Journal of The Korea Society of Computer and Information
Vol. 21 No. 1, pp. 1-8, January 2016
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2016.21.1.001

New Approach to Optimize the Size of Convolution Mask in Convolutional Neural Networks

Young-Tae Kwak *

Abstract

Convolutional neural network (CNN) consists of a few pairs of both convolution layer and subsampling layer. Thus it has more hidden layers than multi-layer perceptron. With the increased layers, the size of convolution mask ultimately determines the total number of weights in CNN because the mask is shared among input images. It also is an important learning factor which makes or breaks CNN's learning. Therefore, this paper proposes the best method to choose the convolution size and the number of layers for learning CNN successfully. Through our face recognition with vast learning examples, we found that the best size of convolution mask is 5 by 5 and 7 by 7, regardless of the number of layers. In addition, the CNN with two pairs of both convolution and subsampling layer is found to make the best performance as if the multi-layer perceptron having two hidden layers does.

▸ Keyword : Convolutional Neural Network, Convolution Mask, Convolution Layer

I. Introduction

전방향 전달 신경회로망(feedforward neural networks)은 계층을 구성하는 방법과 학습 방법에 따라 다양한 형태로 나눌 수 있다. 계층을 구성하는 방법에는 일반적인 다층퍼셉트론(multil-layer perceptron) 구조가 있고[1], 학습이 진행됨에 따라 계층이 늘어나는 cascade correlation 구조가 있다[2]. 또한 학습 방법으로는 간단한 오류역전파 학습과 오차 함수의 기울기 방향에 따라 가중치를 조정하는 RPROP(resilient back-propagation)가 있다[3]. 이외에도 오차 함수의 2차 미분을 이용하는 Conjugate Gradient 방법[4]과 Levenberg-Marquardt 방법[5] 등이 있다.

다양한 분야에 적용되고 있는 다층퍼셉트론은 입력과 출력 노드의 수가 적은 함수 근사화나 제어와 같은 경우에는 우수한 성능을 보이고 있다. 그러나 2차원 영상에서 얼굴 인식, 객체 추출과 같은 응용에서는 입력 노드의 수가 증가함으로 학습 시간이 오래 걸리는 문제점과 2차원 영상을 1차원으로 입력하기 때문에 2차원 영상 정보가 제거되는 문제점을 가지고 있다. 이런 문제점을 해결하기 위해 LeCun은 회선처리 신경회로망(CNN: Convolutional Neural Network)을 제안하였다[6].

LeCun에 의해 제안된 회선처리 신경회로망은 피라미드형 회선처리 신경회로망과 2차원 및 3차원 영상의 패턴 인식 분야에 다양하게 적용되어 왔다[7]. 회선처리 신경회로망은 영상의 특징 추출과 분류를 하나의 구조로 수행하고 있으며 다른 구조의 신경회로망보다 지역적, 기하학적 왜곡에 강한 특징을 가지고 있다. 최근에는 빅데이터 분석을 위한 기계 학습의 일종으로 딥 뉴럴 네트워크에 회선처리 신경회로망이 비교사 학습 형태로 사용되고 있다[8].

회선처리 신경회로망은 두 종류의 계층을 가지고 있는데, 하나는 회선처리 기법을 수행하는 회선처리 계층이며 다른 하나는 영상의 크기를 축소시키는 서브샘플링 계층이다. 회선처리 계층의 회선 마스크는 중심 픽셀과 이웃하는 픽셀 사이의 영상 정보를 수집한다. 그리고 회선 마스크의 크기는 신경회로망의 전체 가중치의 수와 계층의 수를 결정하며, 학습 속도에도 영향을 미치는 중요한 요소이다. 마스크의 크기가 크면 가중치의 수는 증가하나 계산 시간이 오래 걸린다. 그리고 회선처리 신경회로망의 전체 계층의 수가 축소된다. 반면에 그 크기가 작으면 가중치의 수는 적어지고 학습은 어려워지며, 적어진 가중치의

• First Author: Young-Tae Kwak, Corresponding Author: Young-Tae Kwak
*Young-Tae Kwak(ytkwak@jbnu.ac.kr), Dept. of IT and Engineering, Chonbuk National University
• Received: 2015. 11. 03, Revised: 2015. 11. 24, Accepted: 2015. 12. 30.

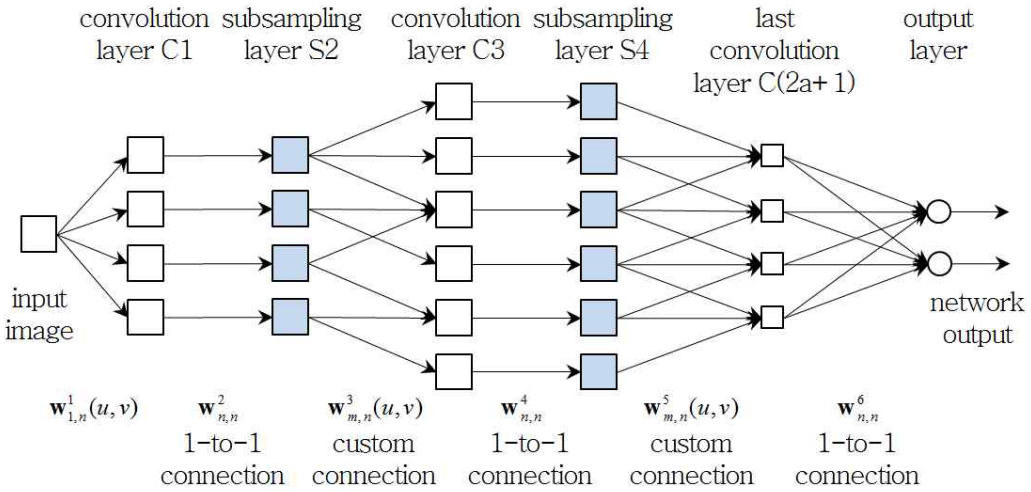

Fig. 1. Network Architecture

수를 보완하기 위하여 전체 계층의 수를 증가시켜야 한다.

지금까지 회선처리 신경회로망을 사용하는 많은 응용에서 학습의 중요 역할을 담당하는 회선처리 마스크의 크기를 단지 추측이나 시행착오를 겪으면서 결정하였다. 또한 회선처리 마스크의 크기 결정에 대한 이론적인 접근이나 분석은 신경회로망이 비선형적인 특성을 지니고 있기 때문에 어느 논문에서도 다루어지지 않았다. 단지 그 추측만이 있을 뿐이었다. 따라서, 본 논문에서는 많은 양의 학습 패턴으로 이루어진 실험을 통하여 회선처리 신경회로망의 계층의 수와 최적의 마스크 크기를 결정하는 방법을 제안한다.

논문의 또 다른 주제로는 회선처리 마스크의 크기에 따른 계층의 수와 구조를 결정하는 방법이다. 회선처리 신경회로망은 회선 계층과 서브샘플링 계층이 쌍으로 존재하여야 하며 다층퍼셉트론보다 상대적으로 많은 계층을 가지고 있다. 다층퍼셉트론의 경우 계층의 수가 증가함에 따라 학습 시간이 오래 걸린다[9]. 회선처리 신경회로망도 다층퍼셉트론과 같은 활성화 함수를 사용하는 경우, 계층이 증가함에 따라 학습이 오래 걸릴 가능성이 높다. 그러므로 본 논문은 회선처리 마스크의 크기에 따른 회선처리 신경회로망의 계층 구성 방법을 제시한다.

논문의 전개는 다음과 같다. 2장에서는 회선처리 신경회로망의 구조, 학습 방법 그리고 마스크 크기와 계층 구성 방법을 소개한다. 3장 실험에서는 얼굴 인식을 대상으로 회선처리 신경회로망에서 계층 수와 마스크의 크기를 변화시킴에 따라 학습 성능을 분석하여 최적 회선처리 마스크를 확인한다. 그리고 마지막으로 결론을 맺는다.

II. Convolutional Neural Networks

1. Network Architecture

회선처리 신경회로망은 그림 1과 같으며 크게 회선처리 계층과 서브샘플링 계층 그리고 출력층으로 구성된다. 회선처리 계층과 서브샘플링 계층은 쌍으로 존재하며 2차원 영상을 입력으로 받는데 이것을 플랜(plane)이라 하고 그 출력을 특징맵(feature map)이라 한다. 그리고 마지막에 있는 회선처리 계층의 출력은 크기가 1×1인 플랜이다. 따라서 출력층은 마지막 회선처리 계층의 1차원 정보를 입력으로 받는다.

각 계층 사이의 연결은 회선처리 계층과 서브샘플링 계층은 1대 1이며, 서브샘플링과 회선처리 계층은 사용자가 연결을 정의할 수 있다. 일반적인 사용자 연결형은 완전 연결형(fully connected)이다.

본 논문에서 사용한 회선처리 신경회로망의 구조와 알고리즘의 표현은 LeCun의 방법을 체계적이고 명료하게 표현한 Phung의 논문을 참고했다[10].

1.1 Convolution Layers

회선처리 계층의 각 플랜은 하나의 회선처리 마스크를 가지며 마스크의 값이 가중치 역할을 한다. 그리고 이런 회선처리 마스크는 하나의 플랜에서 공유된다. 그림 2는 하나의 특징맵이 계산되는 과정을 나타낸다. 그림 2에서 l은 회선처리 계층의 인덱스로서 홀수($l=1,3,\cdots,2a+1$)이다.

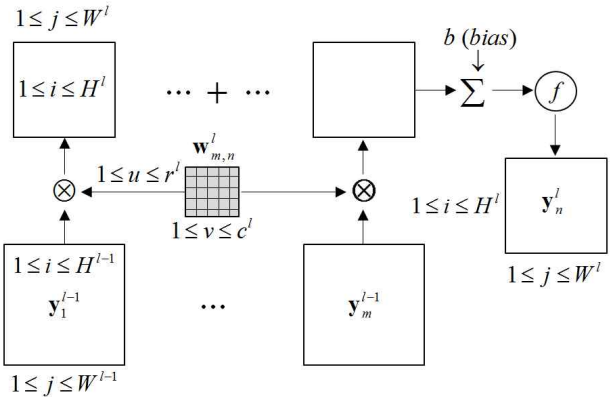

Fig. 2. Convolution Layer

$$y_n^l(i,j)=f^l\left(\sum_{m\in \mathbf{V}_n^l}\left(\sum_{(u,v)\in C} y_m^{l-1}(i+u,\, j+v)w_{m,n}^l(u,\, v)\right)+b_n^l\right)$$
$$\mathbf{y}_n^l=f^l\left(\sum_{m\in \mathbf{V}_n^l}\mathbf{y}_n^{l-1}\otimes \mathbf{w}_{m,n}^l+b_n^l\right) \quad (1)$$

그림 2의 과정을 식으로 표현하면 식(1)이다. 식(1)의 첫 번째는 원소를 기준으로 작성한 것이고 두 번째는 벡터를 기준으로 작성했다. $\mathbf{w}_{m,n}^l=\left\{w_{m,n}^l(u,v)\right\}$은 $l-1$계층 m 특징맵에서 l계층 n 특징맵으로 연결되는 회선처리 마스크로 그 크기는 $r^l\times c^l$이다. $C=\{(u,v)\in N^2|0\le u<r^l, 0\le v<r^l\}$는 회선처리 마스크의 인텍스를 나타낸다.

W^l,H^l은 l계층의 특징맵의 크기이며 각 특징맵의 인텍스는 $P=\{(i,j)\in N^2|1\le i\le W^l, 1\le j\le H^l\}$로 나타낼 수 있다. 또한 $\mathbf{V}_n^l$은 l계층 n 특징맵에 연결되는 $l-1$계층의 모든 플랜을 나타낸다. 식(1)을 수행한 후 l계층의 특징맵($\mathbf{y}_n^l$)의 크기는 $(H^{l-1}-r^l+1)\times(W^{l-1}-c^l+1)$가 된다.

1.2 Subsampling Layers

서브샘플링 계층은 $l=2,4,...,2a$와 같이 짝수 층이며 플랜의 수는 전 층(회선처리 계층)의 플랜 수와 같고 1대 1로 연결되어 있다. 그리고 각 플랜은 하나의 가중치만 존재하며 그림 3은 서브샘플링 계층의 계산 과정을 나타낸다. 우선 서브샘플링 계층은 2차원 입력을 중복되지 않게 2×2크기의 블록으로 나누고 각 블록을 식(2)와 같이 합한다. 그리고 그 결과를 가중치와 곱한 후 바이어스를 추가한 다음 활성화 함수의 입력으로 받아들여 식(3)과 같이 특징맵의 출력을 생성한다. 식(2)와 (3)에서 진한 로마체는 벡터를 의미하며 이후에도 같은 의미로 사용된다.

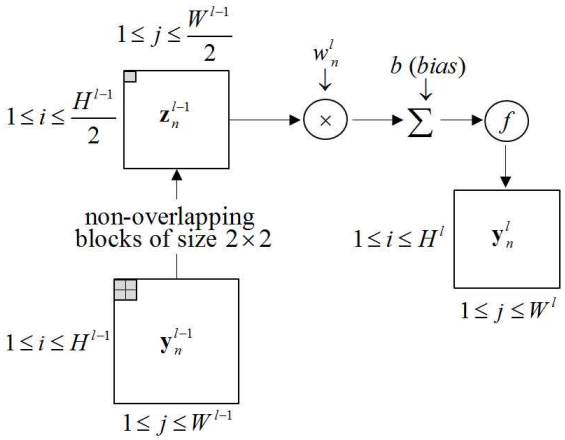

Fig. 3. Subsampling Layer

$$z_n^{l-1}(i,j)=y_n^{l-1}(2i-1,2j-1)+y_n^{l-1}(2i-1,2j)+y_n^{l-1}(2i,2j-1)+y_n^{l-1}(2i,2j)=\mathbf{z}_n^{l-1} \quad (2)$$

$$y_n^l(i,j)=f^l\left(z_n^{l-1}(i,j)\times w_n^l+b_n^l\right)$$
$$\mathbf{y}_n^l=f^l(\mathbf{z}_n^{l-1}\times w_n^l+b_n^l) \quad (3)$$

서브샘플링 계층에서 하나의 특징맵의 크기는 다음과 같다.

$$H^l=H^{l-1}/2,\qquad W^l=W^{l-1}/2 \quad (4)$$

1.3 Other Layers

기타 계층에는 마지막 회선처리 계층과 최종 출력층이 있다. 마지막 회선처리 계층의 각 플랜의 크기는 회선처리 마스크의 크기와 같다. 이것은 마지막 회선처리 계층의 출력을 1×1로 생성하기 위해서다.

최종 출력층은 비선형적인 특성을 반영하기 위하여 활성화 함수로서 시그모이드 함수나 radial-basis 함수를 사용할 수 있으며 그 출력은 식(5)와 같다. 여기서 L은 최종 계층, 즉 전체 출력 층을 의미하고 N^L은 최종 출력층의 노드 수이다.

$$y_n^L=f^L\left(\sum_{m=1}^{N^{L-1}}y_m^{L-1}w_{m,n}^L+b_n^L\right)$$
$$\mathbf{y}=\left[y_1^L,y_2^L,...,y_{N_L}^L\right] \quad (5)$$

2. Learning Algorithm

학습 알고리즘에서 오차를 측정하는 함수는 식(6)과 같은 평균 제곱근 오차함수(Mean Squared Error)를 사용한다. 여기서 $d_n^{L,k}$은 최종 출력층의 n번째 노드의 k번째 목표값이다. 또한 학습 알고리즘은 일괄 학습을 사용한다. 각 계층의 활성화 함수는 모두 동일한 함수를 사용할 수도 있으며, 계층별로 다른 활성화 함수를 사용할 수도 있다.

$$E(\mathbf{w})=\frac{1}{K\times N^L}\sum_{k=1}^{K}\sum_{n=1}^{N^L}\left(y_n^{L,k}-d_n^{L,k}\right)^2 \quad (6)$$

- **출력층** $l=L$

우선 각 층의 가중치를 오차에 따라 수정하기 위해서는 각 층의 오차 신호(error signal)를 계산해야 한다. 이런 오차 신호(δ)는 최종 출력층에서 계산되어 입력층으로 역전파된다. 따라서 먼저 최종 출력층의 오차 신호를 계산하면 다음과 같다. 여기서 $s_n^{L,k}$은 활성화 함수의 입력이다. 즉, 가중치와 입력의 곱에 대한 합이다.

$$e_n^{L,k}=y_n^{L,k}-d_n^{L,k}$$
$$\delta_n^{L,k}=\frac{\partial E}{\partial s_n^{L,k}}=\frac{2}{K\times N^L}e_n^{L,k}f^{L'}\left(s_n^{L,k}\right)$$
$$n=1,2,...,N^L \quad (7)$$

그리고 이런 오차 신호를 이용하여 최종 출력층의 가중치와 바이어스를 계산하면 다음과 같다.

$$\text{weights: } \frac{\partial E}{\partial w_{m,n}^{L}} = \sum_{k=1}^{K} \delta_n^{L,k} y_m^{L-1,k} \quad (8)$$

$$\text{biases: } \frac{\partial E}{\partial b_n^{L}} = \sum_{k=1}^{K} \delta_n^{L,k}$$

• **마지막 회선처리 계층** $l = L-1$

마지막 회선처리 계층($l = L-1$)은 1×1의 특징맵(출력)을 가지므로 오차 신호는 다음과 같다.

$$\delta_n^{l,k}(i,j) = \delta_n^{l,k}(1,1) = \delta_n^{l,k} \quad (9)$$
$$= \frac{\partial E}{\partial s_n^{l,k}} = f^{l\prime}(s_n^{l,k}) \sum_{q=1}^{N^{l+1}} \delta_q^{l+1,k} w_{n,q}^{l+1}$$

여기서 $w_{n,q}^{l+1}$은 오차 역전파 시 n에서 q로 연결되어 있다. 즉 n은 $l+1$이다. 이것은 출력 값 계산 시 $w_{m,n}^{l}$이 m에서 n으로 연결된 것과 같은 형태이다. 마지막 회선처리 계층의 가중치와 바이어스의 수정은 다음과 같다.

$$\text{weights: } \frac{\partial E}{\partial w_{m,n}^{l}(u,v)} = \sum_{k=1}^{K} \left[\delta_n^{l,k} y_m^{l-1,k}(1+u, 1+v) \right] \quad (10)$$
$$= \sum_{k=1}^{K} \left[\delta_n^{l,k} y_m^{l-1,k}(u,v) \right]$$

$$\text{biases: } \frac{\partial E}{\partial b_n^{l}} = \sum_{k=1}^{K} \delta_n^{l,k}$$

마지막 회선처리 계층의 특징맵의 크기가 1이므로 $i=1, j=1$이다. 그러므로 식(10)의 첫 번째 식은 $i=1, j=1$인 경우를 나타내고 이것은 두 번째 식으로 간략화 할 수 있다. 또한, y_m^{l-1}의 크기(H^l, W^l)는 $r^l \times c^l$의 크기와 같다.

• **마지막 서브샘플링 계층** $l = L-2$

마지막 서브샘플링 계층($l = L-2$)은 단지 하나의 가중치만 가지고 있으므로 오차 신호는 식(11)과 같다.

$$\delta_n^{l,k}(i,j) = \frac{\partial E}{\partial s_n^{l,k}(i,j)} \quad (11)$$
$$= f^{l\prime}\left[s_n^{l,k}(i,j) \right] \sum_{q=1}^{N^{l+1}} \delta_q^{l+1,k} w_{n,q}^{l+1}(i,j)$$

그리고 가중치와 바이어스의 조정은 다음과 같다.

$$\text{weights: } \frac{\partial E}{\partial w_{m,n}^{l}} = \sum_{k=1}^{K} \left[\sum_{(i,j)\in P} \delta_n^{l,k}(i,j) \sum_{(u,v)\in\{0,1\}^2} y_m^{l-1,k}(2i+u, 2j+v) \right] \quad (12)$$
$$= \sum_{k=1}^{K} \left[\sum_{(i,j)\in P} \delta_n^{n,k}(i,j) z_n^{l-1,k}(i,j) \right]$$

$$\text{biases: } \frac{\partial E}{\partial b_n^{l}} = \sum_{k=1}^{K} \left[\sum_{(i,j)\in P} \delta_n^{l,k}(i,j) \right]$$

• **일반 회선처리 계층** $l = 2a+1$

그 밖의 다른 회선처리 계층($l = 2a+1$)은 가중치($w_{n,n}^{l+1}$)의 출력에 서브샘플링 계층이 1대 1로 연결되어 있으므로 오차 신호는 식(13)과 같다.

$$\delta_n^{l,k}(i,j) = \frac{\partial E}{\partial s_n^{l,k}(i,j)} \quad (13)$$
$$= f^{l\prime}\left[s_n^{l,k}(i,j) \right] \delta_n^{l+1,k}(\lfloor i/2 \rfloor, \lfloor j/2 \rfloor) w_{n,n}^{l+1}$$

이런 오차 신호에 의한 가중치와 바이어스의 조정은 식(14)에 의해 수행된다.

$$\text{weights: } \frac{\partial E}{\partial w_{m,n}^{l}(u,v)} = \sum_{k=1}^{K} \left[\sum_{(i,j)\in P} \delta_n^{l,k}(i,j)\, y_m^{l-1,k}(i+u, j+v) \right] \quad (14)$$

$$\text{baises: } \frac{\partial E}{\partial b_n^{l}} = \sum_{k=1}^{K} \left[\sum_{(i,j)\in P} \delta_n^{l,k}(i,j) \right]$$

일반 회선처리 계층과 마지막 회선처리 계층을 비교했을 때 마지막 회선처리 계층은 서브샘플링 계층에 연결되지 않고 $i=1, j=1$인 특수한 경우로 식(10)이 된다.

• **일반 서브샘플링 계층** $l = 2a$

다른 서브샘플링 계층($l = 2a$)은 각 플랜이 오직 하나의 가중치($w_{n,n}^{l}$)를 가지기 때문에 오차 신호는 다음과 같다.

$$\delta_n^{l,k}(i,j) = \frac{\partial E}{\partial s_n^{l,k}(i,j)} = f^{l\prime}\left[s_n^{l,k}(i,j) \right] \times \quad (15)$$
$$\left(\sum_{q=1}^{N^{l+1}} \sum_{(u,v)\in C} \delta_q^{l+1,k}(i+u, j+v) w_{n,q}^{l+1}(u^*, v^*) \right)$$
$$u^*, v^* : \text{indeces of } u, v \text{ rotated by } 180$$

여기서 u^*, v^*는 180도 회전된 u, v의 인텍스이다. 그리고 가중치와 바이어스의 조정은 식(16)과 같다.

$$\text{weights: } \frac{\partial E}{\partial w_{n,n}^{l}} = \sum_{k=1}^{K} \left[\sum_{(i,j)\in P} \delta_n^{l,k}(i,j) \sum_{(u,v)\in\{0,1\}} {}_{2}\, y_m^{l-1,k}(2i+u, 2j+v) \right] \quad (16)$$
$$= \sum_{k=1}^{K} \left[\sum_{(i,j)\in P} \delta_n^{l,k}(i,j) z_n^{l-1,k}(i,j) \right]$$

$$\text{biases: } \frac{\partial E}{\partial b_n^{l}} = \sum_{k=1}^{K} \left[\sum_{(i,j)\in P} \delta_n^{l,k}(i,j) \right]$$

식(15)의 오차 신호를 마지막 서브샘플링 계층 식(11)과 비교했을 때 $\delta_q^{l+1,k}(i,j) w_{n,q}^{l+1}(u,v)$에서 $i=1, j=1$로 스칼라 값이고 u, v또한 i, j와 크기가 같다.

3. Proposed Algorithm for Constructing CNN

다층퍼셉트론은 입력층과 출력층이 사전에 결정되고 응용에 따라 은닉층의 수와 노드 수를 결정하지만, 회선처리 신경회로망은 다르다. 회선처리 신경회로망은 회선처리 마스크가 입력에 대해 공유되기 때문에, 최종 출력층로 부터 시작하여 회선처리 계층과 서브샘플링 계층을 하나의 쌍으로 추가하면서 입력층까지 각 계층의 입력 크기와 계층의 수를 결정한다. 이와 같이 계층의 수가 결정되면 입력층의 크기가 최종 결정된다. 따라

서 본 논문은 각 계층의 회선처리 마스크의 크기에 따른 입력 영상의 크기를 결정하는 다음 알고리즘을 제안한다.

1. 최종 출력층의 출력 크기를 결정한다. 여기서는 1×1이다.

$$H^L \times W^L = 1 \times 1$$

2. 마지막 회선처리 계층의 출력이 1×1되도록 한다. 이와 같은 출력이 나오도록, 적용하고자 하는 회선처리 마스크 ($r \times c$)를 결정한다.

$$H^{L-1} \times W^{L-1} = 1 \times 1$$

- 마지막 서브샘플링 계층의 출력이 마지막 회선처리 계층의 마스크와 크기가 같은지 확인한다.

$$H^{L-2} \times W^{L-2} = r \times c$$

3. 나머지 계층은 서브샘플링 계층과 회선처리 계층을 하나의 쌍으로 처리하여 입력층까지 추가한다.
- 회선처리 계층의 출력은 다음에 연결되는 서브샘플링 계층의 출력의 2배가 되도록 한다.

$$H^l \times W^l = 2(H^{l+1} \times W^{l+1})$$

- 서브샘플링 계층의 출력은 다음에 연결되는 회선처리 계층의 출력에 마스크의 크기보다 하나 적은 값을 더한다.

$$H^l \times W^l = (H^{l+1} + r - 1) \times (W^{l+1} + c - 1)$$

4. 입력층의 크기는 최초의 회선처리 계층의 출력에 마스크의 크기보다 하나 적은 값을 더한다.

$$H^{Input} \times W^{Input} = (H^1 + r - 1) \times (W^1 + c - 1)$$

5. 입력층의 크기와 계층의 수가 결정된 후, 응용 문제의 복잡도에 따라 각 계층의 플랜 수를 결정한다. 플랜 수에 의해 최종 가중치의 수가 결정된다.

III. Experiments

신경회로망의 가중치 수는 응용에 따라 또는 문제의 복잡도에 따라 다르고 이것을 이론적으로 결정하는 것은 어렵다. 따라서 많은 응용 시스템에서는 시행착오를 겪으면서 가중치의 수를 결정한다. 회선처리 신경회로망 또한 여러 번의 실험을 거쳐 적정한 가중치의 수를 결정한다. 이런 회선처리 신경회로망에서 가중치의 수는 회선처리 마스크의 크기가 결정하므로 본 실험에서는 학습 데이터가 많은 얼굴 인식을 대상으로 결과를 분석하여 최적의 마스크의 크기와 계층의 수를 제안한다.

여기서 얼굴 인식은 얼굴이 누구인지 인식하는 것이 아니라, 디지털 영상에서 일정 영역이 얼굴인지 아닌지를 구분하는 것이다. 논문에서 사용한 영상 데이터베이스는 Phung이 사용한 데이터베이스이다[11]. 그림 4는 실험 영상의 예이다. 학습으로는 1000개의 얼굴 영상과 1000개의 비얼굴 영상을 사용하였다. 이런 비얼굴 영상은 그림 4의 (b)와 같이 다양한 종류의 이미지를 포함하고 있어 제안한 방법이 얼굴 영상뿐만 아니라 다양한 이미지에도 적용될 수 있음을 의미한다. 학습 결과를 테스트하기 위해 시험 영상으로 10000개의 영상을 사용하였다. 이런 시험 영상에도 얼굴 영상과 비얼굴 영상이 포함되어 있다. 각 영상의 기본 크기는 32×32이다.

(a) Face Images

(b) Non-face Images

Fig. 4. Examples of Training Images

Phung은 그림 4와 같은 실험 영상을 가지고, 하나의 회선처리 신경회로망에서 각 회선처리 계층의 마스크 크기를 다르게 하여 실험하였다. 이런 구조는 시행착오를 하여 얻은 결과였다. 그러나 각 회선처리 계층의 마스크 크기를 다르게 하면, 마스크 크기에 따른 비교가 어려우므로 본 논문에서는 각 회선처리 계층의 마스크를 동일하게 하고 3×3, 5×5, 7×7, 9×9, 11×11의 경우로 나누어 실험하였다. 그 결과가 표 1에 있다.

표 1의 내용을 설명하기 위해 기준 마스크인 (b) 5×5 마스크를 가지고 설명하면 다음과 같다. 계층은 입력층과 6개의 계층으로 구성되고 마지막은 'F' 계층은 최종 출력층을 의미한다. 그리고 출력 픽셀은 각 계층의 출력으로 생성되는 이미지(특징맵)의 크기이다. '특징맵(마스크)'은 각 계층이 '(마스크)'의 크기를 이용하여 출력하는 2차원 이미지의 수이다. 이것은 다음 계층의 입력으로 들어가는 플랜에 해당한다. 가중치는 가중치와 바이어스의 수를 합으로 표현했다. 예를 들어 'C3' 계층의 경우, 'S2'의 특징맵은 2이고 'C3'의 특징맵은 5인 경우, 이런 연결이 완전 연결(fully connected)되어 있으므로 10개의 회선처리 마스크가 필요하다. 이런 회선처리 마스크는 5×5=25이므로 총 250개의 가중치를 가지며 5개의 특징맵은 하나의 바이어스를 가지고 있으므로 '250+2'가 된다. 이런 가중치를 모두 합하면 5×5 마스크를 사용하는 회선처리 신경회로망은 총 576개의 가중치를 가진다.

2장 3절의 제안한 알고리즘에 따라 구성된 예로 (b) 5×5 마스크를 가지는 회선처리 신경회로망을 설명하면 다음과 같다. 우선 출력층 'F6'은 입력층 영상이 얼굴인지 비얼굴인지 구분하기 위하여 하나의 출력 노드를 가진다. 그리고 마지막 회선처리 계층 'C5'는 출력층 'F6'에 연결되기 위해 하나의 출력을 생성해야 한다. 즉, 1×1 된다. 또한 마지막 서브샘플링 계층 'S4'의 출력은 'C5'의 회선처리 마스크와 크기가 같아야 하므로 5×5이다. 이것은 'S4'의 5×5 출력과 'C5'의 마스크 5×5가 회

Table 1. Leaning Results of Each Mask

CNN	Layer	Input	C1	S2	C3	S4	C5	S6	C7	F8	Weights Sum
	Output Pixel	38×38	36×36	18×18	16×16	8×8	6×6	3×3	1×1	1×1	
	Feature Map (Mask)	1	4(9)	4	4(9)	4	4(9)	4	4(9)	1	513
	Weights		36+4	4+4	144+4	4+4	144+4	4+4	144+4	4+1	484+29
Trial	1	2	3	4	5	6	7	8	9	10	Average
Learn	87.5	81.65	96.05	50.00	93.75	84.35	72.85	92.2	92.85	92.35	88.17
Test	84.52	76.35	90.31	50.00	89.43	81.90	70.37	87.13	88.95	86.76	83.97

(a) 3×3 Mask

CNN	Layer	Input	C1	S2	C3	S4	C5	F6			Weights Sum
	Output Pixel	32×32	28×28	14×14	10×10	5×5	1×1	1×1			
	Feature Map (Mask)	1	2(25)	2	5(25)	5	2(25)	1			576
	Weights		50+2	2+2	250+5	5+5	250+2	2+1			559+17
Trial	1	2	3	4	5	6	7	8	9	10	Average
Learn	85.55	93.50	96.90	95.05	93.50	96.80	97.65	95.70	98.90	94.45	94.80
Test	82.44	91.44	91.92	91.60	89.38	92.51	92.58	91.11	93.25	89.33	90.56

(b) 5×5 Mask

CNN	Layer	Input	C1	S2	C3	S4	C5	F6			Weights Sum
	Output Pixel	46×46	40×40	20×20	14×14	7×7	1×1	1×1			
	Feature Map (Mask)	1	2(49)	2	2(49)	2	3(49)	1			607
	Weights		98+2	2+2	196+2	2+2	294+3	3+1			595+12
Trial	1	2	3	4	5	6	7	8	9	10	Average
Learn	97.95	88.90	97.25	95.30	86.65	96.45	97.35	98.25	97.90	92.10	94.81
Test	92.74	83.99	92.84	92.14	81.40	90.31	91.58	93.92	92.45	88.49	89.99

(c) 7×7 Mask

CNN	Layer	Input	C1	S2	C3	S4	C5	F6			Weights Sum
	Output Pixel	60×60	52×52	26×26	18×18	9×9	1×1	1×1			
	Feature Map (Mask)	1	1(91)	1	2(91)	2	2(91)	1			581
	Weights		81+1	1+1	162+2	2+2	324+2	2+1			572+9
Trial	1	2	3	4	5	6	7	8	9	10	Average
Learn	95.75	50.00	50.00	97.15	97.10	50.00	96.40	81.75	96.50	96.10	86.56
Test	93.38	50.00	50.00	92.36	92.97	50.00	91.89	75.97	92.64	92.51	83.14

(d) 9×9 Mask

CNN	Layer	Input	C1	S2	C3	F4					Weights Sum
	Output Pixel	32×32	22×22	11×11	1×1	1×1					
	Feature Map (Mask)	1	3(121)	3	1(121)	1					738
	Weights		363+3	3+3	363+1	1+1					730+8
Trial	1	2	3	4	5	6	7	8	9	10	Average
Learn	50.00	97.05	72.40	50.00	94.65	50.00	50.00	94.00	50.00	77.30	87.08
Test	50.00	92.95	69.05	50.00	91.52	50.00	50.00	89.27	50.00	74.00	83.36

(e) 11×11 Mask

선처리되어 1×1의 출력을 생성하기 때문이다. 'C3'의 출력은 'S4'의 입력되기 위해 2배, 10×10이 되어야 하며, 'S2'는 'C3'과 회선처리를 위하여 (10-(마스크 크기-1))×(10-(마스크 크기-1))가 되어 14×14가 된다. 다시 'C1'은 'S2'의 2배가 되어야하므로 28×28 된다. 그리고 최종 입력층의 크기는 'C1'의 출력 크기에 마스크의 크기보다 1이 작은 만큼의 영상이 되므로 32×32가 된다. 이와 같이 계층에 따른 영상의 크기가 결정된 후, 각 계층의 특징맵(또는 플랜)의 수를 결정하여 전체 가중치의 수를 결정한다.

각 회선처리 마스크에 따른 총 가중치의 수는 다른데 이것은 회선처리 마스크의 배수에 의해 발생한다. 따라서 논문의 실험에서는 기준 마스크(5×5)의 총 가중치 수와 최대한 유사한 수의 가중치를 가지도록 계층의 수와 특징맵의 수를 조정하여 각 회선처리 신경회로망의 구조를 구성하였다. 계층의 수가 달라짐에 따라 입력층의 영상 크기도 달라진다. 이것은 2장 3절에서 제시한 방법에 따라 입력층의 영상 크기를 결정하였다. Phung은 다양한 크기의 영상을 제공하지 않기 때문에, 1로 초기화된 영상의 중앙에 32×32의 원본 영상을 위치시켜 다양한 크기의 입력 영상을 생성하였다.

학습 알고리즘으로는 RPROP법을 사용하였으며 학습 완료는 epoch가 1000이 될 때 까지 수행하였다. 여기서 epoch란 전체 학습 패턴이 한 번의 학습을 수행했을 때를 1로 보는 학

습 횟수의 단위이다. 학습은 각 마스크를 10씩 시도하여 그 결과를 인식률(%)로 표시하였고 시험 패턴에 대한 인식률도 나타냈다. 여기서 인식률이 50%이하이면 실패로 간주하고 회색으로 표시하였다. 우측 평균 인식률은 실패한 시도를 제외한 평균 인식률이다.

표 1의 결과, 학습 실패가 없고 총 가중치의 수가 적정한 5×5, 7×7의 회선처리 마스크가 가장 최적의 결과를 얻었다. 우선, 회선처리 마스크가 3×3인 (a)의 결과는 비록 총 가중치의 수가 기준 마스크보다 적어 학습이 실패할 가능성이 크지만 회선처리 마스크의 크기가 작기 때문에 계층의 수를 증가 시켜야 한다. 이것은 학습 시간을 오래 걸리게 하고 학습 실패의 가능성을 높일 수 있다. 따라서 학습 시간을 단축시키는 대안으로는 서브샘플링 계층의 활성화 함수를 선형 함수로 사용하는 것을 추천할 수 있다.

회선처리 마스크의 크기가 9×9 또는 11×11의 경우는 총 가중치의 수가 유사하거나 많아 학습 성공률이 높을 것 같지만 중간 계층의 수나 특징맵의 수가 작아 학습에 실패하는 경우가 많았다. 이것은 회선처리 마스크를 너무 크게 하면, 특징맵의 수가 줄어들고 서브샘플링 계층과 회선처리 계층 사이의 특징맵의 정보를 교환하는 연결 방법이 제한적이 되므로 영상의 2차원 정보를 활용하지 못하는 것으로 분석된다.

학습 시간은 총 가중치의 수가 크고 마스크의 크기가 큰 순서대로 오래 걸렸다. 즉, 11×11, 9×9, 7×7 순으로 오래 걸렸고, 3×3과 5×5의 마스크 크기에서는 3×3의 입력 이미지의 크기가 크기 때문에 5×5 마스크보다 오래 걸렸다. 따라서 입력층의 이미지 크기와 계층의 수를 고려한 회선처리 신경회로망에서 최적의 회선처리 마스크의 크기는 5×5, 7×7이다. 그리고 최종 출력층과 마지막 회선처리 계층을 제외한 나머지 중간층은 회선처리 계층과 서브샘플링 계층이 쌍으로 추가되어야 하는데 이런 계층의 쌍은 1개 또는 2개가 적당한 것으로 실험에서 확인되었다. 이것은 다층퍼셉트론의 은닉층의 수가 1개 또는 2개가 학습에 적절하다는 것과 같은 의미를 지닌다.

IV. Conclusions

지금까지 회선처리 신경회로망을 사용하는 많은 응용에서 학습의 중요 역할을 담당하는 회선처리 마스크의 크기를 단지 추측이나 시행착오를 겪으면서 결정하였다. 이것은 비선형적인 회선처리 신경회로망을 이론적으로 접근하거나 분석하기 어렵기 때문이다. 따라서 본 논문은 많은 학습 패턴과 시도를 통하여, 회선처리 마스크의 크기와 학습 성능의 관계를 확인하였고 그 결과 회선처리 마스크의 크기가 5×5나 7×7 일 때, 최적의 성능을 발휘한다는 결론을 얻었다. 이것은 지금까지의 추측이나 시행착오를 거쳐 결정한 구조를 실험을 통하여 구체적으로 확인한 것이다.

논문의 또 다른 제안 알고리즘으로는 회선처리 마스크의 크기에 따라 신경회로망의 구조를 결정하는 방법을 제안하였다. 제안된 알고리즘은 회선처리 마스크의 크기에 따라 전체 가중치의 수를 결정하고, 각 계층의 입력 영상의 크기, 특징 맵의 수 등을 결정하여 회선처리 신경회로망을 구조화하는 방안을 제안했다. 제안한 알고리즘을 통한 실험에서 회선처리 계층과 서브샘플링 계층의 2개의 쌍으로 구성된 구조가 가장 좋은 성능을 보였다. 이것은 다층퍼셉트론의 경우 은닉층의 수가 2개인 경우 학습이 잘 수행되는 것과 유사하다.

연구 결과는 향후 회선처리 신경회로망을 실제적인 응용에 적용하고자 하는 경우, 회선처리 마스크의 크기를 결정할 수 있는 가이드라인을 제시하고 있으며, 추측과 시행착오에 의해 발생할 수 있는 학습의 문제점을 해결할 수 있는 장점이 있다.

REFERENCE

[1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” in Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Cambridge, MA: Bradford Books, vol. I, pp. 318-362, 1986.

[2] S. E. Fahlman and C. Lebiere, “The cascade correlation learning architecture,” Neural Information Processing System 2, D. S. Touretzsky, ed. Morgan Kaufman, pp. 524-532, 1990.

[3] M. Riedmiller and H. Braun, “A direct adaptive method of faster backpropagation learning: The RPROP algorithm,” in Proc. IEEE Int. Conf. Neural Netw., San Francisco, CA, pp. 586-591, 1993.

[4] E. K. P. Chong and S. H. Zak, An Introduction to Optimization. New York: Wiley, 1996.

[5] M. T. Hagan and M. B. Menhaj, “Training feedforward networks with the Marquardt algorithm,” IEEE Trans. Neural Netw., vol. 5, no. 6, pp. 989-993, Nov. 1994.

[6] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel, “Handwritten digit recognition with a back-propagation network,” in Touretzky, David (Eds), Advances in Neural Information Processing Systems (NIPS 1989), 2, Morgan Kaufman, Denver, CO. 1990.

[7] Lawrence, S., Giles, C.L., Ah Chung Tsoi, Back, A.D., "Face recognition: a convolutional neural-network approach," IEEE Trans. Neural Netw., vol. 8, no. 1, pp. 98-113, Jan 1997.

[8] LeCun, Yann, Koray Kavukcuoglu, and Clément Farabet. "Convolutional networks and applications in vision." Circuits and Systems (ISCAS), Proceedings of 2010 IEEE

International Symposium on. IEEE, 2010.
[9] J. Villiers and E. Barnard, “Backpropagation Neural Nets with One and Two Hidden Layers,” IEEE Trans. Neural Netwoks, vol. 4, no. 1, pp. 136-141, 1993.
[10] S. L. Phung and A. Bouzerdoum, ”MATLAB library for convolutional neural network,” Technical Report, ICT Research Institute, Visual and Audio Signal Processing Laboratory, University of Wollongong. Available at: http://www.uow.edu.au/~phung.
[11] S. L. Phung, A. Bouzerdoum, and D. Chai, “Skin segmentation using color pixel classification: analysis and comparison,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 1, pp. 148-154, 2005.

Authors

Young-Tae Kwak received the B.S., M.S., and Ph.D. degrees in computer engineering from the Chungnam National University, Republic of Korea, in 1993, 1995, and 2001, respectively. He joined the faculty of the Chonbuk National University in 2002. His research interests include pattern recognition and neural networks.