

주기적 패턴 쌍을 저장하는 Recurrent Neural Network를 찾는 진화 알고리즘

김권일⁰ 장병탁

서울대학교 컴퓨터공학부 바이오지능 연구실
{kikim, btzhang}@bi.snu.ac.kr

Evolutionary Algorithm for Recurrent Neural Networks Storing Periodic Pattern Pairs

Kwonill Kim⁰, and Byoung-Tak Zhang

BI Lab, School of Computer Sci. & Eng., Seoul National University

요 약

뇌 속 뉴런들의 네트워크는 근본적으로 recurrent neural networks(RNNs)의 형태를 지닌다. 이 논문에서는 반복되는 뉴런 반응 패턴들 사이의 관계를 네트워크에 저장함으로써 생물의 기억이 생성된다는 가정하에, 이를 표현할 수 있는 RNN 모델을 제안하였고, evolutionary algorithm을 통해 이러한 여러 쌍의 기억들이 저장된 네트워크가 존재할 수 있음을 보였다.

1. 서 론

생물의 뇌는 신경세포(Neuron)들의 연결을 통해 각 신경세포의 발화(firing)로 부호화된 신호를 전달하고 정보를 처리한다[1]. 이들 신경세포로 이루어진 네트워크는 수 많은 feedback loop들을 가진 recurrent neural network(RNN)의 특징을 가지고 있다[1, 2].

RNN 모델을 사용하는 학습과 기억에 대한 연구는 그동안 많이 이루어져 왔다[3, 4]. 하지만, 아직 충분히 효율적으로 RNN을 학습시키는 알고리즘은 개발되지 않고 있다. 본 논문에서는 진화연산을 통하여 특정 기억을 학습한 RNN이 존재하는지를 보임으로써 어느 정도의 모델링으로 RNN 학습 알고리즘 개발이 가능한지 알아보려 한다.

2장에서는 진화연산과 신경망에 적용한 진화 기법들에 대해 간략히 소개하고, 3장에서 RNN 모델에 대해 설명한 다음, 4장과 5장에서 실험과 그 결과에 대해 이야기하겠다.

2. 진화연산 (Evolutionary Computation)

진화 연산은 다수의 해(염색체)들을 자연선택에 기반하여 여러 세대 진화시켜 최적해를 구한다. 진화연산의 일반적인 알고리즘은 box 1과 같은데, 먼저 n개의 해를 임의로 생성한 다음 이 해집단으로부터 선택(selection), 교차(crossover), 변이(mutation)단계를 거치면서 k개의 새로운 해를 만들어내어 해집단 내의 k개의 해와 대치한다. 이러한 과정을 정지 조건이 만족될 때까지 수행한 후 가장 좋은 해를 최종 답으로 삼는다[5].

```

n개의 초기 염색체 생성;
repeat {
    for i = 1 to k {
        두 염색체 p1, p2 선택;
        offspringi = crossover(p1, p2);
        offspringi = mutation(offspringi);
    }
    offspring1, ..., offspringk를 해집단 내의 k개의 해와 대치
} until (정지 조건 만족);
남은 해 중 최상의 염색체를 return;
    
```

Box 1 유전알고리즘의 전형적 구조 (문병로, 유전알고리즘, 2003)

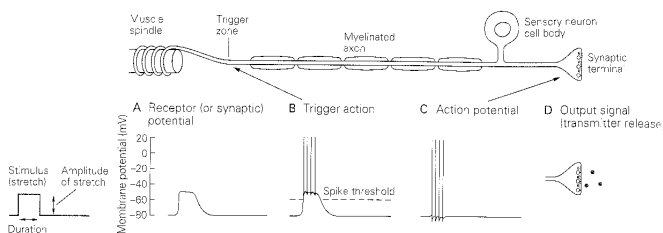


그림 1 신경세포의 발화(firing)을 통한 신호 전달 (E. R. Kendal et al., Principles of Neural Science, 2000)

2.1 Neural Network에 대한 진화연산

Neural network를 진화시키는 다양한 연구가 진행되어왔다. 초기에는 일정한 구조의 네트워크에서 weight 값만 진화시키거나, 구조만 진화시키면서 weight는 backpropagation과 같은 알고리즘을 사용해서 최적화하는 연구들이 많이 행해졌었다. 이후 구조와 weight를 동시에 최적화하는 방법들이 많이 연구되면서 기존 진화 알고리즘을 네트워크에 바로 적용할 때 생기는 몇 가지 문제점들이 알려지게 되었다. 그 중에서 대표적인 것이 crossover를 했을 때 네트워크의 좋은 스키마들이 망가져버릴 확률이 무척 높다는 문제다. 이 때문에 본 논문에서 사용한 GNARL 알고리즘에서는 crossover를 사용하지 않고 mutation만으로 네트워크를 진화시켜 나간다[6].

2.2 GNARL(GeNeralized Acquisition of Recurrent Links) Algorithm

Neural network의 구조와 weight를 동시에 진화시키는 GNARL 알고리즘은 crossover를 사용하지 않는다는 특징을 지닌다. mutation을 통하여 점진적인 변화를 추구하는데, 이 때 뉴런이나 뉴런 사이의 연결이 추가/삭제될 확률, weight 값이 변화하는 정도 등을 network's temperature, $T(\eta)$ 에 비례하도록 하였다. 즉 적합도가 최대값에 근접할수록 network's temperature는 점점 낮아지면서 mutation의 정도가 줄어들게 된다. 이는 최적해에 가까워진 네트워크가 강한 mutation에 의해 파괴되는 것을 어느 정도 방지해준다[6].

3. Modeling

생물의 기억은 그간의 연구들을 통해 다양한 종류로 분류된다[1]. 하지만 본 논문에서는 그림 2와 같은

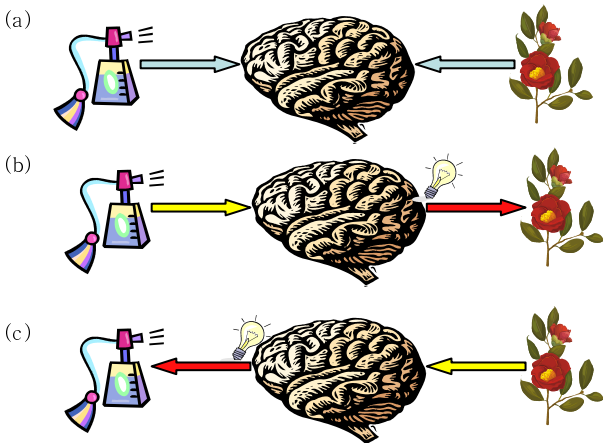


그림 2 뇌에서 일어나는 연상 기억의 예. (a) 꽃의 모습과 향기가 동시에 입력되자, 서로 연관되어 기억된다. (b) (c) 기억된 이후 향기나 모습 중 하나만 접하더라도 다른 하나를 떠올리게 된다.

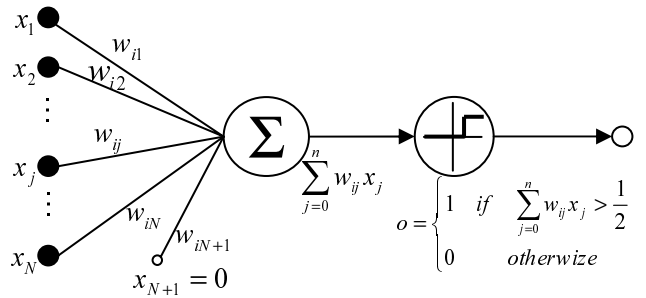


그림 3 모델링에 사용된 perceptron.

단순화된 연상 기억 현상만을 고려할 것이다.

어떤 사람이 난생 처음 이름 모를 꽃을 보고 그 향기를 맡았다고 하자. 이 경우 꽃의 모습과 향기가 연관되어 기억된다. 이후 이 사람은 그 향기를 다시 맡으면 꽃의 모습을 떠올리게 되고, 꽃의 사진을 보면 향기를 떠올리게 될 것이다.

이런 상황을 보다 단순하게 모델링한 것이 그림 4와 같은 RNN이다. 각 뉴런은 그림 3에 표현된 기본적인 perceptron이며 외부와 연결된 단말뉴런(terminal neuron)을 통해 입출력이 이루어 진다. 시간이 t일 때 i번째 뉴런의 발화여부를 $x_i(t) \in \{0,1\}$ 라 하고, i번째 뉴런이 주기 T 동안 발화한 패턴을 $\mathbf{m}_i(t) = [x_i(1) x_i(2) \dots x_i(T)]$ 라 하자. 먼저 저장 단계에서는 단말 역할을 하는 1번과 2번 뉴런을 각각 주기적 패턴 $\mathbf{m}_1, \mathbf{m}_2$ 로 동시에 발화시킨다. 이는 네트워크 전체에서 주기적인 발화를 일으키고, 이러한 반복되는 발화에 네트워크가 적응하는 방향으로 weight matrix \mathbf{W} 가 서서히 변화되어, 결국 네트워크가 뉴런 1, 2로 들어온 패턴 $\mathbf{m}_1, \mathbf{m}_2$ 를 기억하게 된다. 그런 다음, 회상 단계에서는 하나의 단말뉴런에만 저장단계에서 기억시켰던 패턴을 입력하여 네트워크 전체를 발화시키고, 다른 단말뉴런에서 짝으로 기억시켰던 패턴을 조회해 낸다. 예를 들자면 뉴런 1에만 \mathbf{m}_1 을 주기적으로 입력하여 그대로 발화시키면,

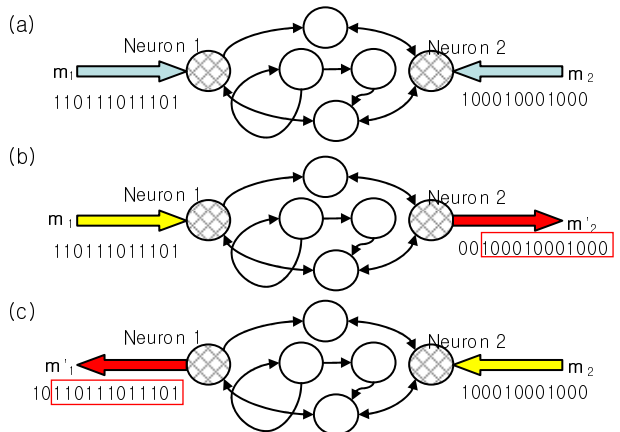


그림 4 이진수열과 RNN을 적용한 연상기억 모델. (a) 패턴들을 반복 입력하여 네트워크의 weight들을 변화시킨다. (b) (c) 성공적으로 기억된 네트워크의 한 뉴런에 일부 패턴을 입력하면 다른 뉴런에서 쌍이 되는 패턴이 출력된다.

그 패턴을 기억하고 있던 네트워크가 적절하게 발화하여 잠시 후부터 뉴런 2에서도 m_2' 로 주기적 발화가 일어난다. 마찬가지로 뉴런 2에만 m_2 를 입력했을 때에도 뉴런 1에서는 m_1' 의 패턴으로 발화가 일어날 것이다. 이 때 $m_1' = m_1$, $m_2' = m_2$ 이면 m_1 과 m_2 가 “성공적으로 연관되어 저장되었다.”고 하자. 나아가 뇌가 엄청난 양의 기억을 하듯이 RNN 모델 역시 K개의 패턴 쌍 $M = \{(m_1^{(1)}, m_2^{(1)}), (m_1^{(2)}, m_2^{(2)}), \dots, (m_1^{(K)}, m_2^{(K)})\}$ 을 모두 저장할 수 있어야 한다.

4. 실험

앞서 3장에서 제안한 RNN모델에는 저장 단계에서 네트워크의 반복적인 발화에 기반하여 weight matrix를 학습시키는 알고리즘이 필수적이다. 본 논문에서는 RNN 학습 알고리즘을 개발하기 전에, 먼저 이러한 알고리즘 개발이 가능한지, 다시 말해서 패턴 쌍들이 성공적으로 저장된 RNN이 존재하는가에 대해 확인해보려고 한다. 이를 위해 앞에서 소개한 GNARL 알고리즘을 사용해서 임의의 패턴 쌍들이 성공적으로 연관 저장된 RNN을 탐색해 보았다.

그림 3의 단순한 perceptron으로 이루어진 RNN을 $(N+1) \times (N+1)$ weight matrix W 로 표현하였고, 이들을 mutation하여 5배수의 자손을 만든 다음, 적합도를 계산하여 1/5만을 다음 세대로 선택하였다. 적합도는 각 RNN을 회상 단계로 실행하여 $m_i^{(k)'}$ 와 $m_i^{(k)}$ 사이의 차이, 전체 네트워크에서 발생한 발화의 횟수, 그리고 뉴런과 연결의 개수를 적용하여 계산하였다. 가장 높은 적합도를 보이는 개체는 mutation과 selection없이 다음 세대로 넘기는 elitism 기법도 적용했다. Population size는 80, 최대 10000세대까지 진화시켜보았다.

5. 결과

저장할 패턴 쌍의 집합 M 을 패턴의 주기 T 와 패턴 쌍의 개수 K 에 따라 무작위로 생성하고, 이를 성공적으로 저장한 RNN을 탐색하였다. ($T=4, K=2$)와 ($T=5, K=2$)인 집합 M 의 경우 탐색은 높은 확률로 성공했다. ($T=5, K=3$)인 경우에는 발견 성공률이 높지 않았으나 성공적으로 패턴들을 저장하고 있는 RNN이 존재함을 확인할 수 있었다.

그림 5는 진화연산을 통해 발견된 주어진 패턴을 성공적으로 저장한 RNN들이다. 저장된 패턴과 그래프, weight값, bias값들을 기록하였다.

6. 결론

본 논문에서는 뇌에서 두 가지의 정보를 연관 지어 기억하는 현상을 모사하는 RNN모델을 제시하였다. 두 개의 이진패턴들이 RNN으로 동시에 반복해서 입력되면 네트워크 역시 특정 패턴으로 주기적 발화를

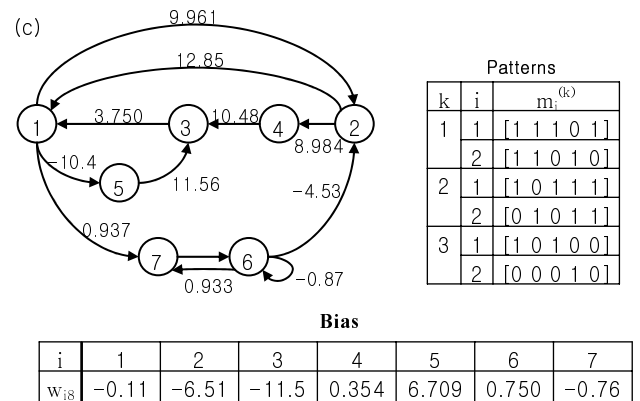
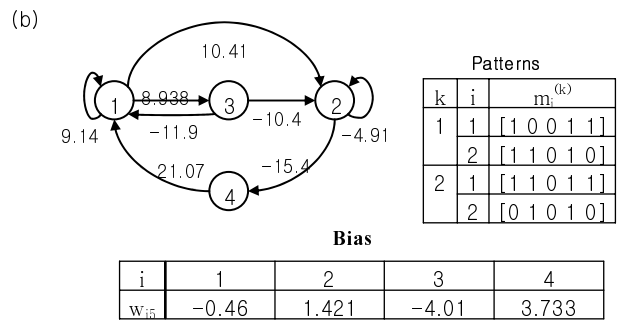
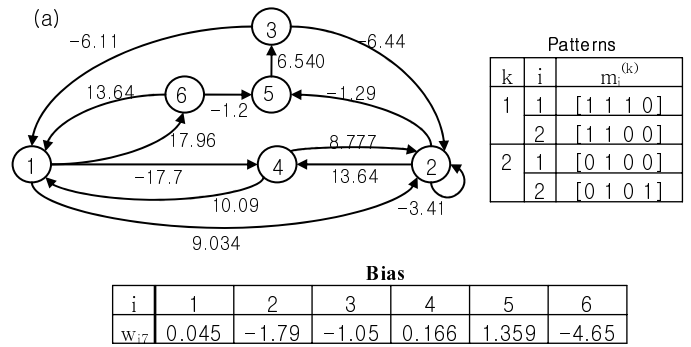


그림 5 패턴들이 성공적으로 연관되어 저장된 RNN들. 뉴런 1과 2가 단말뉴런이다. (a) $T=4, K=2$ 인 패턴 (b) $T=5, K=2$ 인 패턴 (c) $T=5, K=3$ 인 패턴

일으키고, 이 반복적인 발화에 네트워크가 적응하는 방향으로 뉴런들 간의 연결 weight가 변화하게 된다. 그런 다음에는 입력되었던 두 이진패턴 중에서 하나만 들어와도 이미 적응되어 있는 네트워크는 두 개의 패턴이 함께 입력된 것처럼 특정 패턴으로 발화하게 되고 결국 다른 쪽 이진패턴 역시 출력된다. 이러 기억과 조화가 다수의 이진패턴 쌍에 대해서도 가능해야 한다.

이와 같은 네트워크의 weight변화를 통한 적응이 가능한지 알아보기 위해서, 적응의 결과라고 할 수 있는 “패턴 쌍들을 기억한 네트워크”가 존재하는지를 진화연산을 통해 알아보았다. 네트워크를 진화시키는 기법 중 하나인 GNARL 알고리즘을 사용하여 탐색해본 결과, 무작위로 생성된 2쌍 이상의 이진 패턴을

성공적으로 기억하는 네트워크들을 찾아 낼 수 있었다. 그러나, 2쌍의 이진 패턴을 기억하는 네트워크는 대부분 찾아냈음에도 불구하고, 3쌍 이상을 기억하는 네트워크에 대한 탐색은 소수의 네트워크만을 발견했다. 이는 아마도 실험에서 구현한 진화연산 프로그램의 탐색능력이 충분치 못하기 때문으로 짐작되지만, 결론적으로 네트워크 존재의 가능성을 보였을 뿐, 현재의 모델로 임의의 K개의 정보 쌍을 기억하는 네트워크가 존재한다고 확실하게 이야기 하는 것은 좀 이르다는 것이다. 하지만, 이 실험에서 고무적인 점은 바로 뉴런을 가장 단순하게 모델링한 perceptron, 그 중에서도 더욱 간단한 step function을 사용한 perceptron으로 구성된 네트워크만으로도 이 정도의 결과를 보여주었다는 것이다. 앞으로의 연구에서는 탐색능력이 향상된 프로그램의 구현과 더불어 실제 뉴런의 특성을 perceptron에 추가함으로써 연관기억을 구현하는데 가장 적합한 모델을 찾아낼 수 있을 것이라 생각한다.

7. Reference

- [1] Kandel, E., Schwartz, J. & Jessel, T. Principles of Neural Science, Elsevier, New York, 1991.
- [2] Bear MF, Connors BW, & Paradiso MA, Neuroscience: Exploring the Brain. Baltimore: Williams & Wilkins, 2001.
- [3] Hopfield J. J., Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, 79, 2554-2558, 1982
- [4] Molter C., Salihoglu U., Bersini H., The Road to Chaos by Time-Asymmetric Hebbian Learning in Recurrent Neural Networks, Neural Computation, MIT Press, 2006.
- [5] 문병로, 유전알고리즘, 두양사, 2003.
- [6] Angeline P. J., Sauters G. M., and Pollack J. B., An evolutionary algorithm that constructs recurrent neural networks, IEEE Trans. Neural Networks, vol. 5, pp. 54-65, Jan. 1994.