

Segmentation of Objects with Multi Layer Perceptron by Using Informations of Window

Young-Tae Kwak¹⁾

Abstract

The multi layer perceptron for segmenting objects in images only uses the input windows that are made from a image in a fixed size. These windows are recognized so each independent learning data that they make the performance of the multi layer perceptron poor. The poor performance is caused by not considering the position information and effect of input windows in input images. So we propose a new approach to add the position information and effect of input windows to the multi layer perceptron's input layer. Our new approach improves the performance as well as the learning time in the multi layer perceptron. In our experiment, we can find our new algorithm good.

Keywords : Multi Layer Perceptron, Segmentation Objects

1. 서론

IT 기술의 발전과 함께 영상은 아날로그에서 디지털로 변환에 따라, 영상의 저장 및 관리가 용이해지고 높은 정밀도 또한 얻을 수 있었다. 높은 정밀도를 가진 영상은 그 크기가 클 뿐만 아니라, 많은 정보와 객체를 포함하고 있으며 이와 같은 물체의 영상을 추출하는 것은 영상처리, 패턴인식, 컴퓨터그래픽 등 컴퓨터를 이용하여 시각적인 데이터를 처리하는데 필수적이다. 지금도 영상처리나 패턴인식에서 물체의 객체를 추출하는 알고리즘이 개발되어 사용되고 있다. 그러나 객체 추출이 정적으로 이루어지기 보다는 사용자의 의도가 반영되는 동적인 알고리즘 개발이 최근에 많이 연구되고 있다. 이런 연구는 각기 영상의 특징에 따라 다른 알고리즘을 적용하는 어려움을 덜 수 있는 장점이 있다.

인간은 물체 인지 과정에서 먼저 물체의 객체를 추출하고 자신의 축적된 지식을 이용하여 물체를 인식한다. 이와 같은 인지 과정을 모델링한 것이 신경회로망이다. 신경회로망은 수학적인 모델링이나 해결을 위한 구체적인 지식이 없어도 학습 기능에 의

1) 전북 익산시 마동 194-5 익산대학 컴퓨터정보과 조교수
E-mail : kwak@iksan.ac.kr

해 문제를 해결할 수 있기 때문에 연구의 관심이 되고 있으며, 패턴인식, 연상기억, 비선형 함수의 근사화, 최적화 등에 응용되고 있다. 그중 학습 알고리즘이 용이하고 구현이 쉽기 때문에 여러 응용 분야에서 이용되고 있는 모델링이 다층퍼셉트론이다(Lippmann(1987)). 이런 다층퍼셉트론의 학습기능을 이용하여 영상에서 객체를 추출할 수 있는 이미지 분할 알고리즘을 본 논문에서 제안하고자 한다.

일반적으로 기초적인 객체 추출 알고리즘으로 사용되는 윤곽선 추출 알고리즘은 영상의 1차, 2차 미분을 얻기 위해 마스크를 이용한다. 이런 방법에는 소벨, 프리위트, 로버트, 라플라시안, 캐니 등의 윤곽선 추출 방법 등이 있다(Petrou(1999)). 이런 정적인 윤곽선 추출은 잡음 제거라는 전단계가 필요하며, 설정 임계치에 따라 추출 결과가 좌우된다. 그리고 동적인 윤곽선 추출 알고리즘으로는 Kass(Kass(1987))의 스네이크와 Xu의 GVF(Xu(1998)) 등이 있다. Kass의 스네이크는 초기 정점의 위치 결정과 오목한 부분의 윤곽선 추출이 어려운 문제점이 있으며, GVF는 영상으로부터 에지맵(edge map)을 만들고 GVF의 편미분을 계산하는데 많은 시간을 요구한다.

다층퍼셉트론을 영역 분할이나 윤곽선 추출에 적용한 연구로써 Shang(Shang(1994))과 Brown은 영상의 특징을 추출하기 위해서 Co-occurrence matrix를 사용하였다. 그리고 Farrokhio(Farrokhnia(1990))는 Gabor 변환을 사용하는 단층퍼셉트론을 사용하였다. 또한 Haddon(Haddon(1994))과 Boyce는 다층퍼셉트론의 입력에 Hermite 특징을 이용하였다. 이런 연구들의 특징은 이미지 분할 및 객체 추출 과정 전에 다층퍼셉트론의 입력으로 이미지 자체를 사용하기 보다는 미리 추출된 영상의 특징을 이용하고 있다.

반면에 Middleton(Middleton(2004))과 Damper는 사전에 계산된 특징을 사용하지 않는다. 이 연구에서는 윤곽선 마스크와 비슷한 크기의 윈도우를 설정하여 윈도우의 크기를 다층퍼셉트론의 입력으로 추가하고 있다. 그러나 입력으로 사용되는 윈도우 영상은 다층퍼셉트론에서 각각 독립된 학습 데이터로 취급되기 때문에 성능이 매우 낮다. 이런 단점을 극복하기 위하여 본 논문에서는 입력 영상에서 입력 윈도우가 가지고 있는 위치 정보와 입력 영상과의 영향 정보를 다층퍼셉트론의 입력층에 추가하여 다층퍼셉트론의 성능을 개선하고자 한다.

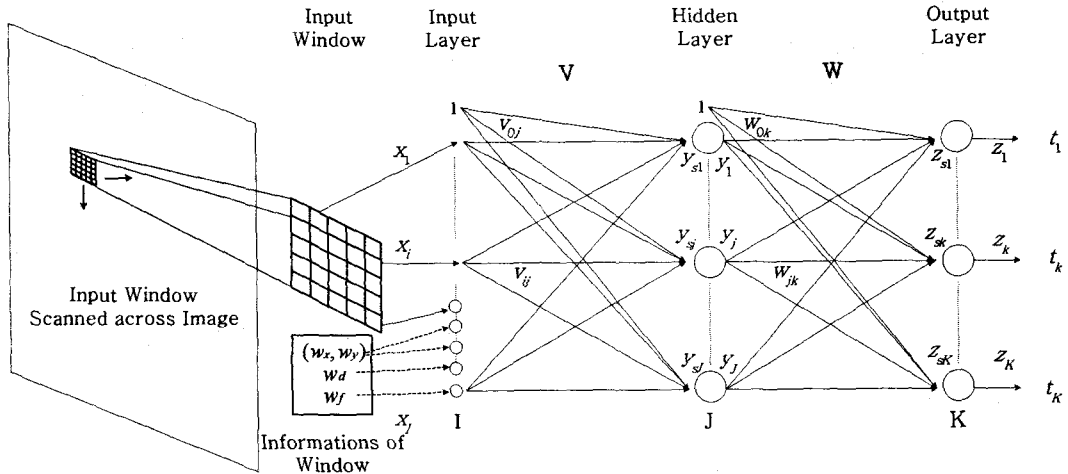
본 논문의 구성으로 2장에서는 사용된 다층퍼셉트론의 구조와 학습알고리즘을 소개한다. 학습알고리즘은 고전적인 오류역전파 알고리즘 대신 개선된 학습알고리즘을 사용한다. 3장에서는 입력 윈도우의 위치 및 거리 정보와 픽셀값의 영향력을 계산하여 다층퍼셉트론의 입력으로 추가하는 방법을 제안한다. 4장에서는 실제 영상을 대상으로 실험하여 결과를 분석하고 5장에서는 결론을 맺는다.

2. 다층퍼셉트론과 개선된 오류역전파 학습

2.1 다층퍼셉트론 구조

대표적인 전방향 신경회로망의 모델 중에서, 많이 사용되는 모델이 다층퍼셉트론이다. 그리고 본 논문에서 사용한 다층퍼셉트론은 <그림 1>과 같은 은닉층이 하나 있는 2층 퍼셉트론 구조이다. 이런 2층 퍼셉트론은 일반적으로 2층 이상의 다층퍼셉트론보다 지역 최소점에 빠질 확률이 낮고, 일반화 성능이 좋다(Islam(2001)). 또한, 2층 퍼셉트론은 어떠한 형태의 사상도 나타낼 수 있는 일반적인 근사화 함수임이 증명되

어 있다(Hornik(1989)).



<그림 1> 다층퍼셉트론의 구조

<그림 1>에서, 입력 노드(xi)는 단지 입력 패턴을 전달만 하며, 노드의 역할에 따라 은닉 노드(yj)와 출력 노드(zk)로 나누어진다. 그리고 입력층의 'Informations of Window'는 입력 영상에 대한 윈도우의 위치, 거리 및 픽셀 비율을 나타내는 추가적인 입력이다. 은닉 노드는 입력값에 가중치를 취하여 합한 뒤에, 그 결과를 활성화 함수로 출력한다. 그리고, 출력 노드는 은닉층의 출력값을 대상으로 은닉 노드와 유사한 일을 수행한다. 여기서, 입력 노드의 수와 출력 노드의 수는 학습 패턴의 차원과 목표값의 차원에 따라 결정되지만, 은닉 노드의 수는 학습할 문제의 복잡도에 따라 다르다.

본 논문에서 전체적으로 사용할 표기법은 다음과 같다.

$$\begin{aligned} \text{은닉층 벡터} : y_p &= [1, y_{p1}, \dots, y_{pj}, \dots, y_{pJ}]^T \\ \text{출력층 벡터} : z_p &= [z_{p1}, \dots, z_{pk}, \dots, z_{pK}]^T \end{aligned} \quad (1)$$

여기서, 각 벡터는 열벡터이고, p는 학습 패턴의 인덱스이며, 각 벡터의 첫 번째 요소 중 1은 바이어스를 위한 입력값이다. 그리고 tk는 k번째 출력 노드에 대한 목표 패턴을 나타낸다. 또한, j번째 은닉 노드로 연결된 가중치 벡터는 $v_j = [v_{0j}, \dots, v_{ij}, \dots, v_{Ij}]^T$ 이며, k번째 출력 노드에 연결된 가중치 벡터는 $w_k = [w_{0k}, \dots, w_{jk}, \dots, w_{Jk}]^T$ 이다. 가중치 벡터도 열벡터이며, v_{0j} 와 w_{0k} 는 각 노드의 바이어스이다. 이런 가중치 벡터를 행렬로 나타내면 다음과 같다.

$$\begin{aligned} V &= [v_1, \dots, v_j, \dots, v_J] \\ W &= [w_1, \dots, w_k, \dots, w_K] \end{aligned} \quad (2)$$

2.2 개선된 오류역전과 학습

<그림 1>의 다층퍼셉트론을 학습하기 위해 Rumelhart는 오류역전과 학습을 제안하였다(Rumelhart(1986)). 오류역전과 학습에서 사용하는 오차 함수는 식(3)과 같은 평균제곱오차(mean square error)함수이다. 이 오차 함수는 각 출력 노드의 출력값과 그 노드에 대한 목표값의 오차를 제곱해서 모두 더한다. 식(3)에서 P는 전체 학습 패턴의 수이고, K는 출력 노드의 수이다.

$$E = \frac{1}{2} \sum_{p=1}^P \sum_{k=1}^K (t_{pk} - z_{pk})^2 \quad (3)$$

경사 강하법을 사용하는 오류역전과 학습은 우선, 가중치를 임의로 초기화 하고 학습 패턴과 목표 벡터를 지정해 준 다음, 식(4)와 같이, 가중치 공간에서 오차가 감소하도록 가중치를 수정하는 과정을 반복한다.

$$\begin{aligned} w_{jk}(t+1) &= w_{jk}(t) + \eta(t_{pk} - z_{pk})f'(net_{pk})y_{pj} = \delta_{pk}y_{pj} \\ v_{ij}(t+1) &= v_{ij}(t) + \eta \sum_{k=1}^K \delta_{pk}w_{jk}f'(net_{pj})x_{pi} = \delta_{pj}x_{pi} \end{aligned} \quad (4)$$

식(4)는 하나의 학습 패턴(p)에 대한 가중치 조정이다. $v_{ij}(t)$ 와 $w_{jk}(t)$ 는 현재의 가중치이며, $v_{ij}(t+1)$ 와 $w_{jk}(t+1)$ 는 새로 조정되는 가중치이다. η 는 학습 상수이고, δ_{pk} 와 δ_{pj} 는 각각 출력 노드와 은닉 노드의 오차 신호이다. 이런 오차 신호에는 시그모이드 함수의 미분이 포함되어 있어, 현재의 학습 패턴에 대한 오차는 크지만, 전체 오차가 충분히 작으면, 현재의 오차를 감소시키지 못하고 오차 신호를 약화시키는 포화상태에 이르러 학습속도의 저하를 가져온다.

이런 문제점을 해결하기 위해 Wang과 Lengelle은 다층퍼셉트론의 계층별 학습을 제안하였다(Wang(1996), Lengelle(1996)). 계층별 학습은 가중치의 조정이 오차에 대해 선형적으로 조정되므로 학습속도가 빠르지만 은닉층의 목표 벡터가 활성화 함수의 범위를 벗어나는 경우, 그 값을 절삭하는 변환 함수를 사용기 때문에, 일정한 학습 시간 이후, 더 이상 학습이 진행되지 않는 현상이 나타난다(Yam(1997)). 이런 단점을 극복하기 위해 광영태(광영태(2001))는 계층별 학습의 장점인 빠른 학습 속도와 오류역전과 학습의 일반화 성능을 가지는 개선된 오류역전과 학습을 제안하였다. 따라서 본 연구에서는 오류역전과 학습에 의한 은닉층의 경사 벡터와 최소 제곱법(least square method)에 의한 은닉층의 목표 벡터를 이용하는 개선된 학습알고리즘을 사용한다.

학습알고리즘에서, 출력층의 가중치는 식(4)와 같은 오류역전과 학습을 하고, 은닉층의 가중치는 식(5)에서 구한다. 식(5)에서 Δy_{pj}^* 는 식(6)과 같이 정의된다. 식(6)에서 $\Delta y_{pj}(t)$ 는 오류역전과 학습에서 $-\partial E/\partial y_{pj}$ 이다. 그리고 yn_{pj}^* 는 계층별 학습에서 은닉층 목표벡터(y_{pj}^*)를 구하여 그 값이 시그모이드 함수의 범위를 벗어나기 때문에 시그모이드 함수의 범위로 정규화한 값이다.

식(6)에서 은닉층 목표 벡터에 대한 각 성분(y_{pj}^*)이 시그모이드 함수내에 존재하면, 그 노드는 학습이 올바르게 된 것이므로, 오류역전과 학습의 $\Delta y_{pj}(t)$ 를 사용한다. 그러

나, 은닉층 목표 벡터의 성분(y_{pj}^*)이 시그모이드 함수의 범위를 벗어나면, 학습이 올바르게 진행되지 않은 상태이므로 은닉층 목표 벡터를 참조하여 새로운 가중치 벡터를 만든다.

$$v_{ij}(t+1) = v_{ij}(t) + \eta \Delta v_{ij} \quad (5)$$

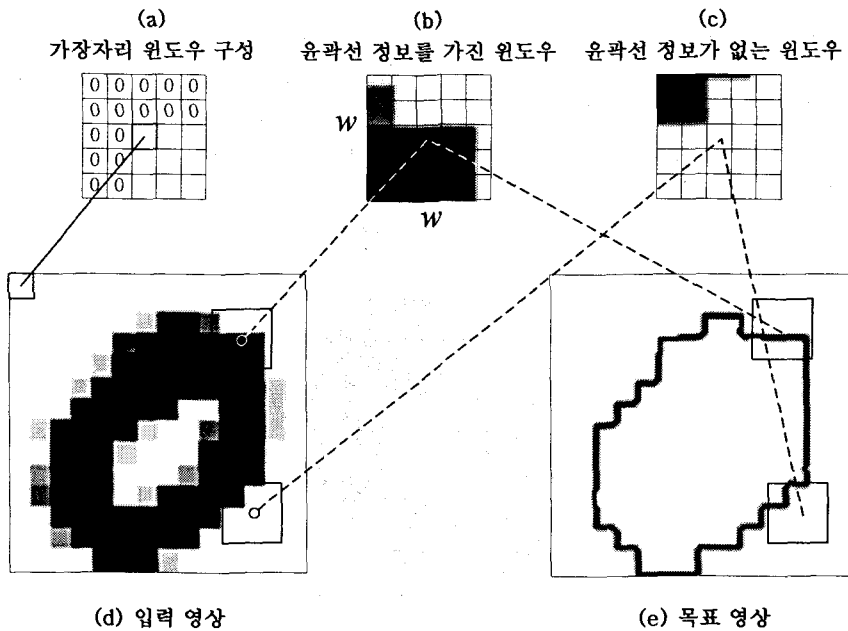
$$\Delta v_{ij} = \Delta y_{pj}^* f'(net_{pj}) x_{pi}$$

$$\Delta y_{pj}^* = \begin{cases} \Delta y_{pj}(t) & \text{if } |y_{pj}^*| < 1 \\ \Delta y_{pj}(t) + yn_{pj}^* & \text{otherwise} \end{cases} \quad (6)$$

개선된 오류역전파 학습은 오류역전파 학습의 비선형적인 학습과 최소 제곱법의 선형적인 학습을 함께 이용한다. 따라서, 계층별 학습의 단점인 낮은 일반화 성능을 극복하고, 오류역전파 학습의 학습 속도를 향상시킬 수 있다.

3. 윈도우 정보를 이용한 학습

오류역전파 학습은 학습 패턴에 대한 목표값을 가지는 교사학습(supervised learning)이므로 학습할 영상에 대해 윤곽선이 추출된 목표 영상이 필요하다. <그림 2>는 이러한 예를 보여준다. <그림 2>의 목표 영상은 입력 영상에 대해 유일하게 결정되지 않으며 사용자에게 따라 다르게 추출될 수 있다. 여기서 우리는 입력 영상에 대해 사용자가 직관적으로 윤곽선을 그리고 이런 영상을 목표 영상으로 설정했다.



<그림 2> 입력 영상과 목표 영상

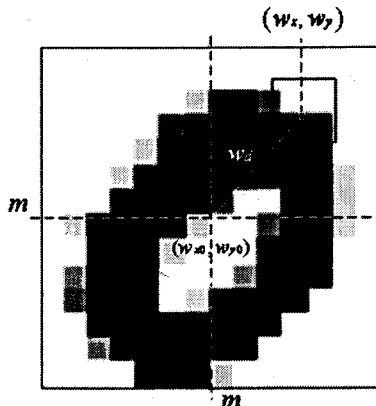
칼라 영상은 흑백 영상보다 많은 정보를 가지고 있으나 불필요한 정보도 가지고 있어 영상 분할이나 윤곽선 추출에서는 흑백 영상을 사용한다. 본 연구에서도 <그림 2>와 같은 그레이 레벨의 흑백 영상을 사용한다. 윈도우를 이용한 스캔은 <그림 2>과 같이 윈도우의 크기(w)를 설정하여 입력 영상을 스캔하며 스캔영역을 벗어나는 <그림 2>의 (a)와 같은 픽셀은 픽셀값을 0으로 하여 다른 노드에 영향을 주지 않게 한다.

다층퍼셉트론의 학습에서 <그림 2>의 (d)와 같은 입력 영상을 모두 학습할 경우 문제가 발생한다. 그것은 영상에서 물체의 객체나 윤곽선은 영상 전체에 균일하게 분포되어 있지 않다. 즉, 윤곽선이나 객체에 대한 중요한 정보를 포함하는 윈도우는 전체 영상중에서 일부분이 된다. 그러므로 입력 영상을 모두 학습할 경우 다층퍼셉트론의 출력노드가 한쪽으로 포화되는 현상이 나타난다. 따라서 윤곽선이나 객체 추출의 기준이 되는 정보를 가진 윈도우를 학습 데이터로 먼저 구성하고, 같은 수만큼의 윤곽선 정보를 갖지 않는 윈도우를 나머지 입력 영상에서 임의로 선택하여 학습 데이터를 구성한다.

<그림 1>의 다층퍼셉트론은 활성화 함수를 식(7)과 같은 -1~+1의 시그모이드 함수를 사용한다. 식(7)의 λ 는 시그모이드 함수의 기울기이다. 그러므로 학습 데이터 중 윈도우의 중심 픽셀이 윤곽선을 구성하는 픽셀을 포함하면 출력 노드의 목표값이 1이 되고 그렇지 않으면 -1이 된다. 또한 시험 데이터에 대한 다층퍼셉트론의 출력은 출력층 노드의 값이 0을 기준으로 하여 그 이상이면 현재 입력 윈도우의 중심이 윤곽선 픽셀에 해당되며, 0이하이면 현재 입력 윈도우의 중심은 윤곽선 픽셀을 포함하지 않는 것으로 설정한다.

$$f(net) = \frac{2}{1 + e^{-\lambda net}} - 1 \tag{7}$$

영상의 객체 추출을 위해 응용된 초기 다층퍼셉트론(윈도우만 이용)은 시험 결과가 매우 낮다. 이것은 단지 윈도우 입력 영상을 각각의 독립된 데이터로 취급하기 때문이다. 이런 단점을 극복하기 위한 방법으로 본 논문에서는 입력 영상에서 입력 윈도우가 가지고 있는 위치 정보와 입력 영상과의 영향력 정보를 다층퍼셉트론의 입력층에 추가하여 다층퍼셉트론의 성능을 개선하고자 한다.



<그림 3> 입력 영상의 윈도우 정보

우선, 입력 윈도우의 위치 정보는 <그림 3>과 같은 윈도우의 중심 픽셀에 대한 좌표값(w_x, w_y)을 사용한다. 좌표계의 원점(w_{x0}, w_{y0})은 전체 이미지의 중심으로 설정하고 각 좌표는 1과 -1로 정규화 하여 입력층에 전달한다. 두 번째는 윈도우의 중심 픽셀과 좌표계 원점 사이의 거리 정보(w_d)를 식(8)로 정의하며 w_d 는 0~1의 값을 가지므로 활성화 함수 범위내로 정규화한 후 입력층에 전달한다. 여기서 m 은 입력 영상에 대한 크기를 픽셀로 나타낸 것이다.

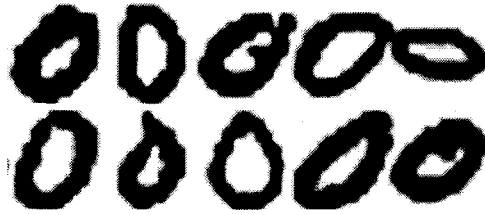
$$w_d = \frac{\sqrt{2}}{m} \sqrt{(w_x - w_{x0})^2 + (w_y - w_{y0})^2} \quad 0 \leq w_d \leq 1 \quad (8)$$

세 번째 사용되는 입력 정보는 입력 영상에서 입력 윈도우가 차지하는 영향력에 관한 정보로써 식(9)로 정의된다. w 윈도우 크기 내에 존재하는 픽셀이 모두 흑이나 백이면 이런 입력 윈도우는 입력 영상에서 윤곽선 정보를 갖지 않는 즉, 학습에서 영향력이 작은 윈도우이다. 반면에 입력 윈도우가 흑백이 일정하게 분포하고 있으면 윤곽선 정보를 포함하여 학습에 도움이 되는 윈도우가 된다. 식(9)에서 $I_{i,j}$ 는 각 픽셀의 그레이값을 나타내고 m 은 입력 영상에 대한 픽셀을 나타낸다. w_f 는 윈도우내의 모든 픽셀이 흑이나 백인 경우는 0에 근사하는 값을 가지며 흑백이 균일하게 분포하는 경우는 1이나 -1로 근사화 하는 정규화 과정을 거쳐 다층퍼셉트론에 입력한다. <그림 3>은 제안된 윈도우 정보에 대한 그림이다.

$$w_f = \frac{\sum_{i=1}^w \sum_{j=1}^w I_{i,j}}{\sum_{i=1}^m \sum_{j=1}^m I_{i,j}} \quad (9)$$

4. 실험 결과

실험은 필기체 숫자를 대상으로 실험했다. 필기체 숫자는 CEDAR 데이터베이스(Hull(1994)) 중 숫자 0을 대상으로 100개를 학습에 사용했으며, 시험은 50개를 사용했다. 한 숫자의 영상의 12x12픽셀로 구성되며 각 픽셀은 256레벨을 16레벨의 그레이 이미지로 축소한 후 (-1~+1)로 정규화 한 다음 다층퍼셉트론에 입력시켰다. 이것은 각 픽셀변화에 따라 다층퍼셉트론이 학습해야할 데이터를 축소하기 위해서이다. <그림 4>는 학습에 사용된 패턴을 그레이 이미지로 나타낸 것이다. 그리고 다층퍼셉트론의 구성은 윈도우의 크기(w)가 5인 경우는 입력 노드의 수가 25+4(윈도우 좌표=2, 거리, 영향력)=29로 구성되며, 윈도우의 크기(w)가 7인 경우는 49+4=53개이다. 은닉노드의 수는 실험을 통하여 10개로 결정하였고 출력 노드의 수는 입력 윈도우가 윤곽선 정보를 가지고 있는지를 판단하기 위해 1개로 구성하였다.



<그림 4> 실험 영상

<표 1> 윈도우 크기에 따른 실험 결과

윈도우 크기	학습 패턴		시험 패턴	
	기존 방법 (윈도우만 사용)	윈도우 위치정보	기존 방법 (윈도우만 사용)	윈도우 위치정보
w=5	53%	72%	47%	68%
w=7	55%	76%	49%	71%

<표 1>은 윈도우 크기의 변화에 따른 실험 결과를 보여준다. 실험 결과는 다층퍼셉트론의 가중치를 임의의 작은 값으로 각각 5회씩 초기화하고 그 결과를 평균한 값이다. <표 1>이 나타내는 결과값은 숫자 0에 대한 인식률을 나타내는 것이 아니라, 윤곽선 정보를 가진 입력 윈도우가 출력 노드를 윤곽선 픽셀로 출력하며, 반대로 윤곽선 정보를 갖지 않은 입력 윈도우에 대해 출력 노드가 윤곽선이 아닌 픽셀로 인식한 결과이다.

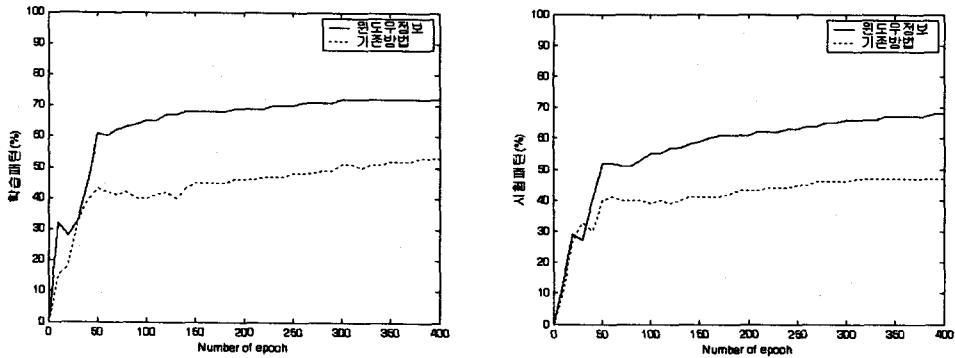
<표 1>에서 ‘기존 방법’은 윈도우의 위치 정보와 영향력 정보를 사용하지 않는 경우로써 기본적인 객체 추출 방법이다. 반면에 ‘윈도우 위치 정보’는 3절에서 소개한 3가지 정보가 다층퍼셉트론의 입력으로 전달된다. 3가지 정보의 단위는 픽셀이다. 우선 입력 윈도우의 좌표(w_x, w_y)는 좌표계의 원점을 (6, 6)으로 잡았다. 그리고 각각 x, y의 좌표를 최대 좌표값을 기준으로 -1~1로 정규화 하여 입력시켰다. 입력 윈도우의 거리 정보(w_d)는 식(8)의 정의처럼 0~1의 값을 가진다. 거리 정보(w_d)를 -1~1로 근사화 하기 위해 좌표계의 1과 3사분면의 값은 양으로, 2와 4사분면을 음으로 정규화했다. 이것은 윤곽선의 대칭성을 고려한 것이다. 입력 윈도우의 영향력 정보(wf)에서는 먼저 입력 영상의 그레이값에 대한 합, 즉 식(9)의 분모를 계산한다. 그리고 입력 윈도우에 대한 그레이값의 합중에서 최대값을 기준으로 wf를 0~1로 반비례로 정규화한 다음, 거리 정보에서처럼 1과 3사분면의 값은 양으로, 2와 4사분면을 음으로 다층퍼셉트론에 입력했다.

<표 1>의 결과, w=5인 경우는 윈도우 자체 픽셀 정보만을 이용하기 때문에 성능이 매우 낮다. 이런 실험 결과는 Middleton(2004)에서도 확인할 수 있다. 반면에 ‘윈도우의 위치 정보’를 사용한 결과는 상대적으로 높은 성능을 얻을 수 있었다. 이것은 입력 윈도우의 3가지 정보에 의해 입력 윈도우가 각각의 독립된 데이터로 인식되는 것이 아니라 입력 영상 전체에서 입력 윈도우가 가지고 있는 정보를 활용하기 때문이다. 그리고 윈도우의 크기(w=7)를 증가하면 성능도 증가하는데 이 원인은 입력 윈도우의 크기가 증가함에 따라 입력 영상에서 전체적으로 입력 윈도우를 파악할 수 있는 능력

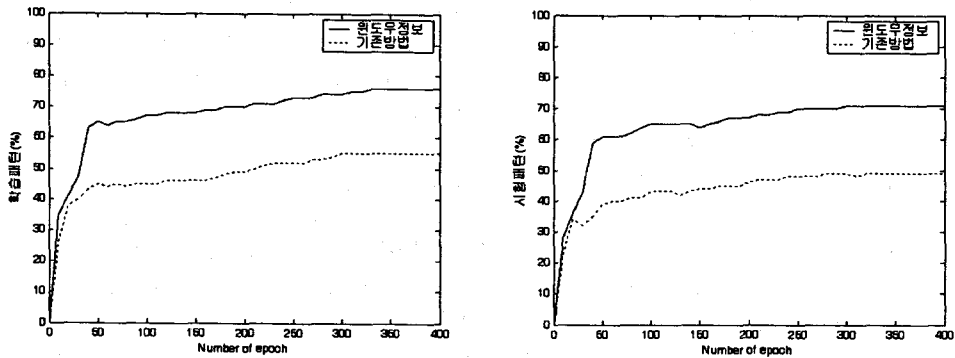
이 증가하여 성능 또한 증가하게 된다. 그러나 $w=9, 11$ 로 증가하는 것은 입력 윈도우의 크기가 입력 영상의 크기와 비슷해지므로 오히려 성능이 떨어지는 결과를 얻었으며, 입력 윈도우의 크기는 입력 영상 크기의 $1/2$ 이하 정도로 결정하는 것이 유용한 결과를 얻을 수 있었다.

<그림 5>는 윈도우 크기가 $w=5$ 인 경우의 윤곽선 인식의 결과를 나타낸다. 여기서 가로축은 학습시간을 epoch 단위로 나타내며 각 실험 결과는 매 10 epoch마다 출력했다. 세로축은 윤곽선 정보를 포함하는 입력 윈도우가 출력 노드에서 입력 윈도우의 중심 픽셀이 윤곽선으로 인식되는 결과를 퍼센트로 나타낸다. <그림 5>의 학습 및 시험 패턴에서 처음 100 epoch까지는 인식의 결과가 진동하는 것을 알 수 있다. 이것은 다층퍼셉트론이 학습 패턴을 이용하여 윤곽선 추출을 위한 특징을 추출하는 과정으로 볼 수 있다. 그리고 학습의 후반부는 인식의 결과가 거의 변하지 않는 안정화된 상태에 이르렀다.

<그림 6>은 윈도우 크기가 $w=7$ 인 경우로 윈도우 크기가 증가함에 따라 인식의 결과는 상대적으로 증가함을 알 수 있었다. 이것은 윈도우의 크기가 증가함에 따라 윤곽선 주위의 정보를 다층퍼셉트론이 많이 활용할 수 있음을 나타낸다. 또한 $w=5$ 인 경우와 비교하여 학습의 초기 단계에 인식률의 진동이 적고, 안정화되는 시간이 짧았다.



<그림 5> 학습 및 시험 패턴의 윤곽선 인식(윈도우 크기($w=5$))



<그림 6> 학습 및 시험 패턴의 윤곽선 인식(윈도우 크기($w=7$))

실험 결과, 본 논문에서 제안한 윈도우의 위치 정보를 이용한 영상의 객체 분할은 윈도우의 위치 정보를 이용하지 않는 방법보다 성능면에서 앞선다. 그리고 입력 노드의 증가로 다층퍼셉트론에서 학습 속도 지연을 가져오는 것 보다는 학습 및 시험 패턴에 대한 윤곽선의 인식률이 높고 안정화 단계를 초기에 정착시킬 수 있는 장점을 발견할 수 있었다.

5. 결론

영상의 객체 추출에 사용되는 다층퍼셉트론은 입력 영상을 일정한 크기의 윈도우로 분할하여 학습을 수행한다. 이런 분할된 윈도우는 객체 추출의 기준이 되는 정보를 포함하지 않는다면 다층퍼셉트론의 포화 현상을 일으킬 수 있다. 이와 같은 다층퍼셉트론의 포화 문제는 객체 추출의 기준 정보를 포함하는 학습 패턴의 수와 그렇지 않은 학습 패턴의 수를 동일하게 하여 해결할 수 있었다.

그러나 다층퍼셉트론이 적정하게 학습 데이터를 구성할지라도 그 객체 추출 성능은 매우 낮은 결과를 얻었다. 이것은 다층퍼셉트론의 입력으로 사용되는 입력 윈도우가 각각 독립된 정보만을 가지고 있기 때문이다. 즉, 입력 영상에 대한 입력 윈도우의 위치나 영향력이 고려되지 않기 때문이다.

따라서 본 논문에서는 다층퍼셉트론의 성능을 개선하기 위해 다층퍼셉트론의 입력으로 입력 윈도우의 위치와 영향 정보를 추가하였다. 입력 윈도우의 위치 정보는 전체 입력 영상에 대한 해당 윈도우의 좌표와 거리를 다층퍼셉트론에 전달하며, 영향 정보는 해당 윈도우의 중요도를 다층퍼셉트론에 제공한다. 필기체 숫자를 대상으로 한 실험 결과, 기존 윈도우 정보만 사용하는 방법에 비해 제안된 방법은 평균적으로 약 20%의 성능 개선 결과를 얻을 수 있었으며 개선된 오류역전파학습으로 학습 시간을 단축시킬 수 있음을 확인하였다.

참고문헌

1. 광영태, 이영직, 권오석, (2001). 다층퍼셉트론의 은닉노드 근사화를 이용한 개선된 오류역전파 학습, *정보과학회논문지(B)*, 제28권, 제9호, 603-611.
2. Farrokhnia, F. (1990). *Multi-channel filtering techniques for texture segmentation and surface quality inspection*, Ph.D. Dissertation, Department of Electrical Engineering, Michigan State University.
3. Haddon, J. F. and Boyce, J. F. (1994). Neural networks for texture classification of temporally consistent segmented regions of FLIR sequences, *Proc. of 12th IAPR Int. Conf. on Pattern Recognition, II*, Jerusalem Israel, 1994, 107-111, IEEE Computer Society Press.
4. Hornik, K., Stinchcombe, M. and White, H. (1989). Multilayer feedforward networks are universal approximators, *Neural Networks*, 2, 359-366.
5. Hull, J. J. (1994). A database for handwritten text recognition research,

- IEEE Trans. Pattern and Machine Intell.*, 16, 550-554.
6. Islam, M. M. and Murase, K. (2001). A new algorithm to design compact two-hidden-layer artificial neural networks, *Neural Networks*, 14, 2001.
 7. Kass, M., Witkin, A. and Terzopoulos, D. (1987). Snakes: Active contour models, *Int. J. Computer Vision*, 1, 321-331.
 8. Lengelle, R. and Denoeux, T. (1996). Training MLPs Layer by Layer Using an Objective Function for Internal Representations, *Neural Networks*, 9(1), 83-97.
 9. Lippmann, R. P. (1987). An Introduction to Computing with Neural Nets, *IEEE ASSP Magazine*, 4(2), 4-22.
 10. Middleton, I. and Damper, R. I. (2004). Segmentation of magnetic resonance images using a combination of neural networks and active contour models, *Medical engineering & physics*, 26(1), 71-86.
 11. Petrou, M. and Bostdogianni, P. (1999). *Image Processing : The Fundamentals*, John Wiley & Sons.
 12. Rumelhart, D. E. and McClelland, J. L. (1986). *Parallel Distributed Processing*, MIT Press, Cambridge, MA, 318-362.
 13. Shang, C. and Brown, K. (1994). Principal feature-based texture classification with neural networks, *Pattern Recognition* 27(5), 675-687.
 14. Wang, G. J. and Chen, C. C. (1996). A Fast Multilayer Neural-Network Training Algorithm Based on the Layer-By-Layer Optimizing Procedures, *IEEE Trans. Neural Networks*, 7, 768-775.
 15. Xu, C. and Prince, J. L. (1998). Snake, Shapes, and Gradient Vector Flow, *IEEE Trans on Image Processing*, 7, 359-369.
 16. Yam, Y. F. and Tommy, W. S. Chow (1997). Extended Least Squares Based Algorithm for Training Feedforward Networks, *IEEE Trans. Neural Networks*, 8, 806-810.