

論文95-32B-3-13

단층 신경망과 이중 기각 방법을 이용한 문자인식

(Single-Layer Neural Networks with Double Rejection Mechanisms for Character Recognition)

林 俊 喲 *, 蔡 洙 翊 *

(Joonho Lim, and Soo-ik Chae)

요 약

오차 역전파(error backpropagation) 알고리즘으로 학습시킨 다층 신경망은 패턴 인식 문제에 가장 널리 쓰이는 신경망이다. 패턴 인식의 많은 응용 분야에서는 인식율을 높이는 것보다는 오인식을 최소화하는 것이 더 현실적이고 중요한 문제이다. 그러나, 다층 신경망은 학습 속도가 느리고 국부 극소점에 빠지는 한계를 가지고 있다. 본 논문에서는, 단층 신경망과 이중 기각 방법을 이용하여 매우 낮은 오인식을 보장하면서 현실적인 문자 인식을 하는 새로운 방법들을 제안한다. 제안하는 신경망은 다층 신경망의 한계들을 극복하고 빠른 학습 능력을 보이며 간단한 하드웨어로 구현이 가능하다. 그리고, 오인식을 최소화하기 위해 새로운 코딩 방법을 도입하였다. 제안된 방법의 효율성을 검증하기 위하여 135,000개의 숫자와 117,000개의 영문자의 데이터 베이스를 구축하여 인쇄체 숫자 및 영문자의 인식 능력을 테스트하였으며, 그 결과에 의하면 오인식은 아주 적게 줄일 수 있고 인식율은 높게 유지함을 알 수 있었다.

Abstract

Multilayer neural networks with backpropagation learning algorithm are widely used for pattern classification problems. For many real applications, it is more important to reduce the misclassification rate than to increase the rate of successful classification. But multilayer perceptrons(MLP's) have drawbacks of slow learning speed and false convergence to local minima. In this paper, we propose a new method for character recognition problems with a single-layer network and double rejection mechanisms, which guarantees a very low misclassification rate. Comparing to the MLP's, it yields fast learning and requires a simple hardware architecture. We also introduce a new coding scheme to reduce the misclassification rate. We have prepared two databases: one with 135,000 digit patterns and the other with 117,000 letter patterns, and have applied the proposed method for printed character recognition, which shows that the method reduces the misclassification rate significantly without sacrificing the correct recognition rate.

* 正會員, 서울大學校 電子工學科 및 半導體共同研究所 (ter)

(Dept. of Elec. Eng., Seoul Nat'l Univ. & 接受日字: 1994年 1月 18日)

Inter-University Semiconductor Research Cen

I. 서론

패턴 인식에 가장 널리 쓰이는 신경 회로망은 오차 역전파(error backpropagation) 알고리즘으로 학습시킨 다층 신경망(multilayer perceptrons)이다. 그러나 다층 신경망은 학습속도가 느리고, 국부 극소점에 빠지는 한계를 가지고 있다. 단층 신경망을 이용한 연구 사례를 보면, STEPNET이라는 구조와 단층 신경망 학습을 이용하여 필기체 숫자를 인식한 연구가 보고되었는데,^[1] 그 구조에서는 각 출력 노드를 임의의 두개의 문자를 구별하도록 학습을 시킨 것으로 다층 신경망에 비하여 적은 오인식과 높은 인식 능력을 보였다. 또한 독립적으로 학습된 두개의 단층 신경망을 연결하여 인쇄체 문자의 인식을 실험한 연구 결과도 발표된 바 있다.^[2] 다층 신경망의 은닉층이 각 패턴 집단을 구분하는데, 네트워크의 분포가 모호하여 완벽히 패턴들을 구분할 수 없음을, 단층 신경망과 비교하여 실험으로 증명한 것도 있었다.^[3]

본 논문에서는, 단층 신경망(single-layer perceptron)과 이중 기각 방법을 사용하여 오인식 없이 문자를 인식하는 구조와 방법을 제안한다. 현실적인 인식 문제-수표 인식, 우편번호 인식 등에 있어서 오인식은 가장 큰 문제점이며, 이 경우 오히려 기각처리를 하여 사람이 직접 분류를 하거나 다른 방법을 이용하는 것이 더 합리적이라고 할 수 있다.^[1] 본 논문에서는 이런 취지로 인식율을 높이고 오인식을 극소화시키는 방법에 초점을 맞추었다. 제 II장에서는, 효과적인 학습을 위한 단층 신경망 학습 방법의 전처리 과정을 설명한다. 학습 패턴들을 가까운 거리에 있는 패턴들끼리 집단으로 분류하고, 각 집단의 중심값을 학습시의 대표값으로 선정하였다. 제 III장에서 제안하는 신경망 구조를 소개한다. 이 구조는 단층 신경망과 간단한 논리회로의 연결로 이루어져 있으며, 신경망의 각 출력 노드는 기각 영역에 의해 기각을 할 수 있도록 구성이 된다. 제 IV장에서는 선형 분리가 가능하며, 효과적인 학습을 가능하게 하는 코딩 방법을 제안한다. 제 V장에서는 오인식을 극소화시키기 위한 방법으로 기각 영역에 의한 기각 방법을 제안하였으며, 이 기각 방법의 자세한 해석과 분석을 하였다. 제 VI장에서는 여러 분석의 예측 및 모의 테스트의 결과를 비교하여 본다. 또한 실제로 하드웨어 구현시 부딪히는 가중치 양자화 문제에 대한 분석 및 결과도 제시하였다. 본문에서의 해석 부분에 대한 간단한 증명들이 부록에 포함되어 있다.

II. 단층 신경망 학습 방법의 전처리 과정

전처리에서 학습까지의 과정은 그림 1과 같다. 전처

리 과정에는 핸드 스캐너로 읽어들이는 데이터를 이진화 및 정규화 하는 과정, 학습 패턴들을 몇 개의 집단으로 나누는 과정, 그리고 각 집단들의 화소(pixel)별 평균을 하여 각 집단들을 대표하는 값들을 구하는 과정이 포함된다.

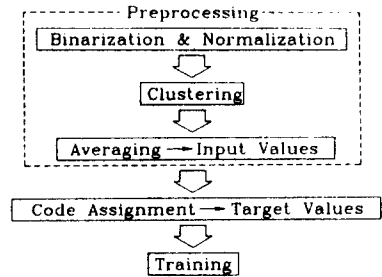


그림 1. 전처리에서 학습까지의 과정

Fig. 1. Procedure from preprocessing to training.

1. 데이터 베이스의 구축과 이진화 및 정규화 (Binarization and Normalization)

학습 및 테스트에 사용되는 패턴들은 핸드 스캐너로 읽어들이어 흑백으로 이진화된 패턴들이다. 입력 패턴으로 인쇄체 숫자와 영문자를 사용하였고, 사용한 폰트는 고딕체이며, 여러 크기로 하여 레이저 프린터로 출력을 하였다. 잡음을 임의로 주기 위하여 출력시 폰트의 굵기를 여러가지로 하였으며, 프린터 토너의 농도를 다양하게 하여 출력을 하였고 이를 복사기를 사용하여 여러 농도로 복사를 하여 데이터 베이스를 구축하였다. 스캐너로 읽어들이는 패턴들은 실제 생활에서 많이 접할 수 있는 잡음(noise) -핸드 스캐너로 스캐닝 할 때의 속도 및 연속성, 문자가 인쇄 또는 복사되어 있는 종이 질의 차이, 복사로 인하여 흐려진 문자들, 그리고 분할 및 정규화 등 여러 이유들 때문에 발생한 잡음 등을 가지고 있다. 정규화는 분리된 문자를 일정한 크기의 패턴으로 바꾸는 과정이다.

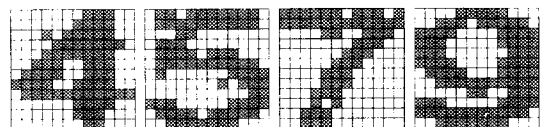


그림 2. 이진화 및 정규화된 숫자들의 예

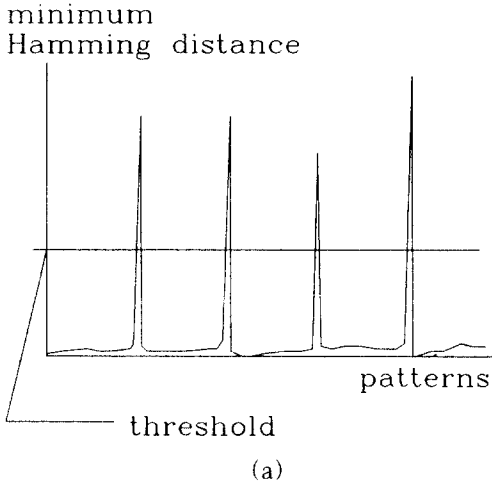
Fig. 2. Examples for normalized digit patterns.

본 논문에서는 이러한 과정들을 거쳐서 구축된 135,000개의 숫자와 117,000개의 영문자를 데이터 베이스로 사용하였으며, 학습 패턴들은 이들 중에서 임의로 5,000개씩을 선택하였고 나머지는 테스트용으로 사

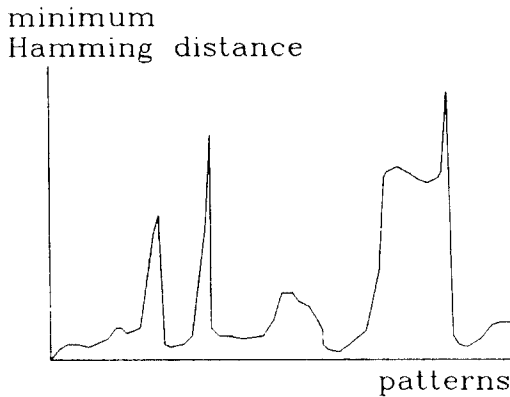
용하였다. 그림 2는 숫자 데이터 베이스의 몇 예를 보인 것이다.

2. 집단화(Clustering)

학습을 용이하게 하기 위하여 많은 학습 패턴들을 상호 유사성이 큰 것들끼리 모아서 몇 개의 집단(cluster)으로 나눈다. 각 집단은 한 개의 문자만을 가리키고, 한 문자가 여러 집단으로 나누어질 수도 있다. 이 집단화는 다음과 같은 순서의 최소 해밍 거리(minimum Hamming distance) 방법을 이용한다.



(a)



(b)

그림 3. 해밍 거리가 가까운 순으로 패턴들을 재정렬한 결과
(a) 잡음이 적은 경우 (b) 잡음이 큰 경우

Fig. 3. Results of resorting the patterns based on the Hamming distance
(a) with small noise and (b) with large noise.

먼저, 기준이 되는 한 패턴을 선택한 후 이 패턴과 나머지 패턴들과의 거리를 계산하여 그 값이 최소인

한 패턴을 선택한다. 그 다음, 기준 패턴과 선택된 패턴에 대한 나머지 패턴들과의 최소 해밍 거리를 계산하여 그 값이 최소인 패턴을 다시 한 개를 선택한다. 이런 방법으로 계속 선택하면서 선택된 순서대로 패턴들을 나열하여 보면 일반적으로 그림 3과 같은 형태로 재정렬이 된다.

그림 3(a)의 경계값(threshold)은 재정렬된 패턴들을 몇 개의 집단으로 나누는데, 즉, 최소 해밍 거리가 경계값보다 크면, 그 위치의 오른쪽 패턴들은 왼쪽의 집단과 다른 집단으로 분류된다. 패턴들의 잡음이 적다면 재배열된 패턴들은 그림 3(a)와 같이 선명하게 구별되는 몇 개의 첨두치(peak)를 가지며, 이 첨두치들에 의해 나누어진 집단의 수는 문자의 수와 같게 된다. 그러나, 패턴들의 잡음이 커지면 그림 3(b)와 같이 모호한 첨두치를 가지며, 제안한 집단화 방법이 성공할 수 없다. 이런 문제점의 해결 방안으로는, 정규화할 때 입력 패턴의 화소의 수를 증가시키는 방법이 있다.

3. 평균(Averaging)

각 화소들의 이진값들을 1과 -1로 바꾸어 한 집단에 속해있는 패턴들을 각 화소의 위치별로 평균을 취하여 그 집단의 중심값을 구하고 이를 대표값으로 하여 학습시 입력값으로 한다.

$$\bar{x}_i = \frac{1}{m} \sum_{k=1}^m x_i^k, \quad i=1,2,\dots,N \quad x_i^k = -1 \text{ 또는 } 1 \quad (1)$$

단, m은 한 집단내의 학습 패턴들의 수이고, N은 한 패턴내의 화소 수이고, x_i^k 는 k번째 학습 패턴의 i번째 화소값이다. 따라서 학습시의 입력값들의 수는 전체 집단들의 수와 같아진다. 이 방법으로 화소별 평균값들로 한 집단을 대표하게 하여 학습시키는 패턴들의 수를 집단의 수와 같게 줄임으로써 학습을 보다 효율적으로 수행할 수 있다.

III. 제안하는 신경망 구조

인쇄체 문자 인식을 위하여 제안하는 구조는 그림 4와 같이 단층 신경망과 간단한 논리회로(combination circuit)로 구성된다. 이 구조는 단층 신경망과 RAM을 결합한 HyLCAM(Hybrid location-content addressable memory) 구조^[4,5]를 변형한 것이다.

입력 패턴의 화소값은 -1 또는 1이고, 단층 신경망의 가중치들은 학습에 의해 결정이 된다. 단층 신경망을 학습할 때는 일반적인 시그모이드 함수 대신에 식 (2)와 같은 시그모이드 함수를 사용하고 식 (3),(4)와

같이 가중치를 갱신하는 'Generalized Delta Rule'^[6] 을 이용하였다.

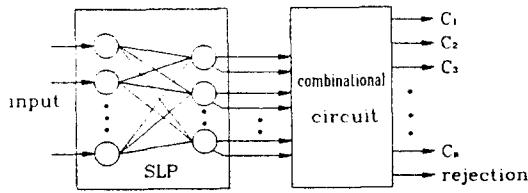


그림 4. 제안하는 신경망 구조

Fig. 4. Architecture of the proposed neural network.

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

$$\delta_j = [y_j - f(net_j)] f'(net_j) \quad (3)$$

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j \bar{x}_i \quad (4)$$

식 (2)에서 목표값은 1 또는 -1이나 코드화시 편의상 1과 0으로 변환하여 설명한다. 식 (3)에서 y_j 는 j 번째 출력 노드의 목표값이고, 식 (4)에서 \bar{x}_i 는 i번째 화소들의 평균값이다. 그리고 단층 신경망은 각 출력 노드에 연결된 가중치들이 다른 출력 노드에 연결된 가중치들과는 독립적으로 갱신되므로, 각 출력 노드별로 학습이 가능하다.

신경망의 출력층의 각 노드는 입력 화소값과 가중치의 곱들을 합하여 네트값을 계산한 후 그림 5에서 설명하는 것과 같이 한 노드에서 두개의 값-코드(code) 비트와 '기각(reject)' 비트-을 출력한다. '코드' 비트는 네트값이 0 보다 크면 '1'을 출력하고, 0 보다 작으면 '0'을 출력한다. 그리고, 네트값이 $-\Theta \sim +\Theta$ 의 영역에 들어오면 '기각' 비트는 '1'이 되고, 그렇지 않으면 '0'이 된다.

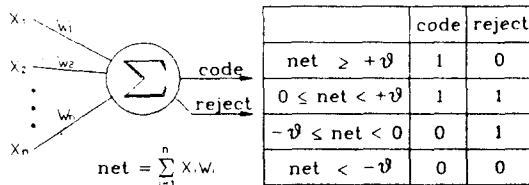


그림 5. 기각 기능이 있는 기본 소자

Fig. 5. A basic element with rejection.

IV. 선형 분리가 가능한 코딩(Linearly Separable Coding)

일반적으로 은닉층이 없고 입력층과 출력층만으로

이루어진 단층 신경망은 학습시 수렴하기 위해서 입력 값에 대한 목표값의 선정이 선형 분리가 가능(linear separability)하여야 한다는 조건이 있다.^[7] 여러가지 선형 분리 가능한 코딩 방법이 제안된바 있으며^[4,5] 선형 분리가 가능한 방법은 매우 많고,^[8] 각각의 방법은 서로 다른 인식 능력을 보인다. 단층 신경망 출력단의 코드간의 거리가 해밍거리상으로 가까운 패턴들에 대하여 어떠한 영향을 주는지가 연구 보고되었고,^[3] '0'과 '1'의 갯수가 비슷한 코드가 좋은 성능을 가지고, 비슷한 집단에는 비슷한 코드를 할당하는 것이 좋은 인식능력을 보인다는 연구 결과도 있다.^[2] 이러한 점들을 고려해서 본 논문에서는 다음과 같은 새로운 선형 분리가 가능한 코딩 방법을 제안한다.

3개 이상의 집단을 포함하는 전체 집단의 집합을 S라고 하고, S집합에 대한 초기 코드 Cs는 공집합이라고 하자.

(i) S안에 집단이 한 개만 있으면, Cs의 코드를 그 집단에 할당하고 코딩을 마친다. 그렇지 않은 경우는 (ii)로 간다.

(ii) 해밍거리상으로 서로 가장 멀리 떨어진 두개의 집단을 S안에서 선택하고, 선택된 집단들은 S에서 제외시킨다.

(iii) S에 더이상 남아있는 집단이 없으면 Cs에 각각 '0'과 '1'을 덧붙여 두개의 코드를 만들고, 이들을 선택된 두 집단에 하나씩 할당하고 코딩을 끝낸다. 그렇지 않으면 (iv)로 간다.

(iv) Cs에 각각 '00'과 '11'을 덧붙여 두개의 코드를 만들고, 이들을 선택된 두 집단에 하나씩 할당한다. Cs에는 '10'을 덧붙여 갱신한다. 이때 이전에 '00'을 할당받은 집단과 거리가 가까운 쪽에 '00'을 할당하고, '11'도 같은 방법으로 할당한다. (i)로 간다.

이 코딩이 끝나면 이미 코드를 할당받은 집단의 코드에는 마지막에 할당된 코드의 차원과 같아지도록 모든 코드에 최종 코드와 같은 '0' 또는 '1'을 반복하고 이를 '코드 1'이라고 하자. 코드 1은 논문^[4]에서 제안된 코딩 방법을 개선한 것으로, 인식율을 높이기 위하여 해밍거리가 가까운 패턴에는 코드도 해밍거리가 가까운 것을 할당한 것이다. 코드 1에서 차원을 같게 하기 위하여 반복한 '0', '1' 대신 don't care들을 덧붙인 것을 '코드 2'라고 하자. 코드 2는 '코딩에 의하여 기각이 되지 않아도 될 패턴들이 기각이 되는 것'을 완화시키기 위하여 don't care를 삽입한 것이다. 즉, 서로 멀리 있는(구별이 쉬운) 패턴들을 구별하기 위해서는 don't care가 많은 코드가 할당되고, 매우 가까이 있는(구별이 어려운) 패턴들을 구별하기 위해서는 don't care가 적은 코드가 사용된다. 그림 6은 위의 방법을

사용하여 5개의 집단을 분할하고 코드 2를 할당한 예이고, 표 1은 숫자의 인식을 위해서 사용한 코드 1과 코드 2이다.

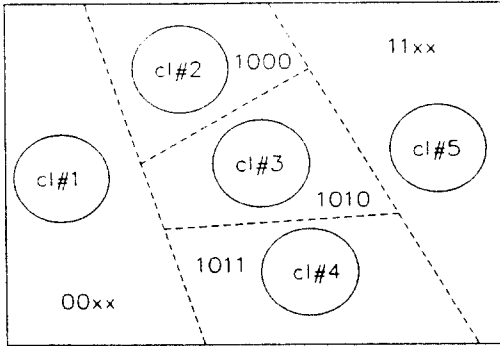


그림 6. 5개 집단에 대한 코드2 할당의 예
Fig. 6. An example of code 2 assignment for five clusters.

표 1. 숫자에 대한 코드('x' 는 don't care)
Table 1. The code for digits ('x' is a don't care).

집단#	코드 1	코드 2	숫자
1	000000000	00xxxxxxx	2
2	100000000	1000xxxxx	1
3	101000000	101000xxx	5
4	101010000	10101000x	3
5	101010100	101010100	8
6	101010101	101010101	6
7	101010111	10101011x	0
8	101011111	101011xxx	9
9	101111111	1011xxxxx	7
10	111111111	11xxxxxxx	4

V. 기각 방법

제안하는 기각 방법으로는 두 가지가 사용된다. 그 한가지는 코드에 의한 기각이고, 다른 하나는 기각 영역에 의한 기각이다. 실행시 출력층의 값들은, 제대로 인식이 되었을 경우 학습시 미리 할당된 코드가 출력되므로 이 코드가 논리회로를 통하여 한 문자를 가리키게 된다. 그러나 입력에 잡음이 많은 문자나 학습되지 않은 패턴이 들어가면 미리 할당된 코드가 아닌 다른 값이 출력될 수 있다. 이때 신경망의 출력층에서 '기각' 비트가 모두 '0'이라 하더라도, 이 코드는 논리회로를 지나면 기각을 가리키게 된다. 즉, 미리 할당된 코드가 출력되지 않으면 이 코드에 의해 구현된 논리회로에 의해 기각이 되는 것이다.

기각 영역에 의한 기각 방법은 출력층의 네트값이

$-\Theta \sim +\Theta$ 의 영역에 들어오면 오인식의 위험이 있으므로 이는 기각을 하는 것이다. 만약 한 패턴내의 i 번째 화소가 '1'일 확률이 p_i 라면, 그 화소의 평균(\bar{x}_i)은 $2p_i - 1$ 이 된다. 한 집단의 대표값으로 학습된 네트값(net value)과 그 집단에 속하는 테스트 패턴에 대한 네트값은 각각 식 (5), (6)과 같이 표현된다.

$$net_i^o = \sum_{i=1}^N \bar{x}_i w_{ji} \tag{5}$$

$$net_j = \sum_{i=1}^N x_i w_{ji} \tag{6}$$

테스트 패턴에 대한 네트값의 평균과 분산은 식 (7), (8)로 주어지고 이 증명은 부록 A에 기술하였다.

$$E[net_j] = net_j^o \tag{7}$$

$$Var[net_j] = \frac{w_{jmax}^2}{3} \sum_{i=1}^N (1 - \bar{x}_i^2) \tag{8}$$

식 (6)에서 net_j 는 동일한 확률 분포를 갖는 N 개의 독립 확률 변수(i.i.d.)로써, N 이 크다면 central limit theorem에 의해 식 (7), (8)과 같은 평균과 분산을 갖는 가우시안 확률 변수로 근사할 수 있다. 학습 후의 네트값의 평균은 식 (7)에서 예측할 수 있는데, 단층 신경망의 각 출력 노드의 최대 에러가 ϵ_{max} 보다 작아질 때까지 학습을 시키면 네트값의 평균은 목표값이 1로 학습이 된 경우,

$$1 - \frac{1 - e^{-\frac{net_j^o}{net_j^o}}}{1 + e^{-\frac{net_j^o}{net_j^o}}} < \epsilon_{max}, \text{ 또는}$$

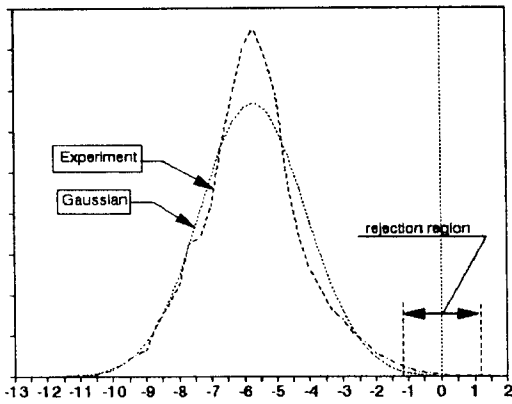
$$\ln\left(\frac{2 - \epsilon_{max}}{\epsilon_{max}}\right) < net_j^o < net_{max} \tag{9}$$

이며, 목표값이 -1인 경우도 같은 방법에 의하여 식 (10)과 같이 표현된다.

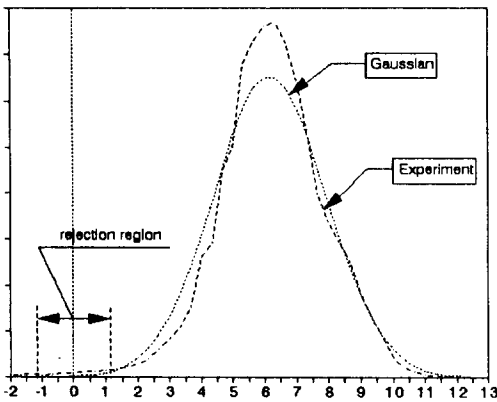
$$net_{min} < net_j^o < -\ln\left(\frac{2 - \epsilon_{max}}{\epsilon_{max}}\right) \tag{10}$$

식 (9), (10)에서 net_{max} 와 net_{min} 은 학습 알고리즘의 한계에 기인한 것으로써, ϵ_{max} 는 일반적으로 완전히 '0'이 될 수 없으므로 네트값의 평균은 유한한 값을 가진다. 위 논리의 타당성을 보이기 위해 그림 7에 한 출력 노드의 네트값 분포를 나타내었다. 각 출력 노드의 최대 에러 ϵ_{max} 를 0.01로 하면, '1'로 학습되었을 경우는 식 (9)에 의해 5.29보다 큰 값을 가지고, '-1'로 학습되었을 경우는 식 (10)에 의해 -5.29보다 작은 값을 가져야 한다. 그림에서와 같이, 각각의 평균은 6.42와 -6.60으로 예상과 일치하고, 모양은 가우

시안 분포와 비슷하다.



(a)



(b)

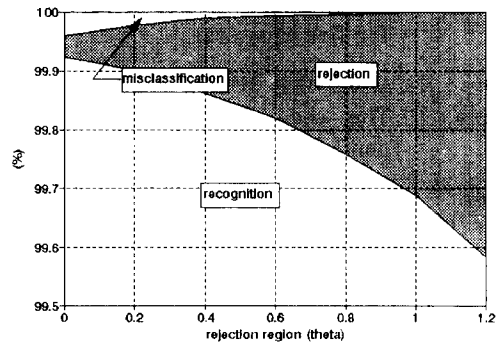
그림 7. (a) '-1'로 학습이 된 패턴들의 네트값의 분포. (b) '1'로 학습이 된 패턴들의 네트값의 분포.

Fig. 7. The Distribution of the net values of output nodes trained to be -1 (a) and trained to be 1 (b).

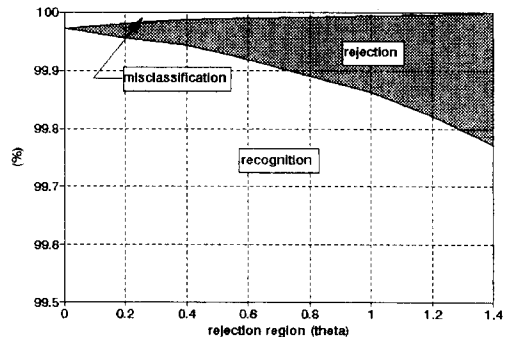
VI. 실험 결과의 분석 및 해석

앞에서 언급한 숫자와 영문자의 데이터 베이스에 대하여, 제안한 구조의 성능을 평가하기 위한 실험을 실시하여 그림 8과 9의 결과를 얻었다. 각 실험에서 특별한 언급이 없는 경우는 숫자 데이터 베이스와 '코드 2'를 사용하여 실험한 것이다. 각 노드별로 최대에러가 0.01 이하가 되도록 학습을 시켰다. 실험에 사용한 컴퓨터는 SPARC 10 이며, 학습에 걸린 시간은 숫자의 경우 약 2시간, 영문자의 경우 약 5시간이 소요되었다. 테스트에 걸린 시간은, 130,000개의 숫자는 5분, 112,000개의 영문자는 8분이 소요되었다.

숫자의 경우 코드 2를 사용하여 실험을 한 결과를 보면, 기각 영역이 없을 때($\theta = 0$)는 인식이 99.971%, 오인식이 0.029%이지만, θ 를 점점 크게 할수록 오인식은 점차 줄어들어 결국 $\theta = 1.4$ 에서 오인식이 없어졌다. 이때의 인식율은 99.771%로서 약간의 감소를 보였다. 코드 1을 사용하였을 경우는 코드 2를 사용하였을 때 보다 다소 인식율은 떨어지고 오인식율은 높았다. 영문자의 경우도 마찬가지로 인식율의 큰 감소 없이 오인식이 '0'이 될 수 있다. 실험 결과 적당한 θ 에 대하여 오인식을 크게 줄일 수 있었다. 그리고 θ 를 크게 하면 오인식이 더 줄고 어느 이상의 값이 되면 오인식은 완전히 없어진다. 그림 10은 제안하는 SLP에서, 오인식이 '0'이 되기전의 θ 값에서 오인식이 된 숫자들의 예이다. 오인식이 된 문자들의 상당수는 사람이 구별하기에도 어려운 것들이 많았다.



(a)



(b)

그림 8. 숫자 데이터 베이스에 대한 인식율, 기각율, 오인식율

(a) 코드 1 이용, (b) 코드 2 이용

Fig. 8. Recognition, rejection and misclassification ratios in the digit database.

(a) using code 1 (b) using code 2.

제 V장에서 설명하였듯이 각 출력 노드의 네트값의 분포가 가우시안이라고 가정하면, 기각 영역의 확률은

에러함수(complementary error function)를 이용하여 근사적으로 구할 수 있다. 각 출력 노드의 네트값이 $-\theta$ 에서 $+\theta$ 의 영역에 들어오는 확률은 다음과 같다.

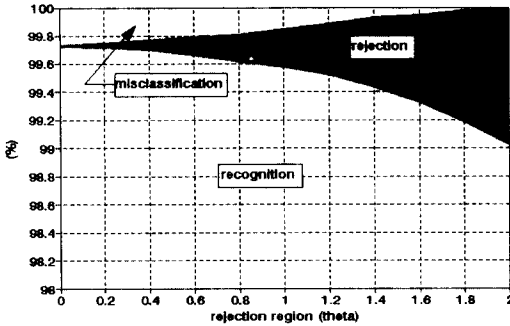


그림 9. 영문자 데이터 베이스에 대한 인식율, 기각율, 오인식율
Fig. 9. Recognition, rejection and misclassification ratios in the alphabet database.

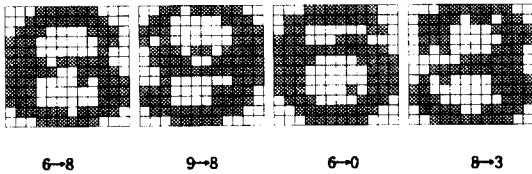


그림 10. 오인식이 된 숫자들의 예
Fig. 10. Examples of misclassified digits.

$$P_{reject}^j(1) = p(-\theta < net < \theta) \approx \text{erfc}(x_1) - \text{erfc}(x_2)$$

$$= \frac{1}{\sqrt{2\pi}} \left[\frac{e^{-\frac{x_1^2}{2}}}{x_1} - \frac{e^{-\frac{x_2^2}{2}}}{x_2} \right] \quad (11)$$

단, $x_1 = \frac{-\theta + \mu_1}{\sigma_1}$, $x_2 = \frac{\theta + \mu_1}{\sigma_1}$

식 (11)은 j번째 출력 노드의 '1'로 학습이 된 경우의 기각 확률이고, '-1'로 학습이 되었을 경우의 확률 $P_{reject}^j(-1)$ 도 μ 의 부호를 바꾸고 같은 방법으로 구할 수 있다. 식 (11)에서 알 수 있듯이, θ 나 σ 가 커질수록 P_{reject}^j 도 증가하나, μ 의 절대값이 증가하면 P_{reject}^j 는 감소한다. 그런데 여기서 각 출력 노드는 '-1' 또는 '1'로 학습이 되는 경우의 수가 다르므로, j번째 출력 노드가 '-1'과 '1'로 학습이 되는 확률을 $P_j(-1)$ 과 $P_j(1)$ 이라고 할 때 전체 출력 노드의 기각 확률은 다음과 같다.

$$P_{reject}(-\theta < net < \theta) = \sum_{j=1}^M [P_{reject}^j(1) P_j(1) + P_{reject}^j(-1) P_j(-1)] \quad (12)$$

여기서 M은 출력 노드의 수이고, 식 (12)에서 예측

한 기각 확률과 실제의 테스트 결과를 비교하면(그림 11), 예측과 실제는 다소의 차이가 나지만 경향성은 매우 근사함을 알 수 있다. 이 차이는 출력 노드의 네트값의 분포를 가우시안으로 가정한 데서 발생하는 오차 때문이다.

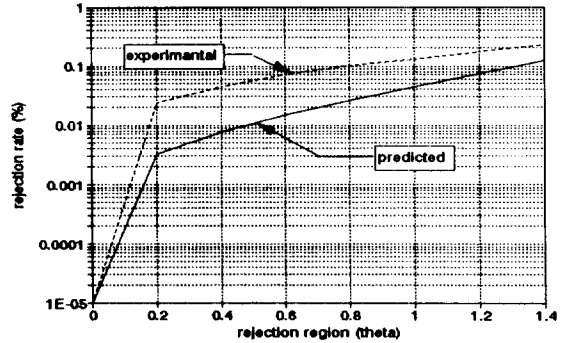


그림 11. 기각 확률의 예측치와 실험 결과
Fig. 11. Predicted and experimental values in the rejection probability.

그림 7의 종모양의 꼬리 부분은 기각 영역을 지나도 확률값을 가지는데, 이 부분은 기각 영역에서 기각이 안되고 오인식이 되는 부분이다. 이 오인식 확률은 식 (13), (14)에서 구할 수 있다.

$$P_{mis}^j(1) = p(net < -\theta) \approx \text{erfc}\left(\frac{\theta + \mu_1}{\sigma_1}\right), \mu_1 > 0$$

$$P_{mis}^j(-1) = p(net > \theta) \approx \text{erfc}\left(\frac{\theta - \mu_{-1}}{\sigma_{-1}}\right), \mu_{-1} < 0 \quad (13)$$

$$P_{mis} = \sum_{j=1}^M [P_{mis}^j(1) P_j(1) + P_{mis}^j(-1) P_j(-1)] \quad (14)$$

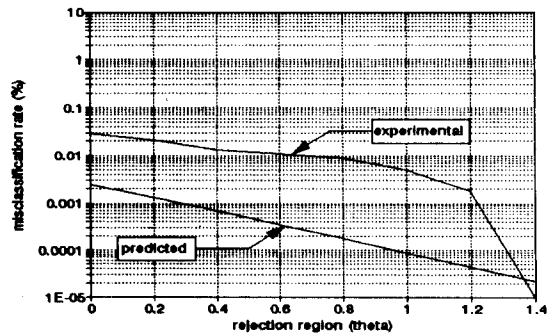


그림 12. 오인식 확률의 예측치와 실험 결과
Fig. 12. Predicted and experimental values in the misclassification probability.

식 (14)에서 예측한 오인식 확률과 실제의 테스트

결과를 비교하면(그림 12), 예측한 확률과 실제 오인식율과는 비슷한 경향을 나타내지만, 약간의 차이가 난다. 이는 출력 노드의 네트값의 분포에서 종의 꼬리부분이 실제 가우시안 분포보다 다소 큰 확률을 가지기 때문이다.

오차 역전파 알고리즘으로 학습시킨 다층 신경망과 비교하기 위하여, 같은 데이터 베이스로 인식율 테스트를 하였다. 한 개의 은닉층을 두었으며, 그 은닉층에는 숫자의 경우 20개의 노드를, 영문자의 경우 50개의 노드를 두었다. 목표값으로는 각 학습 패턴에 대하여 한 개의 출력 노드만 '1'을 출력하고, 다른 노드는 '0'을 출력하도록 하였다. 각 노드별 최대 에러는 단층 신경망의 학습시와 같게 하였다. 학습에 걸린 시간은 숫자의 경우 약 9시간, 영문자의 경우 약 32시간으로 제안한 신경망의 학습 시간에 비하여 약 5 ~ 6배가 소요되었다. 테스트에 걸린 시간은, 숫자는 9분, 영문자는 16분으로 제안한 신경망의 테스트 시간에 비하여 약 2배가 소요되었다. 인식율 테스트 결과는 표 2와 같으며, 다층 BPN은 제안한 SLP에 비하여 인식율이 떨어지고, 같은 오인식율 대비 인식율도 제안하는 SLP가 좋은 성능을 보임을 알 수 있다. 기각 영역을 BPN에도 적용을 하면 오인식율을 '0'으로 낮출 수는 있다. 즉, 출력층에 마찬가지로 기각 영역을 설정하여 한 출력 노드라도 네트값이 이 영역에 들어오면 기각을 하도록 하는 것이다. 그러나, 다층 BPN의 경우는 기각 영역을 넓힘에 따라 인식율이 크게 저하되기 때문에 이 기각 방법이 적합하지 않다.

표 2. 숫자와 영문자 데이터 베이스에 대하여 다층 신경망을 이용한 실험 결과

Table 2. Simulation results on the digit and alphabet databases using multilayer perceptrons.

	인식(%)	기각(%)	오인식(%)
숫자	98.479	1.468	0.053
영문자	96.232	3.719	0.049

같은 패턴들을 더 많은 화소들로 정규화 시키면 이 패턴은 해밍 거리 상으로 충분히 분리되므로 인식율은 좋아진다. 이를 검토하기 위해 앞에서 사용한 12x12개의 화소들로 표현된 문자들 외에, 스캐닝된 같은 문자들을 각각 8x8, 10x10, 16x16의 크기로 정규화 하여 인식율을 조사하였다(그림 13).

하드웨어 구현시에는 학습된 가중치들이 ROM에 저장되는데, 이때 ROM 크기의 한계가 있으므로 제한된 비트(bit)안에 수를 저장하기위해 양자화(quantization)를 하여야 한다. 이 경우 수 표현의 정확도가 떨

어지므로 인식율에도 변화가 온다. 학습된 가중치가 $-w_{max} \sim +w_{max}$ 사이에 분포하고, 양자화 단계가 s 일때 각 가중치가 균일한 확률 분포를 갖는 독립 확률 변수(i.i.d.)라고 가정하여 양자화를 한 후 각 출력 노드의 네트값의 분포의 평균과 분산을 구하면 아래와 같고, 이 증명은 부록 B에 기술하였다.

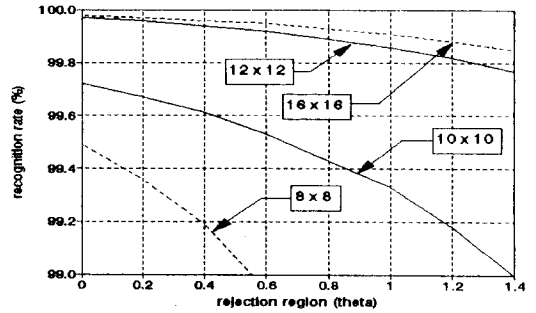


그림 13. 4가지의 패턴 크기(화소수)에 따른 인식율

Fig. 13. Recognition rates for 4 different pattern sizes.

$$E[net_i^{quant}] = E[net_i] = net_i^0 \quad (15)$$

$$Var[net_i^{quant}] = (1 + \frac{1}{s^2})Var[net_i] \quad (16)$$

단, 여기서의 s는 양자화 단계의 수이다.

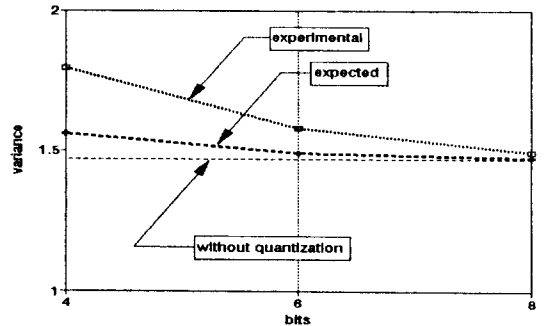


그림 14. 양자화를 하기 전후의 네트값의 분산
Fig. 14. Variances of the net values with and without quantization.

식(16)을 식(8)과 비교해보면, 양자화 단계가 작아질수록 분산은 더 커진다. 이를 검토하기 위해 4비트, 6비트, 8비트의 세 가지로 양자화를 하여 네트값의 분산의 변화(그림 14)와 인식 능력(그림 15)을 테스트하여 보았다. 그 결과, 예측한대로 분산은 비트수가 작아질수록 원래의 분산보다 더 커짐을 알 수 있다. 그리고 인식 능력에 있어서도 비트수가 커질수록 원래의 인식

능력에 가까워짐을 알 수 있고, 6비트 이상에서는 인식율이 크게 떨어지지 않음을 알 수 있다.

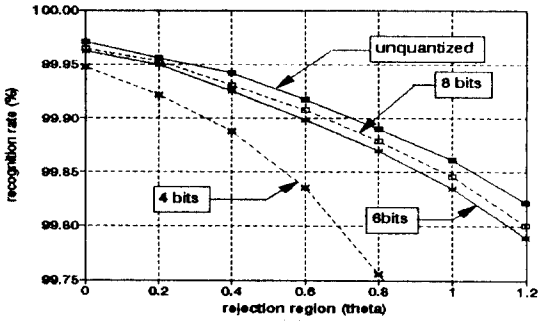


그림 15. 양자화 정확도(비트수)를 4가지로 다르게 하였을 때의 인식율

Fig. 15. Recognition rates for 4 different precisions.

제안하는 구조와 BPN이 '학습하지 않은 패턴'들에 대하여 어떤 결과를 보이는가를 실험하였다. 실험 방법은 숫자를 인식하도록 학습된 네트워크에 영문자를 인가하여 인식률을 조사하는 것이다(표 3). 영문자에서 폰트로 보았을 때 숫자와 구분이 어려운 I, O, Q는 제외를 하였고, 코드 1과 2의 θ 는 각각 앞의 실험에서 오인식이 '0'이 되는 값을 사용하였다.

표 3. 학습하지 않은 패턴들에 대한 기각율, 오인식율

Table 3. Rejection and misclassification ratios for the patterns not in the training set.

	제안하는 구조		BPN
	코드 1 ($\theta=1.2$)	코드 2 ($\theta=1.4$)	
기각(%)	91.31	80.52	95.07
오인식(%)	8.67	19.48	4.93

표 3의 결과에서 알 수 있듯이 학습하지 않은 패턴들에 대한 기각 여부 실험에서는 BPN이 가장 좋은 기각율을 보였고 don't care를 사용한 코드가 가장 낮은 기각율을 나타내었다. 이 실험에서, BPN은 대부분의 영문자에 대하여 기각을 하였지만, 제안한 신경망은 '학습하지 않았지만 비교적 가깝다고 판단되는 패턴'이 학습 패턴에 있으면 그 패턴으로 인식을 하였다. 즉, 상당히 많은 'B'가 '6'이나 '8'로 인식되었으며, 'T'는 '7'로, 'Z'는 '2'로 인식이 되었다. 이 결과를 분석하여 보면, 제안한 코드는 정해진 학습 패턴들에 대하여 인식율을 극대화하기 위하여 해밍거리와 don't care를 사용하였으므로, 학습하지 않은 패턴들에 대하여는 이 이

유로 인하여 가장 가까운 패턴을 찾아가기 때문에 이러한 결과가 나온 것이다. 따라서, 이러한 위험을 배제하기 위하여는 가능한 잡음이 있는 많은 패턴들에 대하여 학습을 시키거나, 또는 이러한 가능성(학습하지 않은 패턴들이 인가되는)을 제외시킬 수 있는 전처리 과정이 필요하다.

VII. 결 론

제안한 SLP는 패턴 인식의 응용 분야에서 큰 문제점인 오인식 문제의 해결 방법을 제시하였다. 높은 인식율을 유지하면서 매우 낮은 오인식을 보장하기 위해 기각 방법을 도입하여 작은 기각 영역에 의해 오인식을 '0'으로 떨어뜨리고 인식율은 큰 영향을 받지 않음을 실험 결과에서 확인하였다. 이 결과는 다층 신경망과 비교하여도 좋은 성능을 보였다. 그리고 이 기각 영역에 의한 기각 방법은 해석과 분석을 토대로 그 방법의 타당성을 증명하였다. 학습 패턴들은 해밍 거리가 가까운 것들끼리 집단으로 분류하여, 화소별로 평균을 취한 값들로 각 집단들을 대표하게 하여 학습시 입력값으로 사용하였기 때문에 학습 패턴의 수가 줄어들어 학습을 매우 용이하게 하였다. 단층 신경망은 학습시 수렴하기 위해 입력값과 목표값이 선형 분리가 가능해야 하기 때문에, 각 집단에 선형 분리가 가능한 코드를 부여하는 방법을 제안하였다. 그리고 제안한 방법은 각 집단의 패턴들이 다른 집단의 패턴들과 겹치지 않고 충분히 분리되어 있다면 오인식 없이 완벽히 문자들을 구분할 수 있다. 제안한 구조는 다층 신경망과 간단한 논리회로의 연결로 구성되어 있기 때문에 하드웨어 구현 면에서도 간단하다. 또한 가중치 양자화 문제에 대한 분석과 실험에 의하면 6비트 이상의 가중치를 사용하면 인식율이 크게 저하되지 않았다. 결과적으로, 널리 쓰이는 다층 신경망에 비하여, 학습 속도가 매우 빠르고, 선형 분리가 가능한 코드로 인하여 항상 수렴하며, 높은 인식율을 유지하면서 오인식율은 매우 낮으며, 하드웨어 구현도 간단함을 확인하였다. 현재 제안된 구조로 하드웨어를 구현하는 작업이 진행중이며, 향후 과제로는 코딩 방법과 인식 능력의 관계, 학습하지 않은 패턴에 대한 면역성, 그리고 기각 영역의 경계값과 학습시 허용오차의 관계에 대한 해석적인 연구 등이 있다.

부록 A

식(5), (6)에서,

$$net_j = \sum_{i=1}^N \bar{x}_i w_{ji} + \sum_{i=1}^N (x_i - \bar{x}_i) w_{ji} \quad (A.1)$$

이고, 이에 기대값을 취하면,

$$E[net_j] = E[net_j^o] + E[\sum_{i=1}^N (x_i - \bar{x}_i) w_{ji}] \quad (A.2)$$

이 된다. 위 식에서 $(x_i - \bar{x}_i)$ 와 w_{ji} 는 독립이고, $E[x_i - \bar{x}_i]$ 는 '0'이므로

$$E[net_j] = net_j^o \quad (A.3)$$

그리고,

$$\begin{aligned} Var[net_j] &= E[(net_j - E[net_j])^2] \\ &= E[(\sum_{i=1}^N (x_i - \bar{x}_i) w_{ji})^2] \\ &= E[\sum_{i=1}^N (x_i - \bar{x}_i)^2 w_{ji}^2] \\ &\quad + 2E[\sum_{i \neq k} (x_i - \bar{x}_i)(x_k - \bar{x}_k) w_{ji} w_{jk}] \end{aligned} \quad (A.4)$$

이다. $(x_i - \bar{x}_i)$ 와 $(x_k - \bar{x}_k)$, $i \neq k$, 는 독립이고, $E[x_i - \bar{x}_i]$ 는 '0'이며, $(x_i - \bar{x}_i)$ 와 w_{ji} 가 독립이므로 각 집단의 중심값(\bar{x}_i)에 대한 입력 벡터의 분산을 σ_i^2 이라고 하면,

$$Var[net_j] = \sum_{i=1}^N \sigma_i^2 E[w_{ji}^2] \quad (A.5)$$

σ_i^2 을 유도하여 보자. 한 화소가 '1'이 될 확률 p_i 는 다음 식과 같이 나타낼 수 있다.

$$p_i = \frac{1 + \bar{x}_i}{2}, \quad 1 - p_i = \frac{1 - \bar{x}_i}{2} \quad (A.6)$$

즉, x_i 가 '1'일 확률은 p_i 이고, x_i 가 '-1'일 확률은 $1 - p_i$ 이다. 따라서 분산은,

$$\sigma_i^2 = E[(x_i - \bar{x}_i)^2] = (1 - \bar{x}_i)^2 p_i + (-1 - \bar{x}_i)^2 (1 - p_i) \quad (A.7)$$

이 된다. 학습된 가중치가 $-w_{max} \sim +w_{max}$ 사이에 분포하고, 균일한 확률 분포를 갖는 독립 확률 변수(i.i.d.)라 가정하면, $E[w_{ji}^2]$ 은 $w_{ji}^2/3$ 이 된다. 따라서, 다음 식을 유도할 수 있다.

$$Var[net_j] = \frac{w_{max}^2}{3} \sum_{i=1}^N (1 - \bar{x}_i^2) \quad (A.8)$$

부록 B

식 (15), (16)을 증명하기 위하여, 각 양자화 에러를

Δw_{ji} 라고 하면,

$$E[\Delta w_{ji}] = 0$$

$$Var[\Delta(w_{ji})^2] = E[\Delta(w_{ji})^2] = \frac{1}{3} \left(\frac{w_{max}}{s} \right)^2 \quad (B.1)$$

이 되고 가중치는 $(w_{ji} + \Delta w_{ji})$ 이 되므로, 양자화된 후 네트값의 평균은 다음과 같이 구할 수 있다.

$$net_j^{quant} = \sum_{i=1}^N x_i (w_{ji} + \Delta w_{ji}) = \sum_{i=1}^N x_i w_{ji} + \sum_{i=1}^N x_i \Delta w_{ji} \quad (B.2)$$

$$\begin{aligned} E[net_j^{quant}] &= E[\sum_{i=1}^N x_i w_{ji} + \sum_{i=1}^N x_i \Delta w_{ji}] \\ &= net_j^o + \sum_{i=1}^N E[x_i] E[\Delta w_{ji}] = E[net_j] \end{aligned} \quad (B.3)$$

분산은 식(A.4)의 유도방법을 이용하여 다음과 같이 나타낼 수 있다.

$$\begin{aligned} Var[net_j^{quant}] &= E[(net_j^{quant} - E[net_j^{quant}])^2] \\ &= \sum_{i=1}^N E[(x_i - \bar{x}_i)^2] (E[w_{ji}^2] + E[(\Delta w_{ji})^2]) \\ &= (1 + \frac{1}{s^2}) \frac{w_{max}^2}{3} \sum_{i=1}^N [1 - \bar{x}_i^2] \\ &= (1 + \frac{1}{s^2}) Var[net_j] \end{aligned} \quad (B.4)$$

참고 문헌

- [1] S. Knerr, L. Personnaz, and G. Dreyfus, "Handwritten digit recognition by neural networks with single-layer training." *Neural Networks*, vol.3, no. 6, pp. 962-968, 1992.
- [2] J. Kamruzzaman, Y. Kumagai and H. Hikita, "On cascaded artificial neural network architecture," in Proc. *IEEE/INNS Int. Joint Conf. on Neural Networks, IJCNN '92*, (Beijing, CHINA), vol. 1, pp. 210-217, Nov., 1992.
- [3] J. Kamruzzaman, Y. Kumagai and H. Hikita, "Robust performance using cascaded artificial neural network architecture." *IEICE Trans. Fundamentals*, vol. E76-A, no. 6, pp. 1023-1030, June, 1993.
- [4] B. H. Wang, "Hybrid location-content addressable memory", *Proc. IEEE Int. Conf. Neural Networks (ICNN)* at San Francisco, Ca., vol. 1, pp. 347-352, Mar., 1993.

- [5] J. Lim, E. Lee and S.-I. Chae, "Character recognition by neural networks with single-layer training and rejection mechanism," in *Proc. IEEE Int. Symp. on Circuits and Systems, ISCAS '94*, (London), vol. 6, pp. 327-330, May, 1994.
- [6] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing*, vol. 1, MA:MIT Press, 1986.
- [7] M. Minsky and S. Papert, *Perceptrons*, Cambridge, MA:MIT Press, 1969.
- [8] Z. Kohavi, "Switching and finite automata theory", McGraw-Hill, pp. 181-190, 1970.

 저 자 소 개



林俊 (正會員)

1967年 9月 29日生. 1992年 한국과학기술원 전기및전자공학과 (공학사). 1995年 서울대학교 전자공학과(공학석사). 1995年 3月 ~ 현재 서울대학교 대학원 전자공학과 박사과정 재학중. 주

관심 분야는 신경회로망 및 영상처리용 집적회로 설계 등임.

蔡洙翊 (正會員) 第 31 卷 B編 第 5 號 參照.

서울대학교 전자공학과 및 반도체공동연구소 교수