

## 시변패턴 인식을 위한 2단 구조의 신경회로망

임정수, 이종호  
인하대학교 전기공학과

### Two stage neural network for spatio-temporal pattern recognition

Chung-soo, Lim, Chong-ho, Lee  
Department of Electrical Engineering, Inha University

#### Abstract

This paper introduces Two-stage neural network that is capable of recognizing spatio-temporal patterns. First stage takes a spatio-temporal pattern as input and compress it into sparse spatio-temporal pattern. Second stage is for temporal pattern recognition with non-uniform inhibitory connections and different cell sizes. These are basic properties for detecting a embeded pattern in a larger pattern. The network is evaluated by computer simulation

#### 1. 서론

자연계에 존재하는 신호는 대부분 시변의 성질을 가지고 있다. 만약 인공지능 시스템을 만들려면 이런 신호들을 인간이 하는 것처럼 처리하는 능력이 필요하다. 즉 기존의 컴퓨터가 할 수 없는 일을 해야 하는 것이다. 이런 방면의 여러연구 중에서 신경회로망을 이용하여 인간의 지능을 모방하려는 시도가 많이 이루어져 왔다. 인공지능분야 중 필연적으로 다루어져야 할 시변패턴의 인식이나 발생은 많은 연구가 진행되었지만 효과적으로 시변패턴을 인식, 발생시키는 신경망 구조는 그리 많이 소개되지 않았다. 이 분야의 모델은 거의 모두가 on-line작업에 부적합하고 출력이 알맞은지 그렇지 않은지를 판단하는 teacher가 필요하다. 이런 견지에서 Supervised learning 보다는 self organizing 능력이 있는 un-supervised learning이 시변패턴 시스템을 구현하는데 더 적합하다고 할 수 있다[1]. 이 시스템은 외부로부터의 새로운 입력에 대해 자체적으로 중요한 정보인지 아니면 잡음인지를 판단하여 인식하게 된다. 본 논문에서 제안된 2단 구조의 신경망은 복잡한 recurrent network의 학습방법이나 불안정적인 성질들을 개선하고자 비교적 단순한 2개의 unsupervised learning networks를 연결하여 이루어진다[3]. 2단구조라함은 각 단이 서로 다른 역할을 담당하는 구조이다. 첫 번째 단은 공간패턴을 압축하는 기능을 수행한다. 이렇게 함으로써 두번째 단에서 이루어지는 시변패턴 인식이 보다 효율적으로 이루어질 수 있다. 두 번째 단에서는 서로 다른 뉴런의 크기와 균일하지 않은 inhibitory connectivity를 이용하여 시변패턴인식을 효율적으로 수행한다. 본 논문에서 제시하는 모델은 3개의 계층으로 이루어지고 입력층은 6개의 뉴런으로 이루어지고 중간층은 32개의 뉴런, 마지막 출력층은 32개의 뉴런으로 이루어진다. 중간층은 첫 번째 단의 출력인 동시에 두 번째 단의 입력이 된다. 시변패턴은 6비트 벡터로 구성된 임의 길이의 열(sequence)로 잡았다.

#### 2. 본론

##### 2.1 첫 번째단 신경회로망

첫 번째 단은 6자리수의 입력을 받아서 32자리수의 출력을 내는데 입력은 이진수가 아닌 영에서 일 사이의 정

규화된 수가 사용된다.

##### 2.1.1 네트워크 구조

구조는 그림1과 같이 되어있는데 두 부분으로 나뉘어져서 3개의 입력뉴런이 16개의 출력뉴런으로 연결되고 또 다른 3개의 입력뉴런도 16개의 다른 출력뉴런으로 연결되어 있다. 이렇게 두 부분으로 나눈 이유는 두 가지가 있는데 첫째는 출력층의 뉴런수가 32개로 제한되어 있기 때문에 6비트의 입력벡터를 한 개의 뉴런에 대응시키는 것 보다는 두 개의 뉴런에 대응시켜서 좀더 표현할 수 있는 입력 벡터의 수를 늘린 것이고 두 번째 이유는 이 네트워크가 경쟁학습을 이용하기 때문에 한 개의 뉴런을 제외하고는 대부분 비슷한 낮은 값을 갖기 때문에 두 번째 뉴런을 선정하기가 쉽지 않으므로 두 부분으로 나누어서 각 부분에서 제일 강한 활성정도를 갖는 뉴런 두 개를 한 개의 입력벡터의 특성을 나타내는 뉴런으로 잡았다.

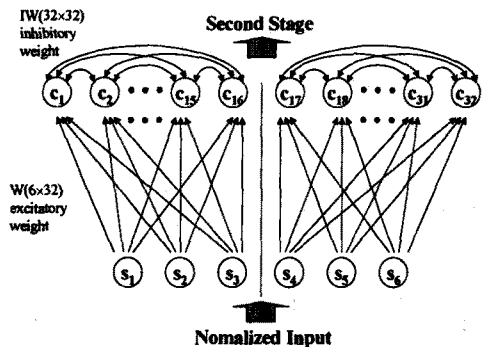


그림1. 첫번째단의 구조

가중치는 흥분성(excitatory)과 억제성(inhibitory)으로 나누어진다. 억제성 가중치의 대각선 성분은 자기자신으로 되먹임되는것을(self-feedback) 나타내므로 0으로 고정된다.

##### 2.1.2 첫 번째단 학습방법

첫 번째단의 학습절차는 다음과 같다.

- 1) 출력뉴런으로 들어가는 총 입력을 구한다.
- 2) 출력뉴런의 활성정도를 가지고 가중치의 증분을 구하고 가중치를 갱신한다.
- 3) 정규화된 입력과 정규화된 가중치를 비교하여 입력을 가장 잘 반영하는 가중치를 구하고 가장 큰 활성크기를 갖는 출력뉴런을 구해 두 뉴런이 일치하는지 검사한다.
- 4) 3번의 결과를 토대로 가장 입력을 잘 반영하는 가중치를 증가시키고 그 가중치와 연결된 출력뉴런으로부터 나가는 억제성 가중치를 증가시킨다. 입력을 잘 반영하지는 못하나 활성크기가 큰 뉴런의 가중치는 감소시킨다.
- 5) 일단 승자뉴런(winning neuron)이 파악되면 그 뉴런의 출력만 1로 하고 다른 뉴런의 출력은 0으로 하여 두

번째 단으로 보내진다.

위의 학습방법을 사용함으로써 가장 큰 가중치를 가지는 뉴런이 다른패턴에 대해서도 계속 활성화되는 것을 방지할 수 있다. 네트워크로 들어가는 입력이 이진수이면 종래의 SOM(self organizing map)의 학습방법이 가능하나 본 논문의 입력은 영부터 일 사이의 값이므로 입력과 비슷한 가중치와 연결된 출력뉴런이 가장 큰 활성크기를 갖는 것이 아니라 가장 큰 크기를 가지는 출력뉴런이 가장 큰 활성크기를 가진다. 이점을 극복하기 위해서 정규화된 입력과 가중치를 비교해 입력과 가장 가까운 상대적 활성정도를 가진 가중치에 연결된 출력뉴런을 승자뉴런으로 만들기 위해 가중치를 적절히 조절한다.

식(1)은 뉴런의 활성정도를 구하는 식이다.

$$C = 0.2 \times C + W^T \cdot S - IW^T \cdot C + W^T \cdot S \quad (1)$$

C는 출력뉴런의 활성정도를 나타내는 행렬이고, S는 입력벡터, W는 가중치 행렬이다.

식(2), (3)은 각각 흥분성 가중치와 억제성 가중치의 증분을 구하는 식이다.

$$\Delta W_{ij} = \eta \times C_j \times (C_i \times S_j - W_{ij}) \quad (2)$$

$$\Delta IW_{ij} = \mu \times C_j \times (1 - IW_{ij}) \quad (3)$$

$\eta$ 와  $\mu$ 는 학습률(learning rate)이다.

## 2.2 두 번째단 신경회로망

32개의 입력뉴런과 32개의 출력뉴런으로 구성되는 두 번째 단은 첫 번째 단으로부터 시변패턴을 받아서 인식하는 역할을 한다.

### 2.2.1 신경망의 주요 특성

두 번째 단에서 쓰이는 신경망은 다음과 같은 특성을 지닌다.

1) 두 번째 단의 출력뉴런은 각기 인식하는 벡터열의 길이에 따라 서로 다른 크기를 갖는다[1]. 벡터열의 길이가 길수록 크기가 커진다. 크기가 커진다는 것은 부피가 커서 활성화가 되기 어렵고 또 일단 활성화가 되면 다시 비활성이 되기 어렵다는 뜻이다. 그리고 크기가 큰 뉴런은 작은 뉴런보다 큰 출력을 낸다. 여기에서 문제가 되는 것은 각 출력뉴런이 인식하는 패턴을 파악하는 방법이다. 일반적으로 생각할 수 있는 것은 가중치의 크기가 되겠다. 그러나 가중치가 크다는 것은 두 가지를 의미한다. 한가지는 입력패턴이 큰 값을 가졌다는 것을 뜻할 수도 있고 또한 그 패턴이 자주 입력되었다는 것을 뜻할 수도 있다. 그러므로 가중치의 크기를 기준으로 뉴런의 크기를 구하기보다는 얼마나 자주 입력되었고 그에 따라 출력뉴런의 활성화가 어느 정도로 이루어 졌느냐가 보다 나은 기준이 될 것이다.

2) 출력뉴런간의 억제성 연결(inhibitory connection)은 균일하지 않다[1]. 억제성 연결의 크기는 억제성 입력을 받는 뉴런의 활성정도에 비례하게 된다. 즉 어떤 패턴이 입력되었을 때 비슷한 패턴을 인식하는 뉴런들이 큰 활성정도를 가진다. 그러므로 서로 비슷한 패턴을 인식하는 것끼리 큰 억제성 연결을 가진다. 왜 비슷한 패턴을 인식하는 뉴런끼리 서로 경쟁해야 하는가를 생각해보자. 만일 벡터열 'AB'와 'CD' 그리고 'ABC'를 인식하는 뉴런이 있다고 가정하자 그리고 입력으로 'ABCD'열이 들어왔다면 균일한 억제성 연결을 가진 경우 'ABC'를 인식하는 뉴런이 가장 큰 활성정도를 가질 것이다. 그러나 이것은 우리가 원하는 것이 아니다. 'ABCD'를 인식하는 새로운 뉴런이 생기거나 'AB'와 'CD'를 인식하는 뉴런이 동시에 활성화 되어야한다. 두 뉴런이 동시에 활성화되기 위해선 'ABC'를 인식하는 뉴런 하나가 'AB' 그리고 'CD'를 인식하는 두 뉴런들과 경쟁을 해야한다. 그러면

'ABC'를 인식하는 뉴런은 크기가 크지만 경쟁에서 지게 됨으로 비활성화 된다. 그러므로 결과적으로 만족할 결과를 얻게된다.

3) 두 번째 단은 시변패턴을 공간패턴으로 변환해서 인식하게 된다. 여기서 변환하는 방법은 시간에 따라 점진적으로 감소하는 패턴을 사용한다[1]. 즉 표1과 같이 변환된다

표1. 점진적으로 감소하는 입력패턴

Time	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>
t <sub>1</sub>	μ	0	0	0
t <sub>2</sub>	μω <sub>2</sub>	μ	0	0
t <sub>3</sub>	μω <sub>2</sub> ω <sub>3</sub>	μω <sub>3</sub>	μ	0
t <sub>4</sub>	μω <sub>2</sub> ω <sub>3</sub> ω <sub>4</sub>	μω <sub>3</sub> ω <sub>4</sub>	μω <sub>4</sub>	μ

μ는 입력뉴런의 기본활성크기이고 다음 시간에 다음 입력이 들어오면 기존의 입력은 ω가 곱해진다. (ω>1)이렇게 점진적으로 감소하는 방법으로 시변패턴을 공간패턴으로 변환시킨다.

### 2.2.2 네트워크 구조

두 번째 단은 첫 번째 단으로부터 32자리수의 입력을 받아서 32개의 뉴런에 대응시킨다. 그림2의 구조에서 보듯이 첫 번째 단의 구조와 거의 동일하다. 차이가 있다면 입력뉴런의 개수가 6개에서 32개로 많아졌다는 것이다.

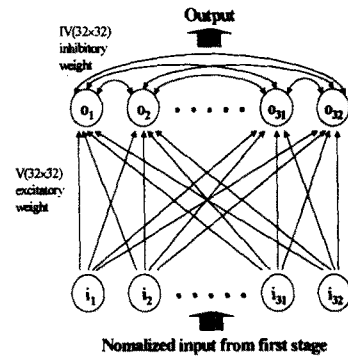


그림2. 두 번째단의 구조

### 2.2.3 두 번째 단 학습방법

두 번째 단의 학습방법은 첫 번째 단으로부터 입력을 받아서 입력으로부터의 흥분성 입력, 다른 뉴런으로부터의 억제성 입력, 그리고 출력뉴런 자신으로부터의 되먹임으로 구성된다. 입력벡터로부터의 총 입력은 입력벡터와 가중치의 내적에 그 가중치가 얼마나 입력을 잘 반영하는지를 나타내는 파라미터(EFF 벡터)를 곱해서 계산된다. 식(4), (5), (6)이 이것을 나타낸다.

$$O_i = 0.2 \times O_i + (\eta_1 \times EXIN_i + INHIN_i) / D_i - (\eta_2 \times INHIN_i) / D_i \quad (4)$$

$$EXIN_i = \left( \sum_j V_{ji} \cdot i_j \right) \times eff_i \quad (5)$$

$$INHIN_i = \sum_j IV_{ji} \cdot EXIN_j \cdot O_j \quad (6)$$

D는 뉴런의 크기를 의미하고  $\eta_1, \eta_2$ 은 학습률이다.  $eff_i$ 는 i번째 출력뉴런으로 연결되는 가중치가 얼마나 입력을 잘 나타내느냐를 나타내는 parameter로 정규화된 입력과 가중치를 비교하여 구해진다.

억제성 가중치와 흥분성 가중치를 구하는 식은 다음과 같다.

$$\text{if } eff_i > eff_j$$

$$\Delta IV_{ij} = \lambda_1 \cdot O_j \times (1 - IV_{ij}) \quad (7)$$

if  $eff_i < eff_j$  and  $IV_{ij} > 0$

$$\Delta IV_{ij} = \lambda_2 \cdot C_j \times (-1 - IV_{ij}) \quad (8)$$

otherwise

$$\Delta IV_{ij} = 0 \quad (9)$$

여기서  $\lambda_1, \lambda_2$  는 학습률이다.

$$\Delta V_{ij} = \xi \cdot O_j \times (i_j - V_{ij}) \quad (10)$$

$\xi$  는 가중치를 학습하는 학습률이다.

출력뉴런이 어떤 패턴을 인식하는지 알아보기 위해서  $U_{ji}$  라는 새로운 가중치( $32 \times 32$ )를 도입한다.  $U_{ji}$  는 j번째 입력뉴런에서 i번째 출력뉴런으로의 가중치이다. 이 가중치는 출력뉴런이 인식하는 패턴을 파악하기 위해서만 사용된다. 식(11)은 입력이 0이 아닌 경우 적용된다.

$$\Delta U_{ji} = \beta \cdot O_i \cdot (O_i - U_{ji}) \quad (11)$$

$\beta$  는 학습률이다. 여기서  $U_{ji}$  는 입력의 크기에 상관없이 증감함을 알 수 있다. U행렬 중 i번째 열에서 가장 큰 값을  $U_{oi}$ 라 하자.  $\theta \times U_{ji} > U_{oi}$  ( $1 \leq \theta < 2$ ) 라면 j번째 입력뉴런은 i번째 출력뉴런이 인식하는 패턴중의 0이 아닌 원소이다. 이 방법으로 각 출력뉴런이 인식하는 패턴의 길이를 알 수 있고 이것으로부터 출력뉴런의 크기 D를 구할 수 있다.

$$D = K^l \quad (12)$$

l는 각 출력뉴런이 인식하는 패턴의 길이, K는 1보다 큰 상수이다.

### 2.3 모의실험

모의실험은 가상의 입력벡터열을 설정하여 시스템의 동작을 matlab으로 검증하였다. 다음의 파형들은 2개의 벡터로 이루어진 열을 입력으로 주었을 때 나타나는 파형이다. 학습률들은 적절히 선정되었다. 이 파형에서 입력패턴을 가장 잘 나타내는 가중치와 연결된 출력뉴런이 활성화되는 것을 알 수 있다. 다음 그래프들에서 가로축은 입력, 출력, 또는 가중치의 색인이고 세로축은 그 크기를 나타낸다.

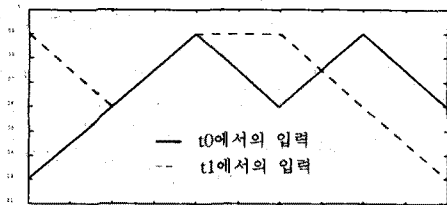


그림3. 첫째단 입력

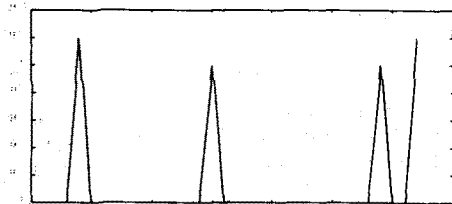


그림4. 두번째단 입력 패턴

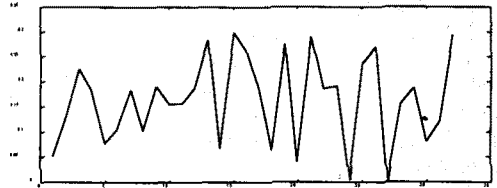


그림5. 학습전 두번째단 승자뉴런의 가중치

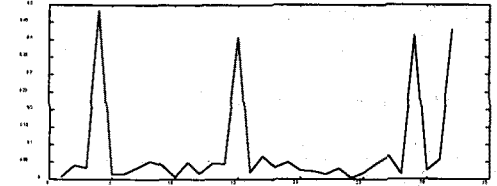


그림6. 학습후 두 번째단 승자뉴런의 가중치

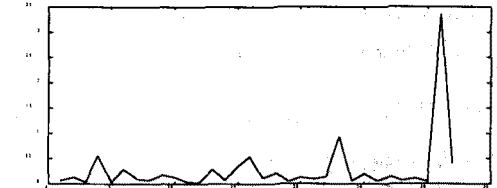


그림7. 두번째단의 출력(6번 학습후)

## 3. 결 론

제안된 시스템은 임의의 길이를 가진 벡터열을 잘 인식하였다. 그러나 벡터열의 길이가 제한되어 있는점, 한 패턴(벡터열)안에서 같은 벡터가 중복되는 경우 인식이 불가능한 것은 개선해야할 점이다. 또한 첫째단에서 둘째단으로 들어가는 입력이 안정된(stable) 값을 가지도록 하는 방법에대한 연구, 그리고 이 시스템이 어느 정도의 분해능(resolution)을 가지고 패턴인식을 할 수 있는지에 대한 분석도 필요하다. 제안된 시스템은 임의의 길이의 시변패턴을 인식하는 능력이 있으므로 음성인식 등에 응용되어질 수 있다.

### (참 고 문 헌)

- [1] Albert Nigrin, Neural Networks For Pattern Recognition, The MIT Press, 1993
- [2] Yifeng Chen and Yuanda Cao, "A Hybrid Neural Network For Spatio-Temporal Pattern Recognition", Proceedings of ICNN, Vol3, pp.1414-1417, 1995
- [3] Bryan W. Stiles and Joydeep Ghosh, "Habituation Based Neural Networks For Spatio-Temporal Classification", Neurocomputing, No15, pp.273-307, 1995
- [4] James A. Freeman and David M. Skapura, Neural Networks Algorithms, Applications, And Programming techniques, Addison-Wesley Publishing company, 1991
- [5] Neil R. Euliano, Jose C. Principe, "Spatio-Temporal Self-Organizing Feature Maps", Proceedings of ICNN, Vol3, pp.1900-1905, 1996