

시변패턴의 저장과 인식을 위한 On-line 연상 메모리의 설계

여성원^{*}, 이종호
인하대학교 전기공학과

On-line Associative Memory Design For Temporal Pattern Storage and Classification

Seong-Won Yeo, Chong-Ho Lee
Dept. of Electrical Engineering, INHA UNIVERSITY

Abstract

Many of the existing neural associative memories are trained and recalled in separate modes and are not suitable for temporal pattern storage and classification in that user must specify the time and length of input patterns. In this paper, a new on-line temporal associative memory model is presented. This memory is structured in layers of neurons and each neuron has limited number of weights so that calculation complexity can be considerably reduced and processing of patterns can be achieved in real time.

1. 서론

실생활에서 인간이 외부 환경으로부터 접하게 되는 대부분의 자극은 공간적으로 다양할 뿐 아니라 시간적으로도 변화한다. 따라서 인간의 생활을 보다 편리하게 만들기 위해서는 인간에게 익숙한 시변패턴을 기계도 지능적으로 처리할 수 있어야 한다.

기존의 인공 신경망은 디지털 컴퓨터가 해결하기 어려운 여러 가지 문제에 있어서 좋은 결과를 보이고 있으나 시변패턴을 저장하고 인식하기 위해서는 구조적으로 적합하지 않으며 따라서 새로운 신경망 모델이 요구된다.

본 논문에서는 시변패턴의 저장과 인식에 적합하도록 새로운 연상 메모리의 신경망 모델을 제안하고 이 메모리의 시변패턴 저장과 인식 능력을 보인다. 제안된 메모리는 On-line 비교사 학습[1]에 기반을 두었다. 여러 개의 채널에서 시간적으로 변화하는 이진 입력 값을 저장하고 향후에 유사한 패턴이 입력되면 출력단에서는 현재 메모리가 저장하고 있는 패턴을 출력시킨다. 또한 시변패턴의 일부만이 입력되었을 때도 기억된 패턴에 근거하여 후속 패턴을 출력할 수 있으며 기억된 패턴과 일부분이 다르게 계속 입력되었을 때 현재 기억한 패턴을 새로운 패턴으로 점진적으로 수정하여 기억하는 On-line 학습을 가능토록 한다.

2. 연상 메모리의 구조

2-1. Encoding 과 Decoding

제안된 연상 메모리는 시간적으로 공급되는 입력패턴을 Encoding 하는 과정을 거쳐 메모리에 전달하며 메모리는 입력패턴과 자신의 예측값에 근거하여 출력을 생성시킨 뒤 Decoding 과정을 거쳐 패턴을 출력한다. 입력패턴이 중단되면 예측값에만 근거하여 패턴을 출력한다.

Encoding 과정은 시간적으로 들어오는 각 채널의 이진 입력값을 검색하여 입력이 중단될 때 이벤트를 생성한다. 이벤트에는 채널별 입력값의 지속시간과 이전 이벤트와의 시간 간격(t)가 할당된다. Decoding 과정은 이벤트에 근거하여 시간적 출력값을 생성한다.

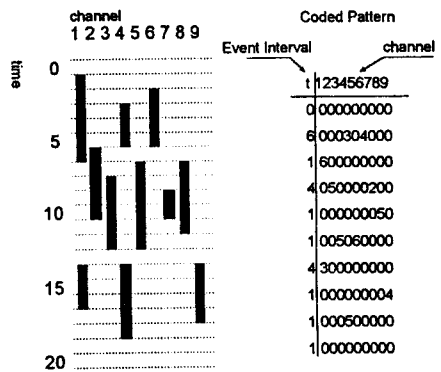


그림 1. Encoding 과정의 예

2-2. 메모리의 구조

하나의 이벤트는 메모리상의 하나의 Layer를 패턴의 형태로 점유한다. 각 Layer에는 채널과 길이에 따라 Neuron이 2차원적으로 배열된다. 메모리는 Layer가 시간적으로 학습이 진행되어감에 따라 배열된다.

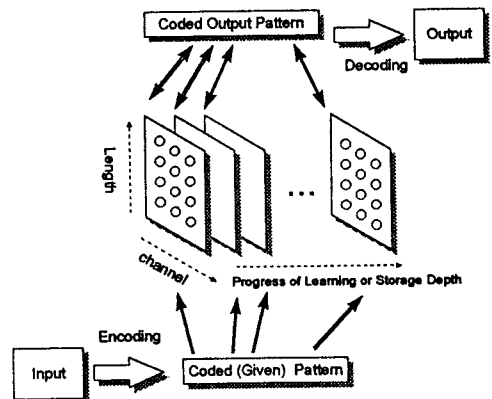


그림 2. 메모리의 구조

2-3. Neuron의 구조 및 Activation 함수

Layer 상의 한 개의 Neuron은 다음 Layer로 연결되는 제한된 수의 Weight를 가진다. 기본적으로 Outstar[4]형에 근간을 두며 자신의 net 값에 Activation 함수를 취하여 출력을 결정한 뒤 Weight를 통하여 다음 Layer의 net 값에 기여한다.

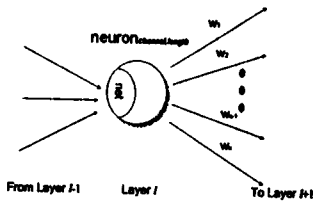


그림 3. Neuron 의 구조

Activation 함수는 이전 출력값을 가지며 이전단에 의해 결정된 현재의 net 값이 1 근방의 값을 가질 때만 1을 출력하게 된다.

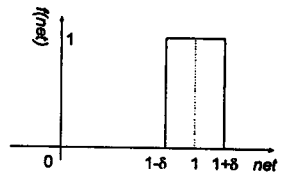


그림 4. Activation 함수

2-4. 패턴의 저장과 출력

연속적으로 입력되는 패턴은 여러 Layer 에 걸쳐 Weight 의 연결로서 저장된다. 현재 Layer 상의 출력에 근거하여 학습에 의해 생성된 Weight 로 출력값을 보내면 다음 Layer 의 예측된 패턴상의 Neuron 의 net 값은 증가 하게 된다. 그림 5 은 학습이 완료된 후의 Weight 연결 예를 보인다.

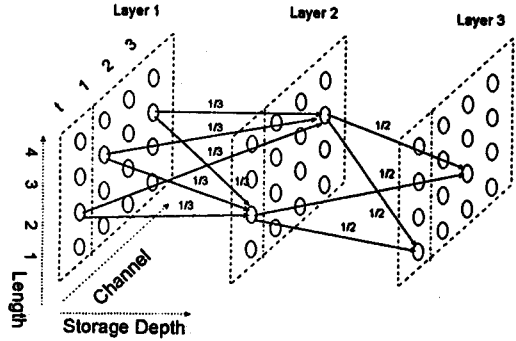


그림 5. 패턴의 저장 예

그림에서 각 Layer 에 저장된 패턴은 지속시간 t 를 포함하여 $p_1=(2,3,0,3)$, $p_2=(2,0,0,3)$, $p_3=(1,0,2,0)$ 이며 Layer 1 에서 Layer 2 로의 Weight 들은 각각 1/3, Layer 2 에서 Layer 3 로의 Weight 들은 각각 1/2 이다. 이는 Layer 1 에서 Layer 2 로의 연결이 3 개의 Neuron 에 의해 이루어지므로 Activation 함수에 의해 1 근방에서 올바른 출력을 이루기 위함이다.

제한된 메모리는 주어지는 패턴에 대한 학습과 인식이 동시에 이루어 지며 이 과정은 한 Layer 상에서 다음같이 수행된다.

Self-Recall Mode의 결정

- Given Pattern이 Zero Input이 지 표시 Zero Input이면 Self-Recall Mode 형성
- Self-Recall Mode중 Nonzero Input이 주어지면 Self-Recall Mode 중단 새로운 입력 받을 준비

Output Pattern의 결정

- Layer상 Neuron들의 net값에 Activation 함수를 취한다.
- Self-Recall Mode이면 1의 값이 나오는 곳을 Output Pattern으로 한다.
- Self-Recall Mode아니면 1의 값이 나오는 곳 중 Given Pattern의 근방에 존재하는 곳만 Output Pattern으로 한다.

i-1단의 Weight Update

- Self-Recall Mode가 아닌 경우에만 수행
- 그림 7 참조

i-1단의 net값 증가

- Output Pattern 또는 Given Pattern상의 Neuron이 가지고 있는 Weight 의 값 만큼 이 Weight가 지시하는 i-1단의 Neuron의 net값을 증가.

그림 6. Layer 의 처리 과정

학습된 시변패턴이 입력 도중 영입력이 들어오는 경우 후속패턴의 출력을 위해 Self-Recall Mode가 활성화 되며 이전단 Layer 에 의해 예측된 패턴에 의해서만 출력을 결정해 나간다.

2-5. Weight 의 변화

Weight 의 Update 는 다음과 같이 처리된다.

- step 1. /단의 Given Pattern의 한 Point를 선택
 $a_i : i=1 \dots N_i, a_i = \{a_{i1}, a_{i2}, a_{i3}, \dots, a_{iN_i}\}$
 N_i = Number of Point in Given Pattern a_i
- step 2. /-1 단의 Given Pattern의 한 Point를 선택
 $a_{i-1} : j=1 \dots N_{i-1}, a_{i-1} = \{a_{i-11}, a_{i-12}, a_{i-13}, \dots, a_{i-1N_{i-1}}\}$
 N_{i-1} = Number of Point in Given Pattern a_{i-1}
- step 3. a_i 에서 / 단으로 연결되는 Weight들에 대해

a. a_i 로 향하는 Weight가 존재하면 \Rightarrow Weight증가

$$W_{a_i \rightarrow a} = W_{a_i \rightarrow a} + c \left(\frac{1}{N_{i-1}} - W_{a_i \rightarrow a} \right)$$

c: Learning Factor

b. 존재하지 않으면

1. a_i 의 근방 (Distance < ρ) 으로 향하면 \Rightarrow Weight감소

$$W_{a_i \rightarrow a} = W_{a_i \rightarrow a} - d \times W_{a_i \rightarrow a}$$

d: Forgetting Factor

2. a_i 로 향하는 Weight생성

$$W_{a_i \rightarrow a}^{new} = c \frac{1}{N_{i-1}}$$

- step 4. /-1 단 Given Pattern의 다음 Point에 대하여 step 2 반복
 step 5. / 단 Given Pattern의 다음 Point에 대하여 step 1 반복

그림 7. /-1 단에서 / 단으로의 Weight Update

출력은 이전단에 의하여 예측된 패턴의 net 에 Activation 을 취한 값 중 현재 주어진 패턴과 일정 범위 안에서 일치되는 값이 되며 주어진 패턴에 예측된 패턴 이 예측하지 못한 새로운 지점이 있을 경우 이전단에서 새로운 지점을 향하여 Weight 를 생성시킨다. 새로운 Weight 는 Neuron 이 소유한 제한된 Weight 중 사용하지 않는 것을 생성 시키며 여분의 Weight 가 없을 시에는 연결 강도가 가장 약한 것을 끊고 새로운 입력을 향해 배치 시킨다.

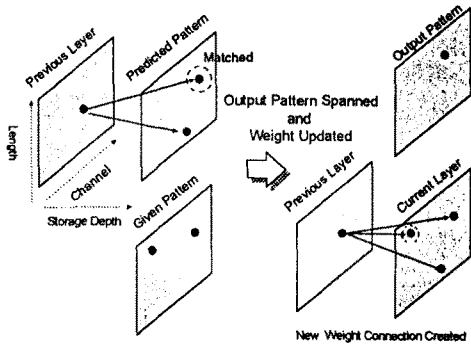


그림 8. Output의 결정과 새로운 Weight 생성

Weight는 예측된 패턴과 주어진 패턴이 정확히 일치되는 경우 강화 시키나 일치하지 않고 근방에 있는 경우 Weight는 감소시키며 주어진 패턴을 행해 새로운 Weight를 생성시킨다. 학습의 반복 후 처음의 Weight는 소멸되며 새로운 Weight가 활성화 된다.

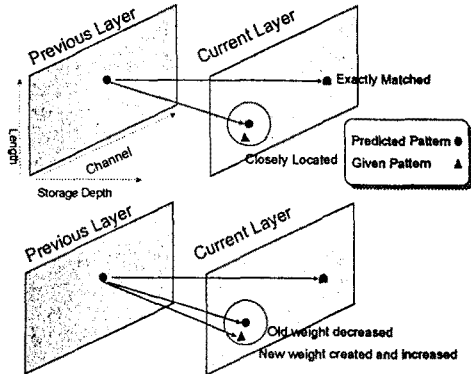


그림 9. Weight의 생성과 소멸

이와 같이 Weight 값을 연속적으로 조정할 때 Weight는 수렴되는 특성을 갖는다. 다음 그림은 학습에 의해 두 개의 Neuron의 Weight가 증가 하며 이것에 의해 다음 단의 net 값이 증가하여 약 4회의 학습이 이루어진 후 Activation 함수의 문턱값을 넘는것을 보인다.

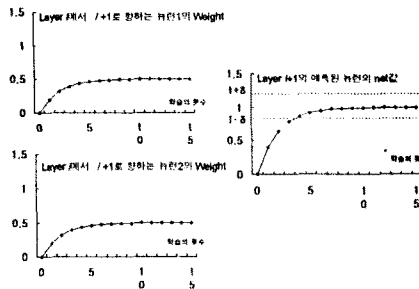


그림 10. 반향학습에 의한 Weight의 증가

3. 실험 결과

그림 11은 2가지의 패턴을 반복적으로 주었을 때 출력에서 기억된 결과가 나타남과 일부의 패턴만 주었을 때 후속 패턴을 출력시킴을 보인다.

여기서 학습계수는 0.4, 기억감쇄계수는 0.11이다.

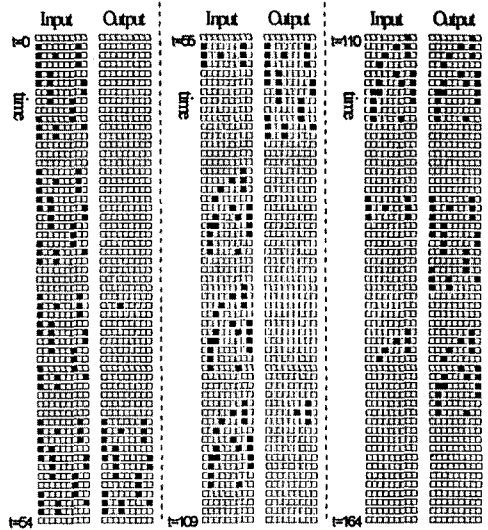


그림 11. 패턴의 기억과 출력

그림 12는 그림 11에서 기억시킨 패턴과 일부 변화된 패턴을 입력시켰을 때 출력이 점차 입력과 일치되는 과정을 보인다.

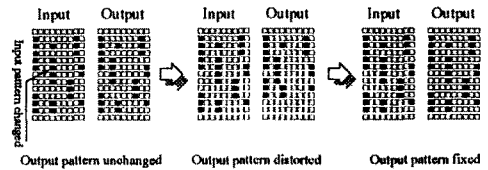


그림 12. 입력의 변화에 의한 출력의 변화

4. 결론

본 논문에서는 시변패턴의 저장과 인식에 적합한 연상 메모리의 신경망 모델을 제안하였다. 제안된 메모리는 많은 양의 패턴에 대한 On-line 학습이 가능하고 Neuron이 가지는 출력속 Weight의 수와 반응하는 Neuron의 수를 제한하여 실시간 동작을 가능케 하였으며 반응하는 Neuron의 근방에서도 Weight의 계산과 출력패턴의 결정이 이루어지므로 패턴의 미세 변화를 극복하며 적용 가능하다. 그러나 Event에 따른 입력패턴의 Encoding이 이루어 지므로 Event 자체의 변화에 Layer가 올바르게 동작할 수 없고 출력단의 Decoding에 따른 Delay가 있다. 따라서 Layer와 Event간의 동기화 문제[3]가 요구되며 후단 Layer를 전단으로 연결하여 보다 긴 패턴에 대한 효율적 저장을 고려해볼 수 있겠다.

5. 참고문헌

- [1] Jacck M. Zurada *Introduction to Artificial Neural Systems*, West info Access, 1992
- [2] John G. Sutherland *The Holographic Neural Method*, Wiley-Interscience, 1992
- [3] Robert Hecht-Nielsen *Neurocomputing*, Addison-Wesley, 1990
- [4] James A. Freeman/David M. Skapura *Neural Networks*, Addison-Wesley, 1991