'96 대한전기학회 하계학술대회 논문집 (1996. 7. 22~24)

유전자 알고리즘을 이용한 연상메모리의 설계

신누리다슬 , 이종호
인하대학교 전기공학과

Design for Associative Memory Using Genetic Algorithm

Nu-lee-da-sle Shin , Chong-Ho Lee
Dept. of Electrical Engineering, INHA Univ.

Abstract

Hopfield's suggestion of a neural network model for associative memory aroused the interest of many scientists and led to efforts of mathematical analyses. But the Hopfield Network has several disadvantages such as spurious states and capacity limitation. In that sense many scientists and engineers are trying to use a new optimization algorithm called genetic algorithm. But it is hard to use this algorithm in Hopfileld Network because of the fixed architecture. In this paper we introduce another method to determine the weigth of Hopfield type network using Genetic Algorithm.

1. 서론

1982 년 J. Hopfield[1]에 의하여 제안되어진 연상메모리(Associative Memory) 원리는 많은 과학자들에 의해 수학적으로 또는 공학적으로 분석하고 이용하려는 노력이 아기되고 있으나 Hopfield network은 그 구성상 의사극소상태(Spurious state)와 $P_{max} = N/4\ln N$ (Pattern을 99% 재생시)[4]으로 저장 용량의 제한을 갖게 된다. 이러한 단점을 극복하기 위하여 Optimizing Algorithm 의 한 종류인 Genetic Algorithm[2]을 Hofield network 에 적용하여 더 낳은 결과를 얻으려는 노력이 이루어 지고 있으나 다른 Neural Network 과는 달리 Fixed 된 Network 구조로 인한 Algorithm 적용의 많은 어려움이 있다.

본 연구에서는 유전 알고리즘(Genetic Algorithm)을 Hopfield network의 Weight 결정에 적용하는 방법을 제시하고 저장 및 재현의 결과를 보였다.

2. 안정점 과 수렴성

Recurrent Autoassociative Memory 에서 State가 Asynchronous Update 하는 동안 $E(v^{k+1}) \le E(v^k)$의 특성을 만족한다면 그 State는 안정한 점을 향해 간다고[5] 할 수 있으며 $E(v^{k+1}) = E(v^k)$: K > K0 이면 Network이 Fixed Point 또는 안정점 v^k에 도달 했다고 할 수 있다. 이러한 조건을 만족시키는 Weight Matrix는 원하는 Pattern의 Energy를 가장 안정하게 하는 Attractor 라고 할 수 있고 이러한 Weight Matrix를 Network에 적용 시키면 기억 시키고자 하는 Pattern을 저장하는 Network이 될 것이다.

이러한 조건을 만족하는 결과의 비교를 위해서 GA 에 의한 연상 기억 오차를 Hopield 학습에 의한 기억 오차와의 비교 하였다. 그림 1은 전체 흐름도이다.

2.1 Initial Population

G.A.의 Initial Population을 생성하기 위해서 Hopfield 학습에 의한 각 Weight 소자가 가질 수 있는 최대값과 최소값 즉 - Number of Pattern ~ Number of Pattern 까지의 값을 Random 하게 생성하여 Initial Population을 생성한다. 본 실험에서는 1000 개의 Initial Population으로 실험하였다.

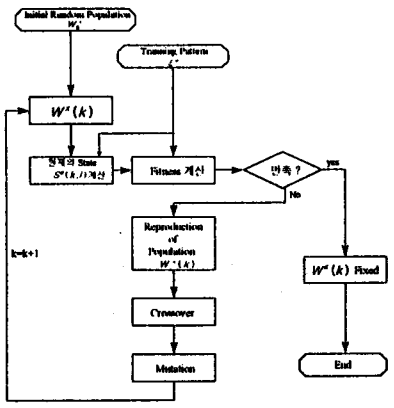

그림 1 G.A 에 의한 Weight 산출

2.2 Network

Hopfield 형의 순환 신경망으로서 실험의 편이를 위해 5 개의 Neuron으로 Network을 구성 하고 이 수를 Chromosome의 개수로 한다.

2.3 Fitness

본 실험에서는 기존의 Hopfield Network의 State Update 방식인 Asynchronous State Update[4] 를 사용하여 유전자 알고리즘에 사용될 Fitness 값을 G.A algorithm에 의해 매회 수정된 Population에 대하여 측정 하였다. 다음식들은 State Update의 식이다.

(1) Net값 계산

$$U_i(l+1) = \sum_j W_{ij} S_j(l) + \Theta_i \qquad (1)$$

N: 총 Neuron의 개수,

$U_i(l)$: 시간 l 에서의 i 번째 Neuron의 총 Input

$S_i(k)$: 시간 l에서의 i 번째 Neuron의 값

(2) New State

$$S_i(l+1) = \begin{cases} 1 & \text{If } U_i(l+1) > 0 \\ -1 & \text{If } U_i(l+1) > 0 \\ S_i(l) & \text{If } U_i(l+1) = 0 \end{cases} \qquad (2)$$

앞에서 언급한 Fixed point 개념에 의해서 저장 되어진 Pattern은 State를 반복해서 update하더라도 그 State가 바뀌지 않을 것이므로 Fitness 함수로서 학습 Pattern과 현재의 State와의 내적으로 정의 하였다.

$$F^x(k) = \sum_{\mu=1}^{p}\sum_{j=1}^{n} \xi_j^\mu g\left(\sum_i W_{ij}^x(k) S_i^\mu\right) \qquad (3)$$

2.4 Reproduction

본 논문에서의 음의 Fitness 값은 의미가 없기 때문에 0에서 1로 Normalize 하였다. 본 논문에서는 미세한 Fitness의 변화에 더 잘 반응 하도록 단순하게 Fitness 값에 따라 Reproduction을 하는 일반 Reproduction 방법이 가지는 단점 (큰 Fitness 값을 가진 Population이 생기더라도 다음 population 생성시에 Fitness 값이 큰 Population의 Reproduction의 개수가 많지 않을 수 있다) 을 보완한 일반적인 Reproduction 방법이 아닌 개선된 Exponential Reproduction[6] 방법을 사용했다.

2. 5 Crossover와 Mutation

임의의 점에서 Crossover Point를 잡아서 이를 Swap하는 방식의 Crossover와 현 세대에서 다음 세대로 반복하기 전에 현재 존재하는 Population중에서 일정 미세 확률만큼의 변이를 가지도록 Mutation을 취하였다. 본 실험에서는 0.015의 변이 확률을 사용하였다.

2.5 Hopfield Weight

실험 결과의 비교 Test를 위해서 Hebbian Learning Rule을 사용해서 Hopfield Weight를 생성한다.

$$W_{ij} = \sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu \qquad (4)$$

여기서 ξ_i^μ 는 μ번째 pattern의 i번째 Neuron, P 는 Pattern의 개수이다.

3. Simulation 및 결과

그림 2와 그림 3은 Hamming Distance가 3인 Pattern 2개와 4개를 가지고 0.015의 변이(Mutation) 확률로 370번과 470번의 Iteration으로 Generation 시킨 결과이다. 여기서의 Fitness 값은 각 Iteration당 발생된 Population에 의해 계산된 개별 Fitness $F^x(k)$들의 평균적인 값으로 각 세대가 지고 있는 평균적 추이를 반영한 것이다. 표 1은 그림 2과 그림 3의 Fitness 값을 가지는 Weight와 Hopfield의 Weight로 Recall한 결과를 비교한 것이다.

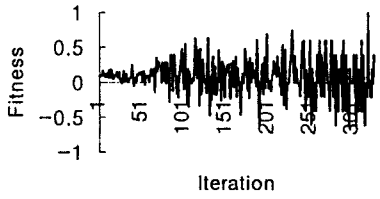

그림 2. HD=3 인 2 개 Pattern 을 기억시킨 경우의 Simulation 결과

학습방법	2 개의 학습 Pattern 으로 Recall 했을경우	4 개의 학습 Pattern 으로 Recall 했을경우
Hopfield 학습	0 Bit Error	0 Bit Error
Genetic Algorithm	0 Bit Error	1 Bit Error

표 1. Hopfield Network의 Weight와 G.A.로 만든 Weight의 Recall Error 비교

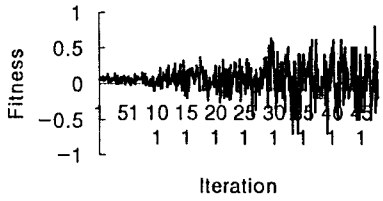

그림 3. HD=3 인 4 개 Pattern 을 기억시킨 경우의 Simulation 결과

4.결론 및 고찰

그림에서와 같이 각 세대의 Population이 가지는 평균 Fitness 값은 각 String이 가지는 Fitness 값의 분포가 몇 개의 부류로 나뉘고 이 String의 Crossover에 따른 진동에 따라 일정한 값으로 수렴하지 않음을 보여주고 있다. 수렴성 및 안정성의 원리에 의해서 만들어진 Fitness Function으로 생성된 최종 Weight Matrix는 표 1에서 보

는 바와 같이 학습된 Pattern 으로 수렴하는 성질을 보여준다. 또한 기억시킬 Pattern 의 수가 많아지면 Fitness 가 1 로 수렴하지 않을 수 있지만 Fitness 값이 0.85 정도로 수렴해도 Hopfield Model 이 보여주는 Recall 특성과 비슷한 특성을 보여주고 있다. 여기서 수렴의 경우는 단일종으로 모든 Population 이 형성된 경우이며(이상적이건 아니건 간에) 또한 Mutation 이 변종에 미치는 영향이 무시할 만큼 작다는 것을 의미한다. 본 논문에서 제시한 Weight Matrix 의 결정 방법은 Initial Population 의 범위가 Pattern 수에 의해 결정되었지만, Initial Population 을 Hopfield Weight 에 Dilution Process 를 거쳐서 만든 Population 으로 실험하면 Hopfield Network 의 Weight 로 수렴함도 알 수 있었다. 이러한 Weight 결정 방법은 Hopfield Network 뿐만 아니라 BAM Type Network[3] 그리고 Feed forward network 에서의 응용도 가능 하리라 생각된다. 본 논문에서는 Genetic Algorithm 에 의해서 결정된 Weight Matrix 가 가질 수 있는 Local minima 에 대해서는 고려 하지 않았으나 Genetic Algorithm 을 이용한 Local Minima 의 제거 및 방지를 위한 방법과 연상 기억 능력의 증대에 관한 연구가 진행 중에 있다.

참고 문헌.

[1] J.J Hopfield, *Neural Networks and Physical System with Emergent Collective Computational Abilities.* Proceeding of the National Academy of Science, April 1982.

[2] D. Goldberg, *Genetic Algorithm in Search, Optimization and Machine Learning.* Addison-Wesly , 1989

[3] Bart Kosko, *Neural Networks and Fuzzy Systems* Prentice Hall , 1992.

[4] John Hertz et.al., *Introduction to the Theory of Neural Computation.* Addison-Wesley, 1991

[5] Jacek M Zurada *Introduction to Artificial Neural Systems*, West info Access, 1992

[6] S.W. Kang *Digital Halftoning Techniques Using Iterative Computational Methods*, INHA UNIV. 1996 . 8