

홉필드 신경망을 이용한 부분패턴의 복원에 관한 연구

김지훈*, 이주영*, 남궁재찬*

*광운대학교 컴퓨터공학과

e-mail:k98co023@kw.ac.kr

A Study on Partial Pattern Restoration using Hopfield Neural Network

Gi-Hun Kim*, Joo-Young Lee*, Jae-Chan NamKung*

*Dept of Computer Engineering, Kwangwoon University

요 약

본 논문에서는 hopfield 신경망을 사용한 다양한 부분적인 패턴 복원에 관하여 연구하였다. 여섯 개의 32×32 비트맵 훈련패턴들은 한글자음 ㄱ, ㅋ, ㆁ, ㄷ, ㅌ, ㅍ 그리고 남자과 여자 이미지로 구성되어 있다. 그리고 부분패턴들의 크기, 범위, 방향의 효과를 알아보기 위해서 훈련패턴에서 여덟 가지 형태의 테스트 패턴을 만든다. 한글 자음의 경우 유사 패턴이 많기 때문에 완전히 복원되지 못하였으나, 400 회 정도 수렴된 후에는 테스트패턴들이 건본패턴과 비슷한 모양으로 복원되었다. 이 유사도를 측정하기 위해 해밍거리(Hamming distance)를 이용하였다. 유사도를 측정하여 해밍거리가 가장 적은 것으로 본래의 이미지를 복원하였다.

1. 서론

동종 연상 메모리 기능을 갖는 hopfield 신경망은 1970년대 칠패기에 신경망의 연구에 활력을 불어넣은 역할을 하였다[1][2]. hopfield 신경망의 이산 모델과 연속 모델은 신경 연상 메모리에서는 패턴 짝을 학습함으로써 연상작용이 가능하게 된다. 연상 메모리는 자극패턴이 왜곡되었거나 불완전한 경우에도 연상작용을 이용하여 정확한 응답패턴을 생성할 수 있다. hopfield 신경망이 상당한 주목을 받은 이유는 이 연상 메모리를 하나의 에너지 함수인 Liapunov 함수로 표현할 수 있기 때문에 수렴과정을 분석할 수 있고, 대규모 신경망을 통하여 동기신호를 전파할 때 발생하는 문제점을 피하기 위해 뉴런들의 계산을 비동기 방식으로 수정한다는 점이다 [2][3]. 그리고 정보는 분산된 병렬 형태로 저장되므로 각각의 뉴런들이 전체 정보를 갖게 되는 필요성을 줄여 준다는 특징도 가지고 있다. 뉴런들간 연결 하중 값들은 준비된 건본 패턴들로부터 계산된다. 또한 이 신경망은 학습 알고리즘을 사용하여 단일 패스에 의해 쉽게 적응적으로 학습하며 갱신과정에

서 뉴런들의 선택은 랜덤하게 선정되며 모든 뉴런들의 상태가 변하지 않을 때 신경망은 수렴하게 된다. 이러한 hopfield 신경망을 이용한 본 논문의 구성은 다음과 같다. 2장에서는 hopfield 신경망에 대해 설명하고, 3장에서는 hopfield 신경망의 학습과 회상 알고리즘에 대해 논의하고 4장에서 본 연구에 대한 결과에 대해 설명하고, 5장에서 결론에 대해 논의한다.

2. 신경망의 개요

1940년대 초, 맥컬럭(McCulloch)과 피츠(Pitts)는 인간의 두뇌를 수많은 신경세포들로 이루어진 잘 정의된 컴퓨터라고 여겼다. 그들은 단순한 논리적 업무를 수행하는 모델을 보여주었고, 또한 패턴분류 문제가 인간의 지능적인 행위를 규명하는 이론에 매우 중요하다는 것을 인식하였다. 헵(Hebb)의 학습 규칙은 두 뉴런 사이의 연결강도(Weight)를 조정할 수 있는 최초의 규칙이었다. 이 규칙은 학습에 관한 연구를 발전시켰으며 적응적인(adaptive) 신경망 연구에 많은 영향을 끼쳤다[3][5]. 로젠블러트(Rosenblatt)은 1957년 퍼셉트론(perceptron)이란 최초의 신경망 모델을 발표하였는데, 여기서는 학습 프로세스에

알파강화 규칙을 사용하였다. 퍼셉트론 모델은 그 당시에는 매우 가능성이 큰 것으로 여겼으나, 그 후 XOR 함수와 같이 단순한 비선형 분리 문제도 풀 수 없는 것으로 밝혀지며 신경망에 관련된 연구는 약 20년간 침체의 길을 걷게 되었다. 그 후 1980년대 전후에 들어 코호넨(Kohonen), 홉필드(Hopfield), 커크패이트릭(Kirkpatrick), 헌턴(Honton), 그로스버그(Grossberg), 러멜하트(Rumelhart) 등이 신경망을 다시 활성화 시켰다. hopfield 모델은 연상 기억과 최적화문제의 해를 구하는데 매우 유용한 모델로 주목받고 있다[1][2][4].

2.1 홉필드 신경망

1982년 미국의 존 홉필드에 의해 제안된 hopfield 신경망은 물리학적 스핀 모델로부터 hopfield 신경망을 착안하였다. hopfield 신경망은 많은 수의 비동기적이고 국부적인 계산을 통하여 최적화 문제를 해결할 수 있을 뿐만 아니라 내용을 주소로 이용하는 연상 메모리로도 사용할 수 있다. 상호 결합형 신경망인 hopfield 신경망은 연상기억이나 최적화 문제를 푸는데 많이 사용되는데, 에너지 함수를 사용하는 hopfield 신경망은 다음과 같은 두 가지의 제한점이 존재한다[2][4].

첫째 : 많은 패턴을 저장하려고 있으면 수렴할 때 잘못된 패턴으로 도달될 수 있다.

둘째 : 대표패턴이 다른 대표패턴과 유사하여 많은 비트들을 공유한다면 신경망이 불안정해진다.

2.1.1 학습

hopfield 신경망에서는 가장 먼저 메모리에 저장될 훈련 패턴들이 준비되어야 하며 뉴런들의 수는 각 패턴들을 구성하는 요소들의 수와 같다. 그리고 연결 하중 값은 패턴들에 따라 유일하게 결정된다. 이 신경망은 단층이며 뉴런들이 완전 연결된 구조이다. 뉴런 i 에서 뉴런 j 로의 연결 하중을 W_{ij} 이라 하면 (A_1, A_2, \dots, A_m) 과 같이 m 개의 패턴이 있다고 하면, 패턴 P 는 $A_p = (a_{p1}, a_{p2}, \dots, a_{pm})$ 과 같이 표현되며 여기서 a_{pi} 는 p 번째 훈련벡터(training vector)의 i 번째 요소이다. 그러면 hopfield 신경망에서 i 번째와 j 번째 뉴런간의 하중값은

$$W_{ij} = \frac{1}{n} (2a_{pi} - 1)(2a_{pj} - 1) \quad \text{식(1)}$$

이다. W_{ij} 는 a_{pi} 및 a_{pj} 일 때 1 증가한다. 이러한 증가 또는 감소 과정은 모든 패턴들 내에 있는 모든 짝들 i, j 에 대해서 수행된다. 그리고 하나의 패턴이 신경망에 제공될 때마다 하나의 상태벡터가 발생된다.

신경망에 기억되는 기본패턴에 대응하는 신경망의 에너지 상태가 최소가 되도록 학습시키고자 한다. m 개의 훈련 벡터들의 주어진 집합 A^p 를 신경망에 기억시키려면 훈련 방정식인 식(1)을 이용한다[3][4].

2.1.2 갱신 과정

대부분의 다른 신경망들은 하나의 층에 있는 뉴런들에 대해서 동시에 갱신되지만 hopfield 신경망에서는 비동기적으로 한 번에 하나씩 갱신되며 이 방법은 뉴런에 가해지는 사건에 따라 상태를 바꾸는 생물학적 뉴런의 동작과 아주 비슷하다. 이 신경망의 갱신과정은 다음과 같다.

뉴런 j 에 대한 자극은 식(2)이다.

$$S_j = \sum_{i=1, i \neq j}^n u_i W_{ji} \quad \text{식(2)}$$

여기서 u_i 는 뉴런 j 에 연결된 입력이며 뉴런 j 의 출력은 u_j 가 1이면 자극은 0보다 크게되고, u_j 가 0이면 자극은 0보다 작게 된다. 이러한 뉴런 j 의 상태 변화는 이 뉴런의 입력 단에 연결된 다른 뉴런들의 상태가 변했기 때문에 발생한 것이다. 특정 뉴런의 갱신 전 값은 그 뉴런의 갱신시에는 고려되지 않으며 갱신과정은 각 뉴런의 상태에 영향을 미친다. 갱신과정은 뉴런들의 상태가 어떤 변화도 발생하지 않을 때까지 순차적으로 계속되며 궁극적으로 최소의 에너지 상태로 수렴됨이 보장된다[2][3].

2.1.3 수렴

hopfield 신경망에서 국부 최소점(local minimum)에서 전역 최소점(global minimum)을 다시 찾아가는 방법은 없으며 이렇게 하려면 잡음을 추가하여 신경망을 흔드는 방법을 사용하여야 한다. 그러나 신경망의 초기 위치를 다시 설정하여 다른 초기 위치에서 수렴과정이 시작되면 전역 최소점을 찾을 수 있다. 먼저 단 하나의 뉴런이 갱신됨에 따른 에너지의 변화를 살펴보기로 한다. 뉴런 j 에 의한 에너지는 식(3)로 표현된다.

$$E = -\left(\frac{1}{2}\right) \sum_{i,j} W_{ij} u_i u_j \quad \text{식(3)}$$

여기서 u_i 는 i 번째 뉴런의 출력이다. 식(3)에서 합은 뉴런 i 에 대해서만 취해지기 때문에 식(3)은 식(4)로 나타낼 수 있다.

$$E_j = u_j \left[-\left(\frac{1}{2}\right) \sum_{i,j} W_{ij} u_i \right] \quad \text{식(4)}$$

그러므로 뉴런 j 의 상태변화가 에너지의 변화량 ΔE_j 를 발생시킨다. 즉,

$$\Delta E_j = \Delta u_j \left[- \left(\frac{1}{2} \right) \sum_{i,j=1}^n W_{ij} u_i \right] \quad \text{식(5)}$$

hopfield 신경망의 전체 상태는 식(7)처럼 하나의 에너지 함수로 나타낼 수 있다.

$$E = - \left(\frac{1}{2} \right) \sum_j \sum_{i,i \neq j} W_{ij} u_i u_j \quad \text{식(6)}$$

식(6)의 에너지 함수는 신경망에 의해 최소화 될 수 있는 하나의 목적함수이다. hopfield 신경망의 연속적인 갱신과정은 이 에너지 함수가 점차적으로 작아지는 하나의 수렴과정이며 결국 신경망은 국부적 또는 전역적인 에너지 최소상태로 안정된 상태에 도달한다. 이러한 안정상태에 도달하면 어떤 뉴런도 그들의 상태를 바꾸지 않는다. hopfield 신경망이 영원히 그 상태를 유지하는 하나의 평형상태이며 또한 하나의 어트랙터(attractor)에 해당된다. hopfield 신경망에서 고려해야 할 중요한 사항은 하나의 주어진 입력패턴에 대하여 저장된 패턴이 아주 비슷하게 닮게 하는 기준과 이 신경망의 기억 용량에 관련되는 사항이다. 현재까지 알려진 연구 결과는 혼련집합의 비적교성 멤버들이 가상의 어트랙터(spurious attractor)를 발생시키는 원인이며 이 경우는 안정된 최소에너지 상태가 혼련 패턴에 대응하지 않는다는 것이다. 또한 입력패턴의 부의(negative)상태 또한 하나의 어트랙터가 된다는 바람직하지 않은 특성도 있다. hopfield 신경망은 만일 패턴에 잡음이 인가된 경우나 부분적으로 훼손된 경우에도 메모리에 저장된 기본패턴과 충분히 닮았으면 정확한 패턴을 복원하는 기능을 갖고 있다[1][3][4][5].

3. Hopfield 모델의 학습과 회상 알고리즘

3.1 Hofield 모델의 학습 알고리즘

Hopfield 모델은 학습 알고리즘 다음과 같다.

단계1. 이진값 in_0 에서 in_{n-1} 값을 갖는 in 이라 불리는 길이 N 의 입력벡터가 주어진다.

단계2. out 이라 불리는 in 벡터와 같은 형태의 출력벡터를 만든다.

```
for( i = 0; i = N-1; I++) outi = ini;
```

단계3. W 라 불리는 weight matrix $N \times N$ 을 초기값이 모두 0 상태로 구성한다.

```
for( i = 0; i = N-1; I++) {
    for( j = 0; j = N-1; j++) Wij = 0;
}
```

단계4. $i \neq j$ 인 W_{ij} 의 각 요소에 대한 적절한 값을 다음 식에 의해서 구한다.

$$W_{ij} = (2in_i - 1)(2out_j - 1)$$

이 식의 의미는

- 만약 in_i 와 out_j 가 둘 다 1이거나 둘 다 0일 때 W_{ij} 요소에 1을 기억시킨다.
- 그렇지 않으면 W_{ij} 값은 -1이 된다.

```
for(i = 0; i = N-1; I++) {
    for(j = 0; j = N-1; j++) {
        if(i != j) Wij = (2ini - 1)(2outj - 1)
    }
}
```

3.2 Hofield 모델의 회상 알고리즘

Hopfield 회상 알고리즘 다음과 같다.

조정된 하중행렬은 벡터 01110을 갖고 있는 상태이다. 하중행렬로부터 이 벡터를 회상하기 위해서 다음 알고리즘을 사용한다.

단계1. 길이 N 인 벡터를 기억하는 하중행렬이 주어진다.

단계2. 출력요소(output element) j 에 들어오는 모든 연결강도(connection strength)값을 합산한다.

```
for(j = 0; j = N-1; j++) {
    SUM = 0;
    for(i = 0; i = N-1; I++) SUM = SUM + Wij;
    netj = SUM
}
```

단계3. 벡터 net를 테스트하고, 값을 구한다.

```
for(j = 0; j = N-1; j++) {
    if netj >= 0 netj = 1;
    else netj = 0;
}
```

단계4. 단계3으로부터 net 벡터의 값을 구한다.

4. 실험 및 결과

본 연구에서는 한글 자음 부분패턴의 방향성과 패턴 영역의 크기 등의 패턴 영역의 변화에 따른 hopfield



그림 41 32x32 전분패턴

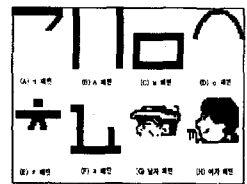


그림 42 32x32 시험 패턴

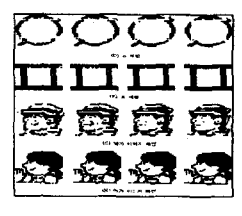
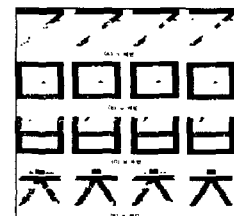


그림 43 각 시험패턴들의 복원 과정

<Continued>

신경망의 복원 특성과 양호한 복원을 가능하게 하는 조건들에 관해서 연구하였다. 그림 4.1과 같이 한글 자음으로 32×32크기의 견본패턴 8개를 신경망에 학습시키고 그림 4.2와 같은 8개의 시험패턴에 대한 복원성능에 관해서 연구하였다. 실험에는 각 견본패턴들의 비트 수가 1024개이므로 hopfield 신경망 뉴런이 1024개 필요하다. 그림 4.1의 8개의 견본패턴들이 hopfield 신경망에 순서대로 제공되고, 그림 4.2의 시험패턴이 입력된다. 8개의 견본패턴들을 이용하여 하중값들이 계산되어 신경망에 저장된다. 이 과정은 일종의 학습과정에 해당한다. 그리고 뉴런들간의 연결선들은 완전 연결된 구조이며 하중값들은 식 (1)에 의해 계산되어 저장된다. 그림 4.1은 견본패턴이고, 그림 4.2는 시험패턴이다. 이 시험패턴들은 견본패턴들의 각 부분패턴들이다. 아래 그림 4.3은 각 견본패턴들의 부분패턴이 100회, 200회, 300회, 400회 반복하여 복원해 가는 과정을 보여준다. 각 시험패턴들은 400회 가량 반복하여 갱신하였을 경우 local minima에 빠져 더 이상의 변화를 보이지 않았다. 하지만 다음의 표에서 보여 주듯이 수렴된 패턴과 견본패턴간의 해밍거리를 비교해 보면 상당히 유사함을 알 수 있다.

<표 4.1> 견본패턴간의 해밍거리

X	A	B	C	D	E	F	G	H
A	0	394	544	361	400	492	542	491
B	38	0	192	355	616	624	524	521
C	54	192	0	437	600	608	498	601
D	36	355	437	0	425	439	479	482
E	40	616	600	425	0	496	460	481
F	49	624	608	439	496	0	444	479
G	54	524	488	479	460	444	0	433
H	49	521	601	482	481	479	433	0

<표 4.2> 견본패턴과 시험패턴간의 해밍거리

X	A	B	C	D	E	F	G	H
A	84	332	524	301	392	446	526	469
B	94	96	288	339	560	576	606	507
C	524	288	104	381	528	568	448	573
D	301	339	381	112	349	435	473	474
E	392	560	528	349	128	468	448	451
F	446	576	568	435	468	128	464	599
G	526	606	448	473	448	464	208	425
H	469	507	573	474	451	599	425	96

<표 4.3> 견본패턴과 수렴패턴간의 해밍거리

X	A	B	C	D	E	F	G	H
A	58	394	498	303	342	462	520	481
B	384	12	180	355	628	636	520	529
C	498	180	9	432	597	617	479	608
D	303	355	432	117	430	526	504	479
E	342	628	597	430	0	496	460	481
F	462	636	617	526	496	0	444	479
G	520	520	479	504	460	444	129	358
H	481	529	608	479	481	479	358	0

<표 4.3>을 보게 되면 견본패턴과 똑같이 수렴되지 않더라도 해밍거리가 적음을 알 수 있다. 따라서 완전하게 복원되지 않은 시험패턴 그림 4.2의 (A), (D), (G)는 견본패턴 그림 4.1의 (A), (D), (G)라고 볼 수 있다. 수렴된 패턴과 견본패턴의 오차율은 13% 이내였다. <표 4.2>에서 알 수 있듯이 시험패

턴과 견본패턴의 유사도가 높을수록 복원이 잘 되었으며, 시험패턴과 견본패턴의 해밍거리가 가장 먼 (G)패턴의 경우 수렴된 패턴과 견본패턴간의 해밍거리 또한 멀다는 것을 알 수 있다.

5. 결론

한글 자음의 경우 유사 패턴이 많기 때문에 완전히 복원되지 못하였으나, 400회 정도 수렴된 후에는 테스트패턴들이 견본패턴과 비슷한 모양으로 복원되었다. 위 실험 사실들로부터 견본패턴들의 해밍거리로 계산되는 유사도가 클수록 부분 패턴에 대응하는 견본패턴들의 복원 성공률은 낮아진다. 그리고 현재 본 논문의 hopfield 네트워크는 9개 이상의 이미지를 입력 시켰을 때 몇몇 개는 local minima에 빠져 제대로 복원이 되지 않는 문제점이 생겼다. hopfield 신경망은 전혀 답지 않은 정보들의 인식 및 복원에 대해서는 우수한 동작을 가졌지만 서로 답은 부분이 있는 정보들의 인식 및 복원의 결과는 불량함을 확인하였다.

참 고 문 헌

- [1] Arun D. Kulkarni, "Artificial Neural Network for Image Understanding", Van Nostrand Reinhold, PP128~148, 1994.
- [2] A. S. Panda and R. B. Macy, "Pattern Recognition with Neural Systems", West Pub, PP281~318, 1996.
- [3] Freeman, J. A. and Skapura, D. M., "Neural Networks: Algorithms, Applications and Programming Techniques", Addison-Wesely, Reading, M. A, PP127~148, 1992.
- [4] Hopfield J., "Neural Network and Physical Systems with Emergent Collective Computational abilities", Proceeding of The Neural Academy Science, 1982
- [5] Jacek. M. Zurada, "Introduction to Artificial Neural Systems", West Pub, PP313~354, 1992.