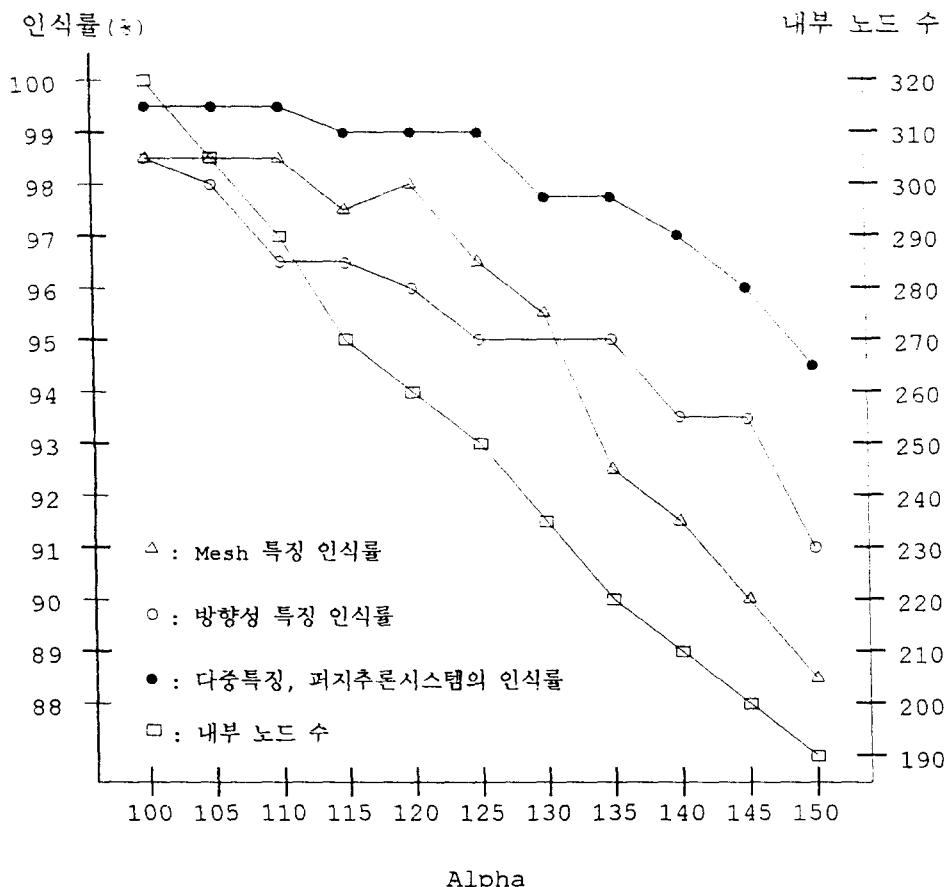


그림 9. 방향성, mesh, 다중 특징의 인식률 비교
 Fig. 9 Recognition rates of the directional, mesh, and
 multifeature character recognition system.

V. 결 론

본 연구에서는 불완전한 특징이라도 다중 특징 방식으로 여러 개를 같이 사용하면 인식률을 향상시킬 수 있다는 사실을 입증하였다. 단일 특징 인식률의 정도에 따 1 ~ 8% 이상 인식률을 향상시킬 수 있었으며, 인식률이 낮은 패턴에 대해 더욱 효과적으로 동작할 수 있었다.

복잡성과 분류율을 조절할 수 있는 신경 회로망을 사용하여 특징들을 분류함으로써 실용성과 유연성을 높였다고 생각되며, 이 신경 회로망의 출력을 퍼지 추론을 이용하여 종합하고 판정함으로써 다중 특징의 사용을 간편화할 수 있었다. 이러한 방식을 사용하면 3개 이상의 다중 특징을 문자 인식에 적용하는 경우에도 무리없이 처리될 수 있으리라 보며, 인식률을 더욱 높일 수 있을 것으로 기대된다.

참 고 문 헌

1. Lei Xu, Adam Krzyzak and Ching Y. Suen, "Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition," IEEE Trans. on Syst., Man, and Cybern., vol.22, no.3, pp.418 -435, May/June 1992.
2. 최원호, 이병래, 최동혁, 박규태, "한글 인식을 위한 신경망 분류기의 응용," 대한전자공학회 논문지 제27권 제8호, pp.93-103, 8월, 1990.
3. C. Nadal, R. Legault, and C.Y. Suen, "Complementary Algorithms for the Recognition of Totally Unconstrained Handwritten Numerals," in Proc. 10th Int. Conf. Pattern Recog., vol. A, pp.434-449, June 1990.
4. S.T. Bow, Pattern Recognition and Image Preprocessing, Marcel Dekker, New York, 1992.
5. D. Duerr, W. Haettich, H. Tropf, and G. Winkler, "A Combination of Statistical and Syntactical Pattern Recognition Applied to Classification of Unconstrained Handwritten Numerals," Pattern Recognition, vol.12, no.3, pp.189-199, 1980.
6. T.Y. Young, K.S. Fu, ed., Handbook of Pattern Recognition and Image Processing, Academic Press, 1986.
7. B. Kosko, Neural Networks and Fuzzy Systems :A Dynamical Systems Approach to Machine Intelligence, pp.314-316, P-H, 1992.
8. 최원호, 최재하, 고속 처리용 문자특징의 성능 비교, 울산대학교 공학 연구논문집, 제22권 제1호, pp.79-84, 1991.