

셀룰라 신경회로망의 연상메모리를 이용한 영상 패턴의 분류 및 인식방법

Image Pattern Classification and Recognition by Using the Associative Memory with Cellular Neural Networks

신윤철 · 박용훈 · 강 훈

Yoon-Cheol Shin, Yong-Hun Park and Hoon Kang

중앙대학교 공과대학 전자전기공학부

요 약

셀룰라 신경회로망의 연상 메모리를 이용하여 시각적인 입력 데이터의 연산을 통하여 영상 패턴의 분류와 인식을 수행한다. 셀룰라 신경회로망은 일반적인 신경회로망과 같이 비선형 데이터의 실시간 처리가 가능하고, 세포자동자와 같이 이 격자구조의 셀로 이루어져 인접한 셀과 직접 정보를 주고받는다. 응용 분야로는 최적화, 선형/비선형화, 연상 메모리, 패턴인식, 컴퓨터 비전 등에 적용할 수 있다. 영상의 이미지 픽셀을 셀룰라 신경회로망의 셀에 대응하여 전체 이미지 영상을 모든 셀룰라 신경회로망의 셀에서 동시에 병렬로 처리할 수 있어 2-D 이미지 처리에 적합하다. 본 논문은 셀룰라 신경회로망에 의한 연상 메모리 구조를 설계하고, 학습된 하중값 메모리에서 가장 적당한 하중값을 선택하여 학습된 영상과 정확히 일치하는 출력을 얻는 방법을 제시한다. 학습을 통한 연상 메모리 구현에는 각각의 뉴런에서 일정하지 않은 다른 템플릿을 사용한다. 각각의 템플릿은 뉴런들 간의 연결 하중값을 나타내고 학습에 따라 갱신된다. 학습방법으로는 템플릿 하중값 학습에 뉴런들 간의 연결 하중값을 조정하는 가장 단순한 규칙인 Hebb의 학습방법이 사용되었고 분류값 학습에 LMS 알고리즘이 사용되었다.

Abstract

In this paper, Associative Memory with Cellular Neural Networks classifies and recognizes image patterns as an operator applied to image process. CNN processes nonlinear data in real-time like neural networks, and made by cell which communicates with each other directly through its neighbor cells as the Cellular Automata does. It is applied to the optimization problem, associative memory, pattern recognition, and computer vision. Image processing with CNN is appropriate to 2-D images, because each cell which corresponds to each pixel in the image is simultaneously processed in parallel. This paper shows the method for designing the structure of associative memory based on CNN and getting output image by choosing the most appropriate weight pattern among the whole learned weight pattern memories. Each template represents weight values between cells and updates them by learning. Hebbian rule is used for learning template weights and LMS algorithm is used for classification.

Key Words : cellular neural network, associative memory, pattern recognition, local/global weight, weight selection

1. 서 론

셀룰라 신경회로망(Cellular Neural Networks : CNN)[1,2,3]은 신경회로망(Neural Network)[4]과 같이 비선형 데이터의 실시간 병렬처리가 가능하고, 세포자동자(Cellular Automata : CA)[5]와 같이 격자구조의 셀로 이루어져 인접한 셀과 직접 정보를 주고받는다[1,2]. 신경회로망보다는 구조적으로 세포자동자와 더욱 유사한 특성을 보이지만 세포자동자가 주변의 조건에 영향을 받는 반면 CNN은 주변의 영향과 출력특성에 따라 신경망적인 특성을 보인다.

응용 분야로는 최적화, 선형/비선형화, 연상 메모리, 패턴인식, 컴퓨터 비전 등을 들 수 있다[1,3]. 특히, 영상의 특징점 추출, 영상처리 및 패턴인식 분야에서 많은 연구가 진행되어 왔고, 최근에 Discrete-Time Cellular Neural Networks (DTCNN)을 이용한 연상 메모리 분야의 연구가 활발히 진행 중이다. 그 방법으로는 Hebbian rule, relaxation algorithm, eigenstructure method, pseudo-inversion technique[6,7,8,9] 등이 있다.

CNN을 이용한 영상처리는 하나의 이미지 픽셀을 하나의 셀에 대응시킴으로서 전체 이미지 영상을 모든 CNN 셀에서 동시에 병렬로 처리할 수 있어 2D 이미지 처리에 적합하다. 원도 템플릿을 사용한다는 점은 기존의 이미지 처리와 같고 기존의 모든 이미지 처리를 할 수 있다. 또한 입력 제어 템플릿과 출력 제한 템플릿 그리고 바이어스에 따라서 기존의 이미지 처리와 다른 처리를 할 수 있다[3]. CNN연산에 있어서 가장 중요한 점은 템플릿 값을 설정하는 것이다. 크게 모든 셀에 대하여 일정한 템플릿(cloning template)과 일정하지

접수일자 : 2002년 5월 15일

완료일자 : 2002년 8월 26일

본 연구는 과학기술부 "뇌신경정보학 연구 사업 (M10107020001-02B2202-00110)"의 지원에 의하여 수행되었습니다.

얇은 템플릿(non-cloning template) 등을 적용하는 두 가지의 방법이 있다. 진화연산 등의 방법[10]으로 템플릿 값을 구하는 경우도 있으나, 그 값을 설정하는 방법은 정의되어 있지 않고 대부분 시행착오에 의하여 값을 구한다.

본 논문에서는 시각적인 입력 데이터의 연산장치의 하나인 영상처리 부분에 응용하여 입력패턴의 분류와 인식을 수행한다. CNN에 의한 연상 메모리 구조를 설계하고, 학습된 하중값 메모리에서 가장 적당한 하중값을 선택하여 학습된 영상과 정확히 일치하는 출력을 얻는 방법을 제시한다.

학습을 통한 연상 메모리 구현에는 각각의 셀에서 일정하지 않은 다른 템플릿을 사용한다. 각각의 템플릿은 셀들 간의 연결 하중값을 나타내고 학습에 따라 갱신된다. 학습방법으로는 템플릿 하중값 학습에 뉴턴들 간의 연결 하중값을 조정하는 가장 단순한 규칙인 Hebb의 학습방법이 사용되었고 분류값 학습에 LMS 알고리즘이 사용되었다. Hebb의 방법은 구조와 학습방법이 간단하고 학습속도, 패턴추출에서 우수한 성능을 가지고 있다. 단점으로는 학습 저장능력에 제한이 있지만 학습률의 제한은 하중값 패턴의 선택을 통하여 해결할 수 있다. 전체의 하중값은 입력 영상(m×n)의 크기와 같은 크기의 행렬(m×n)이 9개(r=1의 이웃한 셀 수)가 중첩된 모양으로 구성되고, 각 패턴의 수만콤의 부분 하중값(local weight)이 존재하고 모든 부분 하중값의 합인 전체 하중값(global weight)이 존재한다. CNN의 연상 메모리를 인간의 뇌의 공학적인 모델링을 통한 기억?연상 메커니즘 분석을 위하여 적용하면 시각을 통한 두뇌의 작용에 적용할 수 있다.

2. Cellular Neural Networks

2.1 Cellular Neural Networks

셀룰라 신경회로망은 셀이라 불리는 연산요소들로 이루어진다. 다수의 입력과 한 개의 출력을 가지며, 그림 1과 같이 한 층의 셀들은 모두 연결되어 각 셀의 정보는 인접한 셀과 그 인접한 셀을 통하는 주변의 다른 셀에 의해 직접 갱신된다. 이러한 특성은 셀룰라 오토마타와 유사한 특성을 보이며 일반적인 신경회로망과는 다른 성질을 보인다. 2D, 3D 영상과 비디오 신호처리 분야를 비롯하여 패턴인식, 신호처리 분야 등에서 폭넓게 적용되고 있다.

그림 2에 보이는 바와 같이 출력은 입력과 셀 상태값에 의해 결정되고, 궤환 성분에 의해 다시 입력으로 반복되어 연속적인 움직임을 얻을 수 있는 네트워크 형태로 표현된다. 하나의 셀에서의 상태 값과 출력은 식 (1) 과같이 비선형 방정식으로 나타난다. 수식에 사용되는 A, B는 출력 궤환값과 입력값에 적용하는 3×3(r=1일 때, r : 셀의 이웃 반경) 또는 5×5(r=2 일 때)의 템플릿으로서, A는 이웃한 셀들의 영향, B는 입력의 영향을 나타내며, x는 셀의 상태값, y는 셀의 출력값, u는 초기의 입력값, I는 바이어스를 나타낸다.

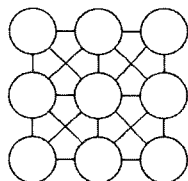


그림 1. 셀룰라 신경회로망의 연결구조
Fig. 1. Cellular Neural Networks Structure

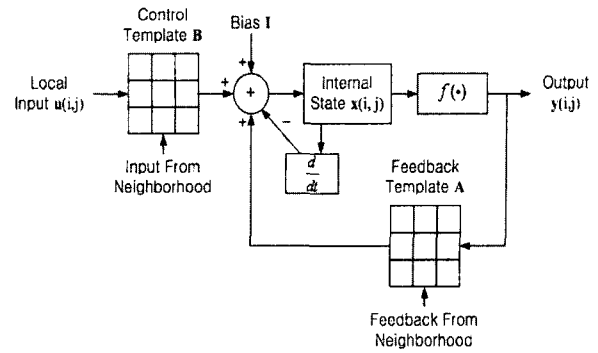


그림 2. 셀룰라 신경회로망의 연산
Fig. 2. Operation of CNN

$$C \frac{dx_{ij}(t)}{dt} = -\frac{1}{R} x_{ij}(t) + \sum_{C(k, \ell) \in N_r(i, j)} A(i, j, k, \ell) y_{k\ell}(t) + \sum_{C(k, \ell) \in N_r(i, j)} B(i, j, k, \ell) u_{k\ell}(t) + I_{ij}$$

$$y_{ij}(t) = f(x_{ij}(t)) = \frac{1}{2} (|x_{ij}(t) + 1| - |x_{ij}(t) - 1|) \quad (1)$$

A(θ), B(θ)	templates matrices	
C(i, j)	i th row, j th column cell	
N(i, j)	neighborhood	
x_{ij}	state	
y_{ij}	output	$i = 1L \ m$
u_{ij}	input	$j = 1L \ n$
I_{ij}	bias	

그림 3은 셀룰라 신경회로망의 출력특성 그래프이다. 출력은 -1과 1 사이의 값을 가지게 되고, 반복됨에 따라 -1 또는 1로 수렴하는 특성이 있다.

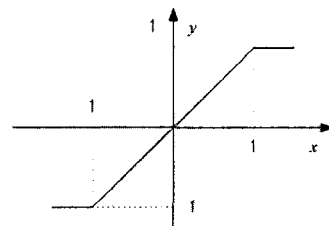
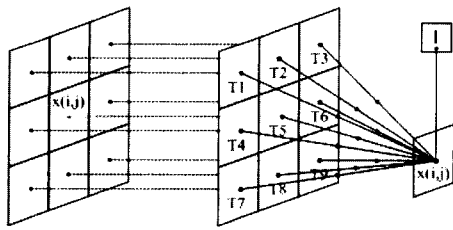


그림 3. 셀룰라 신경회로망의 출력특성
Fig. 3. Output Characteristic of CNN

2.2 Discrete-time Cellular Neural Networks

이산 시간 셀룰라 신경회로망(Discrete-Time Cellular Neural Networks : DTCNN)은 이차원 영상신호처리에 사용하기 위하여 식 (1) 을 변형하여 식 (2) 와 같이 2차원 영상의 변환식으로 표현될 수 있다. 또한 간략히 식 (3) 과 같이 나타낼 수도 있다.

그림 4는 하나의 셀 C(i,j)의 연산을 나타내고 모든 셀에 일관되게 적용된다. 그림 5는 각각 r=1, r=2 인 템플릿이다. 각각 9개, 25개의 이웃셀을 가지고 있다.



$$x_v^N = \{x(i-1, j-1), x(i, j-1), x(i+1, j-1), x(i-1, j), x(i, j), x(i+1, j), x(i-1, j+1), x(i, j+1), x(i+1, j+1)\}$$

$$T_v^N = \{t1, t2, t3, t4, t5, t6, t7, t8, t9\}$$

$$x_v(t+1) = \sum_{N=1}^9 T_v^N x_v^N(t) + I$$

그림 4. DTCNN 템플릿 연산
Fig. 4. DTCNN Template Operations

$$X_{ij}(n+1) = \sum_{C(k, l) \in N(i, j)} A(i, j, k, l) y_k(n) + \sum_{C(k, l) \in N(i, j)} B(i, j, k, l) u_k(0) + I_{ij}$$

$$y_{ij}(n+1) = f(x_{ij}(n+1))$$

$$= \frac{1}{2} (|x_{ij}(n+1) + 1| - |x_{ij}(n+1) - 1|)$$

$$x_{ij}(n+1) = \sum_{C(k, l) \in N(i, j)} T(i, j, k, l) x_k(n) + I_{ij}$$

T: Template matrix

2.3 Image Processing

DTCNN을 이용한 이미지 처리에서는 템플릿 A, B, 바이어스 I에 적당한 값을 취하면 기존의 이미지 변환과 같은 처리를 할 수 있다. 템플릿을 사용한 이미지 처리는 기존의 이미지 처리방법과 유사하지만 초기 입력의 제어 템플릿 B와 바이어스 I에 따라 새로운 이미지 처리를 할 수 있고, 출력값이 수렴하는 특성이 있다. 이미지의 각 픽셀은 DTCNN의 각 셀에 대응하고, 그레이 스케일 영상 (0 ~ 255)의 각 픽셀값은 각각의 셀에 상태값 (1 ~ -1)으로 한다. 이미지 경계에 있는 셀의 처리를 위해 m×n 영상에 대하여 r=1인 경우 (m+2)×(n+2), (m+4)×(n+4)의 DTCNN을 설정하고, 경계셀은 항상 0의 상태값을 가지도록 한다.

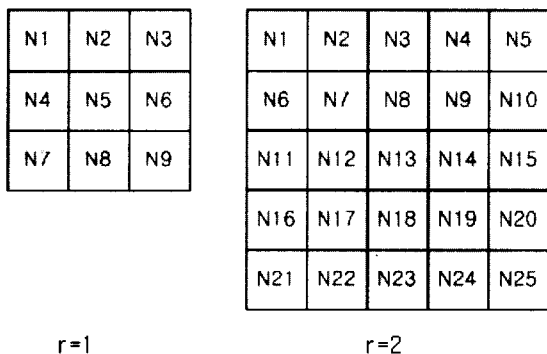


그림 5. r=1, r=2 일때의 이웃셀
Fig. 5. Neighborhood Cell with r=1, r=2

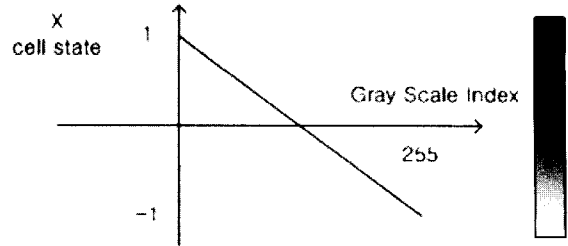


그림 6. 그레이 영상에 따른 셀 상태값
Fig. 6. Cell State vs. Gray Image Value

3. Cellular Neural Network with Associative Memory

3.1 Associative Memory

연상 메모리(Associative Memory)는 주어진 자료에 대해 정보를 저장하는 장치이며, 그림 7에 나타나 있는 Content-Addressable Memory의 일종으로 간주할 수 있다. 대표적인 연상 메모리로는 Bart Kosko의 쌍방향 연상 메모리(Bidirectional Associative Memory, BAM)와 John Hopfield의 Hopfield Memory가 있다.

본 논문의 셀룰라 신경회로망을 사용한 연상 메모리는 비교사학습(Hebbian 학습)과 교사학습(LMS 학습)의 2가지의 방법을 사용하여 학습된 패턴의 정확한 분류와 왜곡된 패턴의 완벽한 복원, 분류를 수행한다.

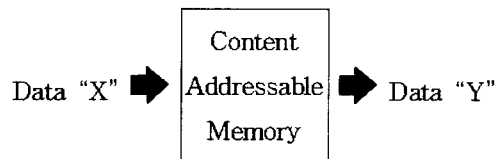


그림 7. Content-Addressable Memory
Fig. 7. Content-Addressable Memory

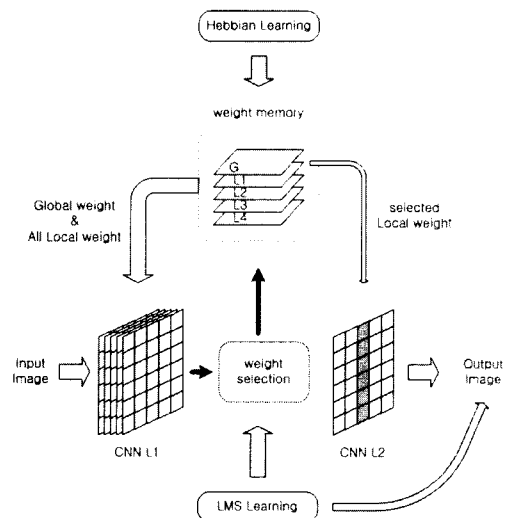


그림 8. CNN 연상 메모리
Fig. 8. AM with CNN

각 셀에 적용하는 템플릿은 일정하지 않은 non-cloning 템플릿으로 이웃한 셀 간의 상태값에 따른 학습된 하중값을 나타낸다. 각 입력 패턴은 LMS 알고리즘에 의해 먼저 분류값이 정해지고, Hebb의 학습방법에 따라 CNN의 각 셀의 템플릿 하중값을 구한다.

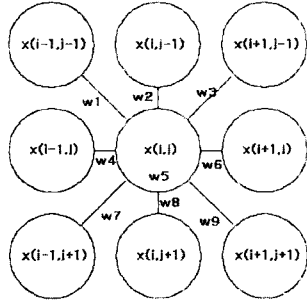


그림 9. 셀 하중값
Fig. 9. Cell Weights

3.2 Cellular Neural Networks 학습방법-Hebbian 학습

CNN 연상 메모리의 학습에 사용한 Hebb의 학습방법은 인접한 두 셀들의 상태값에 따라 두 셀 간의 연결된 하중값을 조정한다. 하중값은 모두 0으로 초기화 하고, 반경(r)에 따라 떨어진 셀의 영향은 고려하지 않고 인접한 9개(r=1) 또는 25개(r=2) 셀의 영향만 받는다. 즉, r=1 일 때 하나의 셀에 적용할 템플릿 값을 인접한 9개의 셀 간의 연결강도에 따라 하중값으로 설정한다.

$$W_{ij}^N = \sum_{k=1}^n x_{ij} x_{ij,k}^N$$

$$x_{ij}^N = \{x(i-1, j-1), x(i, j-1), x(i+1, j-1), x(i-1, j), x(i, j), x(i+1, j), x(i-1, j+1), x(i, j+1), x(i+1, j+1)\}$$

$$W_{ij}^N = \{w1, w2, w3, w4, w5, w6, w7, w8, w9\}$$

$$y_{ij} = f(s_{ij}) = f\left(\sum_{k=1}^9 W_{ij,k}^N x_{ij,k}^N\right)$$

그림 9와 식 (4), (5) 는 9개(r=1)의 인접 셀을 가진 구조에서 적용된다. 각각의 셀의 연결 하중값은 그림 9와같이 연결되고, 는 ij번째 셀과 인접한 N번째 셀, 의 k는 전체 p개의 패턴 중 k번째 패턴, 는 ij번째 셀과 인접한 N번째 셀 간의 연결 하중값을 나타낸다. 셀 출력은 식 5와 같이 인접한 9개의 셀의 상태값과 연결된 하중값의 영향을 받는다.

j \ i	1	...	m
1	(w1,w2,...w9)	...	(w1,w2,...w9)
...
n	(w1,w2,...w9)	...	(w1,w2,...w9)

그림 10. 하중값 행렬 W_{ij}^N (r=1)
Fig. 10. Weight Matrix W_{ij}^N (r=1)

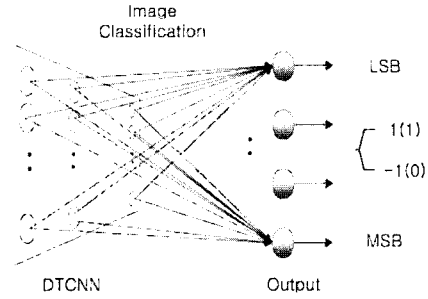


그림 11. 패턴 분류기
Fig. 11 A Pattern Classifier

전체의 하중값은 그림 10과 같은 구조의 3차원 행렬 형식으로 나타내어진다. 즉, m×n 행렬이 9개 중첩된 모양으로 전체의 하중값 메모리의 크기는 m×n×9 이다.

Hebb의 학습방법은 새로운 학습 패턴 추가 삭제시 전체의 패턴을 학습하지 않고 주어진 패턴만 추가, 삭제하여 학습 시간이 빠르고, 새로운 패턴의 학습과 제거가 쉽다.

$$\text{새로운 학습패턴 추가 : } W_{ij}^N = W_{ij}^N + W_{ij,p+1}^N$$

$$\text{학습된 패턴 삭제 : } W_{ij}^N = W_{ij}^N - W_{ij,k}^N \quad (6)$$

3.3 분류 학습 - 최소 제곱평균(LMS) 학습

출력된 이미지의 판별을 위하여 LMS 알고리즘을 사용하고, 하중값 패턴의 선택과 분류값 산출에 이용한다. 최소 제곱평균 학습은 학습쌍에 대한 제곱오차의 평균을 최소화하는 최적의 실제 하중값을 유도할 수 있도록 하중값들을 수정하는 작업이다. 4개의 출력 노드로 그레이 코드를 사용하여 16개의 패턴을 분류하여 인식할 수 있다.

$$w_{k+1} = w_k + 2\mu \epsilon_k x_k \quad (7)$$

3.4 Global Weight & Local Weight

Hebb의 학습방법을 이용한 CNN 연상 메모리는 적은 수의 학습 데이터와 학습 원본의 왜곡이 없는 입력에서는 우수한 성능을 보이나 학습 데이터 수가 증가할수록 그리고 왜곡된 입력에서는 정확한 출력을 얻을 수 없다. 본 논문에서는 CNN의 학습 방법을 2가지로 분류하여 심하게 왜곡된 입력에서도 원본 학습 데이터를 정확히 추출할 수 있다.

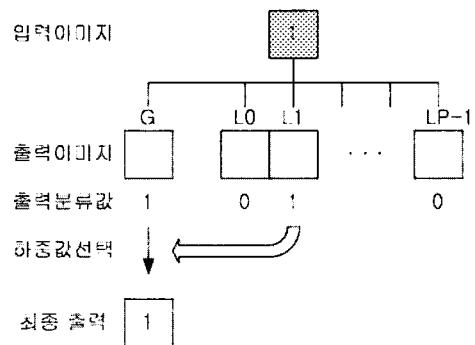


그림 12. 하중값 선택 I
Fig. 12. Weight Selection I

3.4.1 Local Weight

식 4에서 하중값은 각각 패턴의 하중값을 모두 더한 값으로 설정했다. 즉, 패턴의 수만큼의 부분 하중값(local weight)이 존재한다.

$$LW_{ij,l}^N = x_{ij,k} x_{ij,k}^N \quad k=1 \dots P \quad (8)$$

3.4.2 Global Weight

전체 하중값(global weight)은 모든 패턴에서의 하중값이다. 즉, 모든 부분 하중값의 합이다.

$$GW_{ij}^N = \sum_{k=1}^P LW_{ij,k}^N = \sum_{k=1}^P x_{ij,k} x_{ij,k}^N \quad (9)$$

3.5 Weight Selection

하중값 선택은 하중값 메모리의 부분 하중값 패턴 중에서 하나의 하중값 패턴을 선택한다. 입력 이미지는 전체 하중값과 모든 부분 하중값에 적용되어 P개의 학습데이터중 하나의 입력에 1개의 전체 하중값 출력과 P개의 부분 하중값 출력이 존재하며, 출력 분류값을 구하여 선택 알고리즘에 따라 가장 최적의 하중값 패턴을 선택하여 전체 하중값 출력에 적용한다. 선택된 하중값 패턴으로 반복하면 정확한 학습된 출력 영상을 볼 수 있다.

그림 12는 분류값이 일치하는 경우, 그림 13은 분류값이 일치하지 않은 경우의 예이다.

※ 하중값 선택 알고리즘 (그림 12 참조)

1. local pattern의 위치와 부분 하중값 출력의 분류값(L0, L1, , LP-1)이 같은 패턴들을 선택 (패턴 1이 일치)
2. 선택된 패턴들 중 전체 하중값 출력의 분류값(G)과 일치하는 패턴을 선택 (전체 분류값 1, 일치하는 1번째 패턴 선택)
3. 선택된 패턴의 부분 하중값을 전체 하중값 출력에 적용

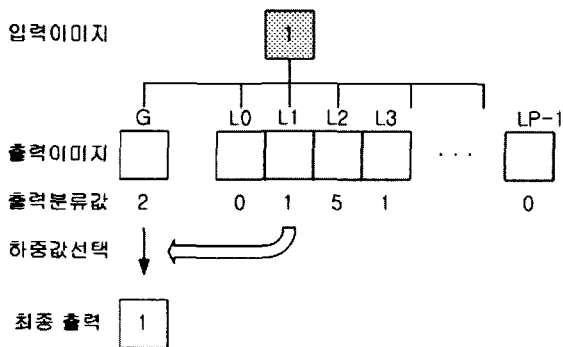


그림 13. 하중값 선택 II
Fig. 13. Weight Selection II

1번에서 일치하는 패턴이 없을 경우 일치하는 값이 있을 때까지 반복 수행한다.(그림 13 참조)

2번에서 일치하는 패턴이 없을 경우 전체 하중값 출력의 분류값과 가까운 패턴을 선택한다.(그레이 코드의 분류값에서 가까운 패턴)

4. 시뮬레이션 결과

시뮬레이션에서는 0~9의 아라비아 숫자 이미지와 16개의 교통표지판 이미지를 분류 인식하였다. 분류기에 4개의 출력노드를 두어 개의 이미지를 분류 인식할 수 있게 하였고 입력 이미지는 256의 그레이 영상을 사용한다. 출력노드의 출력값은 1 또는 -1 이고 오차 범위는 0.0001, 학습률은 0.00001이다.

사용된 LMS 알고리즘

$$\begin{aligned} \text{error} &= \text{node_output} - \text{desired_output} \\ \text{if } |\text{error}| > 0.0001 \\ \text{Weight} &= \text{Weight} - 0.00001 * \text{error} * \end{aligned}$$

4.1 숫자 영상 분류

숫자 영상들을 분류하기 위해 그림 14와 같은 흑백의 2진 이미지를 입력으로 사용하였다. 각 이미지의 크기는 50×50이고, 이미지의 글자체는 돋움체를 사용하였다. 9개의 이웃셀을 가진 CNN은 52×52 크기를 가지고, 25개의 이웃셀을 가진 CNN은 54×54 크기를 가진다. 각 픽셀은 +1(흑)과 -1(백)의 값을 가지고 경계셀의 값은 항상 0이다.

학습 원본과 같은 입력에서는 전체 하중값만으로도 정확한 분류를 수행할 수 있으나, 왜곡된 입력 이미지에서는 전체 하중값만으로는 원하는 결과를 얻을 수 없다. 2진 입력 이미지에서 이미지의 비트 반전 비율을 10%, 20%, 30%로 증가함에 따라 정확한 분류값은 10개, 10개, 9개로 줄어든다. 반전 비율이 증가할수록 그리고 일부 이미지의 삭제, 이동 등 훼손되었을 경우는 더욱 정확한 분류값을 얻을 수 없다. 또한 왜곡된 이미지만으로 정확한 학습 데이터를 추출해 낼 수 없다.

본 논문에서 제안된 방법을 적용하였을 경우 정확한 분류값의 패턴 하중값만 선택하면 학습 데이터를 추출할 수 있다.

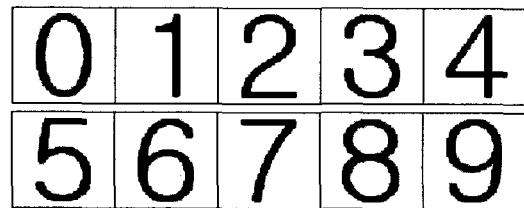


그림 14. 입력 이미지 (숫자 0~9)
Fig. 14. Input Images (Arabic Number 0~9)

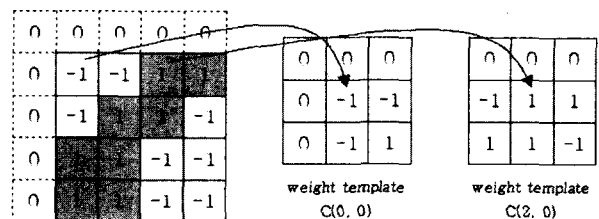


그림 15. 경계셀 연산
Fig. 15. Border Cell Operation

입력 패턴	학습 이미지의 픽셀수	출력의 픽셀수 (r=1 9개의 주변셀)	출력의 픽셀수 (r=2 25개의 주변셀)
0	365	0 365	0 368
1	195	1 194	1 199
2	333	2 332	2 325
3	357	3 357	3 353
4	332	4 336	4 326
5	355	5 355	5 342
6	400	6 401	6 393
7	237	7 236	7 231
8	447	8 446	8 444
9	409	9 414	9 413

그림 16. 패턴별 전체 출력의 픽셀수

Fig. 16. Output Image and the Number of Pixel for Each Pattern

r=1 7 iteration=3, pixel=234
 r=2 7 iteration=9, pixel=198

그림 17. 입력 패턴 7에 대한 수렴과 픽셀수

Fig. 17. Convergence by Iteration and the Number of Pixel for Input Pattern 7

50% 비트 반전된 이미지
 7
 5
 1번 반복후의 출력이미지와 잘못된 분류값(7)
 5
 13번 반복후의 출력이미지와 분류값(5)

그림 18. 왜곡된 이미지에서의 출력
 Fig. 18. Output of Distorted Image

0 5 2 3 4 5 6 7 8 9
 2 9 7 3 2 5 2 3 11 11
 0 0 0 1 0 1 0 0 0 0

그림 19. 부분 하중값의 출력
 Fig. 19. Local Weight Output

그림 19는 부분 하중값 패턴을 적용한 각각의 부분 하중값 출력과 분류값이다. 그 아래에 패턴이 일치하는 3번과 5번 패턴이 표시되어 있다. 전체 하중값 출력의 분류값 7과

일치하는 값은 없으나 그레이 코드에서 가까운 패턴값 5가 선택되어 5번 패턴의 부분 하중값이 선택된다.

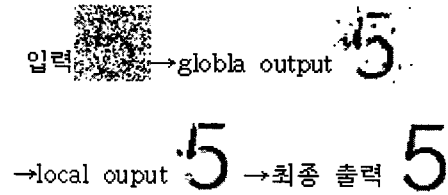


그림 20. 숫자 이미지 분류 인식 과정
 Fig. 20. The Process of Classification and Recognition of the Number Image 5

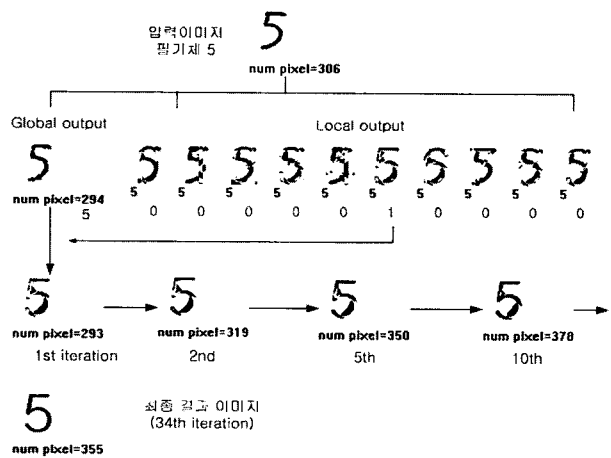


그림 21. 숫자 이미지 분류, 인식, 복원 과정
 Fig. 21. The Process of Classification, Recognition and Restoration of the Number Image 5

4.2 교통표지판 이미지 분류

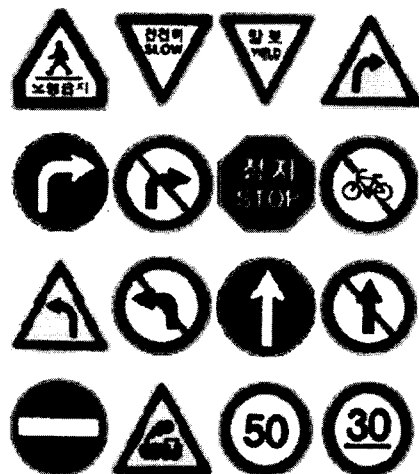
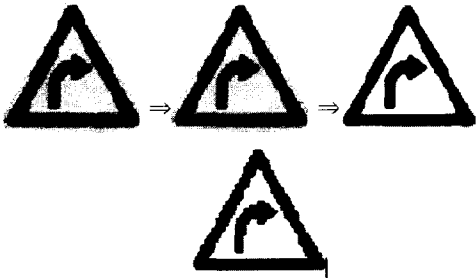


그림 22. 교통표지판 분류에 사용된 16개의 이미지
 Fig. 22. 16 Traffic Sign Images

$$A = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad [= 0.1$$

그림 23. 전처리기 템플릿
Fig. 23. Preprocess Template



컬러 ⇒ 그레이 스케일 ⇒ 전처리기 통과영상 : 그레이 영상의 2진 영상

그림 24. 입력 영상 (우로 굽은 도로)
Fig. 24. Input Image after Preprocessing

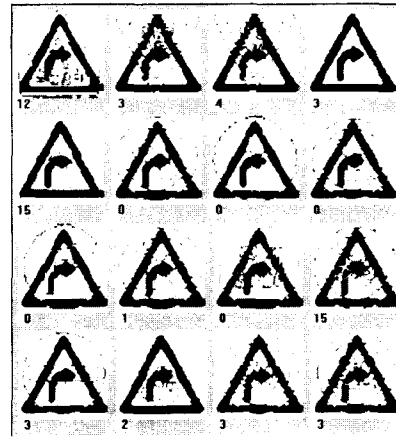
그림 22는 시뮬레이션에 사용한 컬러의 교통표지판 이미지이고, 각각의 이미지는 100×100의 크기를 가진다. RGB 컬러의 이미지를 256 그레이 스케일의 이미지로 변환 후 학습에 사용한다. 학습 이미지의 잡음을 제거하기 위하여 그림 23의 전처리기를 통하여 학습에 사용할 그림 24의 입력 데이터를 얻는다. 그레이 영상은 2진화한 영상보다 부드럽고 잡음이 제거된 영상을 얻는다.

16개의 입력 데이터는 4개의 노드에서 그레이 코드로 분류하고 순서는 그림 25와 같다.

그림 24의 우로 굽은 도로표지판에 대한 출력이 그림 25에 나타나 있다. 우로 굽은 도로의 패턴 번호인 3에서 정확히 부분 하중값을 선택할 수 있고, 최종의 출력 이미지를 볼 수 있다. 학습된 패턴과 같거나 어느 정도의 왜곡된 입력에서는 그림 26의 결과처럼 정확한 패턴의 분류와 인식을 할 수 있다.

패턴이름	출력
0 : 보행자보행금지.bmp	0000 = 0
1 : 서행.bmp	0001 = 1
2 : 양보.bmp	0011 = 2
3 : 우로굽은도로.bmp	0010 = 3
4 : 우회전.bmp	0110 = 4
5 : 우회전금지.bmp	0111 = 5
6 : 일시정지.bmp	0101 = 6
7 : 자전거통행금지.bmp	0100 = 7
8 : 좌로굽은도로.bmp	1100 = 8
9 : 좌회전금지.bmp	1101 = 9
10 : 직진.bmp	1111 = 10
11 : 직진금지.bmp	1110 = 11
12 : 진입금지.bmp	1010 = 12
13 : 횡도건널목.bmp	1011 = 13
14 : 최고속도제한.bmp	1001 = 14
15 : 최저속도제한.bmp	1000 = 15

그림 25. 16가지 입력 패턴과 분류값
Fig. 25. 16 Input Patterns and Their Classification Values



16 학습 패턴의 부분 하중값 출력

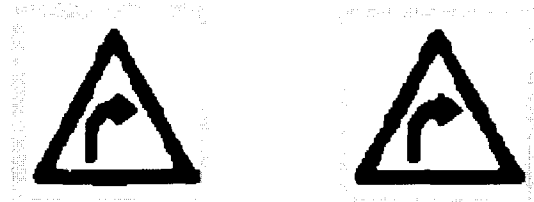


그림 26. 교통표지판 이미지 인식, 복원
Fig. 26. Image Recognition and Restoration

그림 27은 심하게 왜곡된 입력 데이터에 대한 결과이다. 입력은 잡음으로 인해 데이터의 일부가 소실된 영상이다. 위의 경우와 마찬가지로 부분 하중값을 선택하여 원래의 학습 이미지를 복원하고 인식할 수 있다.



(a) 입력 이미지 (컬러컬러⇒그레이스케일⇒전처리기 통과 영상)



(b) global output



(c) local output



(d) 1st iteration



(e) final output

그림 27. 왜곡 이미지의 인식과 복원
Fig. 27. Distorted Image Recognition and Restoration

5. 결론

본 논문에서는 셀룰라 신경회로망을 이용하여 학습과 패턴 매칭을 통한 영상의 분류와 인식 방법을 제안하였으며, 숫자 이미지와 교통 표지판 이미지에 적용하여 그 유용성을 입증하였다.

기존의 Hebb 학습방법만을 이용한 셀룰라 연상 메모리는 입력 이미지의 유사성에 따라 학습률과 학습량에 제한이 있으나 제안된 방법은 메모리의 크기만 충분하면 학습량의 제한을 받지 않는다. CNN의 자체 특성상 이미지 처리에서는 어느 정도 잡음을 제거할 수 있기 때문에 전체 하중값 출력 만으로도 학습된 이미지를 분류하고 인식할 수 있지만, 크게 훼손된 이미지는 분류할 수 없고 학습된 이미지의 모양을 알 수 없다. 이 문제는 부분 하중값 선택으로 적합한 하중값을 선택하여 최종 출력은 학습된 이미지와 같은 출력을 얻을 수 있다. 인간의 뇌를 공학적으로 모델링함으로써 뇌의 기억·연상 메커니즘을 분석하기 위해 이 방법을 인간 두뇌의 연상 작용과 기억 방식에 비교한다면, 하중값 선택과 전체 하중값의 출력은 해마의 연상 작용이라 할 수 있으며, 영구적인 기억은 부분 하중값의 학습이라 할 수 있다.

하중값 패턴을 글자 수에 맞추어 학습을 하면 영문 알파벳에 적용할 수 있고, 한글의 모음 자음 패턴에 적용 하면 음소 단위의 문자인식을 통한 한글 인식도 가능할 것이다.

참 고 문 헌

- [1] L. O. Chua and L. Yang, "Cellular neural networks", IEEE International Symposium on circuits and systems, pp. 985-988, 1988.
- [2] L. O. Chua and L. Yang, "Cellular neural networks: Theory", IEEE Trans. on circuits and systems, vol. 35, pp. 1257-1272, Oct. 1988.
- [3] L. O. Chua and L. Yang, "Cellular neural networks: Applications", IEEE Trans. on circuits and systems, vol. 35, pp. 1273-1290, Oct. 1988.
- [4] J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", Proc. Nat. Acad. Sci., vol. 79, pp.2554-2558, Apr. 1982.
- [5] S. Wolfram, Theory and Applications of Cellular Automata, New York: World Scientific, 1986.
- [6] S. Tan, J. Hao and J. Vandewalle, "Cellular neural networks as a model of associative memories", Proc. of IEEE Int. Workshop on Cellular Neural networks and Their Applications, pp. 26-35, 1990.
- [7] G. Martinelli and R. Perfetti, "Associative memory design using space-varying cellular neural networks for associative memories", Proc. of Symposium on circuits and systems, pp. 549-552, 1994.
- [8] D. Liu and A. N. Michel, "Cellular neural networks for associative memories", IEEE Trans. on circuits and systems, vol. 40, pp. 119-121, 1993.
- [9] M. Brucoli, L. Carnimeo and G. Grassi, "Discrete-time cellular neural networks for

- associative memories: A new design method via iterative learning and forgetting algorithms," Proceedings of the 38th Midwest Symposium on Circuit and Systems, vol. 1, pp. 542-545, 1996.
- [10] R. Caponetto, M. Laborgna, A. Martinez and L. Occhipinti, "Cellular neural network simulator for image processing applications", Proc. of IEEE Int. Workshop on Cellular Neural networks and Their Applications, pp. 360-365, 1998.
 - [11] H. Harrer, P. L. Venetianer and J. A. Nossek, "Some examples of preprocessing analog images with discrete-time cellular neural networks", Proc. of IEEE Int. Workshop on Cellular Neural networks and Their Applications, pp. 201-206, 1994.
 - [12] O. Bandaman, and S. Pudov, "Design and simulations of cellular neural-like associative memory," Int. Workshop on Distributed Interactive Simul. and Real Time App., pp. 49-56, 1997.
 - [13] G. Grassi, "On Discrete-time cellular neural networks for associative memories" IEEE Trans. on Circuits and Systems I, vol. 48, no. 1, pp. 107-201, Jan. 2001.
 - [14] H. Kawabata, M. Nanba and Z. Zhang, "On the associative memories in cellular neural networks," IEEE Int. Conference on Computational Cybernetics and Simulations, vol. 1, pp. 929-933, 1997.

저 자 소 개



신윤철(Yoon-Cheol Shin)

1999년 중앙대학교 제어계측공학과 졸업
(공학사)

2002년동 대학원 전자전기공학부 졸업
(공학석사)

관심분야 : Neural Networks, Cellular Automata and Vision System.

Phone : +82-2-816-8234

Fax : +82-2-816-1856

E-mail : maytime@sirius.cie.cau.ac.kr



박용훈(Yong-Hun Park)

2002년 중앙대학교 전자전기공학부 졸업
(공학사)

2002년~현재 동 대학원 전자전기공학부
석사과정

관심분야 : Intelligent Systems, Robotics
and Neural Networks.

Phone : +82-2-816-8234

Fax : +82-2-816-1856

E-mail : jdpriest@sirius.cie.cau.ac.kr



강 훈(Hoon Kang)

1982년 서울대학교 전자공학과 졸업
(공학사)

1984년 동 대학원 전자공학과 졸업
(공학석사)

1989년 Ph. D Electrical Eng., Georgia
Inst. of Tech(공학박사)

1992년~현재 중앙대학교 전자전기공학부
교수 재직

관심분야 : Fuzzy Systems, Artificial Life, Cellular
Automata, Neural Networks, Pattern Recognition,
Evolutionary Computation, Vision System, Robotics and
Intelligent Control

Phone : +82-2-820-5320

Fax : +82-2-816-1856

E-mail : hkang@cau.ac.kr