

GA를 이용한 신경망의 가중치 최적화

Neural Network Weight Optimization using the GA

문상우*, 공성곤

송실대학교 전기공학과

Sang-Woo Moon* and Seong-Gon Kong

Department of Electrical Engineering

Soongsil University

요 약

신경망은 복잡하게 나타나는 비선형성을 가지는 실제의 다양한 문제들에 적용이 가능할 뿐만 아니라, 정보들이 가중치에 분산되어 저장됨으로서 강인성을 가지고 있다. 그러나 전방향 다층 신경망 구조를 학습할 수 있는 역전파 알고리즘은 초기 가중치의 영향에 의하여 학습된 결과가 지역 최소점에 빠지기 쉬운 경향이 있다. 본 논문에서는 이러한 문제점을 해결하기 위한 한가지 방법으로서 유전자 알고리즘을 이용하여 전방향 다층 신경망의 가중치를 학습하여, 지역 최소점에 빠지지 않고 학습이 이루어짐을 보인다.

1. 서 론

역전파 학습 알고리즘은 패턴인식, 함수 근사화, 적응 제어 등의 매우 복잡한 문제를 성공적으로 해결할 수 있는 전방향 다층 신경망을 학습시킬 수 있는 대표적인 알고리즘이다. 하지만 오차 표면에 대한 기울기를 이용하여 오차가 감소하는 쪽으로만 학습하는 경사하강법에 근거하기 때문에 역전파 학습 알고리즘 역시 경사하강법의 근본적인 약점인 초기화에 의하여 발생하는 지역 최소점에서의 수렴 상태를 벗어나지 못한다는 문제점을 내포하고 있다.[1][2] 이러한 문제점을 해결하기 위하여 전방향 다층 신경망을 학습시키기 위한 다양한 방법들이 연구되어지고 있다.[3]

최근 대두되기 시작한 유전자 알고리즘은 유전원리에 근거하여 해가 존재할 수 있는 구간을 염색체로 코딩한 후 유전자의 교배와 돌연변이등의 개념을 응용한 유전 연산자를 이용함으로써 전역탐색과 지역탐색의 특성을 가지는 최적화 방법으로서, 기존의 경사하강법이 가지는 한계점인 초기화에 의해 발생하는 지역 최소점에서의 국부적 수렴상태를 극복할 수 있는 알고리즘이다.[4][5][6][7]

본 논문에서는 전방향 다층 신경망을 학습함에 있어서 초기 가중치의 영향에 관계없이 지역 최소점에 수렴하지 않고 전역 최소점에 이를 수 있도록 하기 위하여 신경망의 학습 알고리즘으로서 유전자 알고리즘을 사용하였고, 이의 성능과 특성을 역전파 학습 알고리즘을 사

용한 경우와 비교하여 보았다.

2. 신경망 가중치의 최적화를 위한 유전자 알고리즘의 적용

2.1. 신경망과 유전자 알고리즘의 결합

전방향 다층 신경망과 유전자 알고리즘을 결합하기 위하여 그림 1과 같은 구조로 신경망과 유전자 알고리즘의 구조를 혼합하였다.

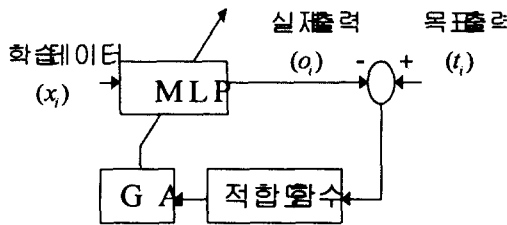


그림 1. 유전자 알고리즘을 이용한 전방향 다층 신경망의 가중치 학습 흐름도

실제 학습시에는 유전자 알고리즘이 학습을 주도하며 신경망은 단지 유전자 알고리즘에서 사용되어지는 적합도를 얻기 위한 수단으로서만 사용되어진다. 신경망의 출력을 사용하여 계산되는 적합도를 이용하여 유전자 알고리즘은 가중치가 주어진 정밀도와 탐색범위 내에서 코딩되어 얻어진 개체들이 최고 적합도를 가지도록 학습된다.

2.2. 유전자 알고리즘 모델

다층 신경망의 가중치 학습을 위하여 유전자 알고리즘을 사용하기 위해서는 사용되는 유전자 알고리즘의 가중치 코딩과 적합도 함수의 설정방법에 관하여 고려해야 한다.

먼저 전방향 다층 신경망의 구조를 결정하고, 그후 결정된 구조에서 사용되어지는 가중치들의 탐색 구간과 정밀도를 설정해야 한다. 이렇게 설정되어진 가중치들을 최하위 비트로부터 나열하여 한 개의 2진 개체를 구성하였다.

본 논문에서는 각각의 가중치의 탐색 구간과 정밀도를 모두 같게 설정하였다. 개체생성을 위한 가중치의 코딩은 그림 2와 같이 입력층과 중간층 사이의 가중치중 중간층 뉴런 1개에 대한 가중치를 입력층 뉴런의 순서에 의하여 최하위 비트로부터 나열하고, 중간층과 출력층 사이의 가중치를 같은 방법으로 이어서 나열하는 방법을 사용하였다.

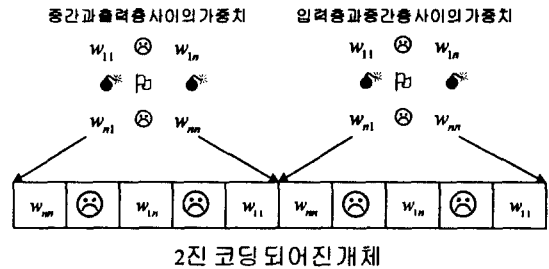


그림 2. 가중치 코딩

이 경우의 한 개의 개체가 코딩되어지는 비트 수는 다음과 같이 계산하였다.

$$l = n_w \log_2 [(SR_{\max} - SR_{\min}) \cdot 10^n + ((SR_{\max} - SR_{\min}) \cdot 10^n) \% 2] \quad (1)$$

l 은 한 개의 개체가 코딩되어지는 비트수, n_w 은 신경망의 가중치 개수, n 은 허용 유효 자리수, SR_{\min} 은 최소 탐색구간, SR_{\max} 은 최대 탐색구간, %는 나머지를 나타내는 modulo 연산자를 나타낸다.

코딩되어진 개체의 해석은 다음과 같이 이루어진다.

$$w_i = SR_{\min_i} + BS_i \cdot \frac{SR_{\max_i} - SR_{\min_i}}{2^i - 1} \quad (2)$$

w_i 는 개체 최하위 비트로부터 i 번째로 코딩되어진 가중치의 실수값, BS_i 는 i 번째로 코딩

되어진 2진 가중치의 크기를 10진수로 나타낸 것이다.

신경망의 가중치를 유전자 알고리즘을 이용하여 학습하기 위한 적합도 함수의 설정은 자승오차를 기본으로 하였다. 유전자 알고리즘의 탐색에 의하여 선택되어진 신경망의 가중치에 의하여 각각의 학습 데이터들의 자승오차를 계산하고, 얻어진 자승오차를 모두 더한 값의 제곱근을 유전자 알고리즘의 개체 1개가 신경망에 가중치로서 사용될 경우에 발생되어지는 오차로 정의하였다.

$$E = \sqrt{\sum_{i=1}^n (t_i - o_i)^2} \quad (3)$$

E 는 각 학습데이터 들의 자승오차들의 합, t_i 는 i 번째 학습 데이터의 목표 출력, o_i 는 사용되어진 가중치의 함수로서 i 번째 학습 데이터와 현재 선택된 개체로부터 해석되어진 가중치로부터 얻어지는 실제 출력을 나타낸다.

유전자 알고리즘은 적합도의 최대치로 모든 개체들을 수렴시키며 탐색하는 방법이므로 이를 반영한 적합도 함수의 식을 다음과 같이 설정하였다.

$$F = E_{wc} - E \quad (4)$$

F 는 적합도 함수, E_{wc} 는 나올 수 있는 최대 에러이다.

출력층 뉴런에 사용한 활성화함수는 시그모이드 함수를 사용하였으므로, 출력층의 뉴런 1개가 나타낼 수 있는 최대 오차 1이고, 하나의 학습 데이터에 의한 최대 오차는 출력층 뉴런의 개수로서 정의할 수 있다. 전체 학습 데이터에 대한 최대 에러 E_{wc} 는 식 (5)와 같이 정의하였다.

$$E_{wc} = \sqrt{pq} \quad (5)$$

p 은 학습 데이터의 수, q 은 사용되어진 신경망의 출력 뉴런의 수를 표현하였다.

3. 시뮬레이션

XOR과 3비트 패리티 문제에 대하여 각각 역전파와 유전자 알고리즘을 신경망의 가중치 조절에 적용하여 보았다. 학습에 사용된 신경망의 구조는 전방향 다층 신경망이며 중간층은 1개만을 사용하였다. 모든 뉴런의 활성화함수로는 시그모이드 함수를 사용하였고, 신경망의 목표 출력으로는 0과 1을 사용하였다.

3.1. XOR 문제

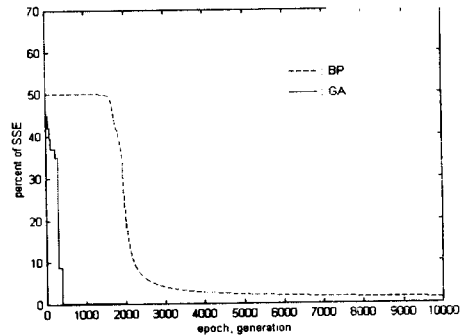


그림 3. XOR 문제의 오차

XOR문제에 대하여 역전파 알고리즘과 유전자 알고리즘을 전방향 다층 신경망에 적용한 결과는 그림 3과 같다.

신경망의 구조로는 2개의 입력층 뉴런, 2개의 중간층 뉴런, 1개의 출력층 뉴런을 사용하였다. 역전파 알고리즘의 경우 학습상수는 0.2, 모멘텀은 0.8을 사용하였고, 유전자 알고리즘의 경우는 개체수 10개, 탐색구간 -20~20(오차 허용범위는 소수점 이하 4자리까지), 돌연변이 확률을 1%, 교배확률을 25%로 선택하였다. 이 경우 유전자 한 개의 길이는 171비트로 설정되었다.

그림 3에서 보이는 바와 같이 역전파 알고리즘으로 학습하였을 경우에는 오차가 1.2% 이상

줄어들지 않는 것을 볼 수 있으나, 유전자 알고리즘의 경우 높은 오차 허용범위 하에서 수세대 만에 오차가 거의 0으로 수렴함을 알 수 있다.

3.2. 3비트 패리티 문제

N 비트 패리티 문제는 신경망 학습 알고리즘의 성능평가를 위하여 자주 사용되는 문제로 입력은 N 차원 공간상에서 2^N 의 패턴들로 구성되며, 각 패턴은 N 비트의 이진 벡터이다. 출력값은 입력된 패턴에서 1의 개수가 짝수일 경우에는 0, 홀수일 경우에는 1을 가지게 된다. 패리티 문제에는 중간층이 1개이고 홀수 비트 패리티일 경우는 중간층 뉴런의 개수가 $(N+1)/2$, 짝수 비트 패리티일 경우에는 $N/2+1$ 개인 전방향 다층 신경망 구조를 사용할 경우 해결이 가능하다는 사실이 알려져 있다.[8]

3비트 패리티의 경우 1개 이상의 중간층에 2개 이상의 중간층 뉴런을 가지는 신경망으로 해결할 수 있다. 하지만 그림 4와 같이 초기치의 영향에 의하여 중간층 뉴런이 2개인 경우 30번의 반복 실험중 12번의 수렴하지 못하는 경우가 발생한다.

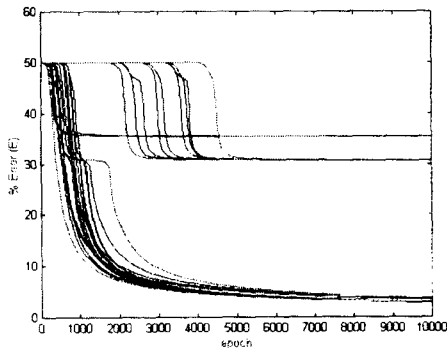


그림 4. 초기치의 영향에 의한 BP의 수렴정도

3비트 패리티 문제를 역전파 알고리즘과 유

전자 알고리즘에 적용한 결과는 그림5와 같다.

신경망의 구조로는 3개의 입력층 뉴런, 2개의 중간층 뉴런, 1개의 출력층 뉴런을 사용하였다. 역전파 알고리즘의 경우 학습상수는 0.2, 모멘텀은 0.8개를 사용하였고, 유전자 알고리즘의 경우는 개체수 500개, 탐색구간 -10~10(오차 허용범위는 소수점 이하 4자리까지), 돌연변이 확률을 1%, 교배확률을 25%로 선택하였다. 이 경우 유전자 한 개의 길이는 1988비트로 설정되었다.

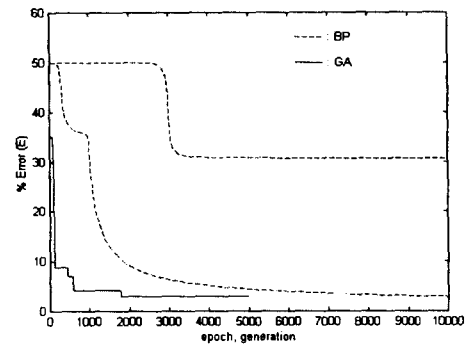


그림 5. 3비트 패리티 문제의 오차 (개체수 500인 경우)

3비트 패리티 문제의 경우 역전파 알고리즘은 초기치의 영향으로 수렴하지 않을 수 있으나 유전자 알고리즘은 오차가 3%정도로 잘 수렴함을 알 수 있다. 그러나 XOR문제의 경우와 비교해 보면, 입력층과 중간층 뉴런수의 증가로 인한 가중치 수는 1.2배 증가한 반면 개체수는 50배가 증가한 것을 알 수 있다. 이는 유전자의 길이가 171비트에서 198비트로 늘어났기 때문이다.

돌연변이 확률을 10%로 높이고, 개체수를 200개로 줄였을 경우 그림 6과 같은 결과를 얻을 수 있었다. 개체수를 줄였을 경우 최대 적합도를 가지는 개체의 오차가 수렴하지 않음을 알 수 있다.

결국 유전자 알고리즘이 비교적 적은 세대에 해를 찾아내기 위해서는, 개체의 길이를 결정하게 되는 탐색구간의 범위와 탐색의 정밀도,

적합도 함수를 나타내는 신경망의 구조, 복잡도를 나타내는 학습 데이터에 따라서 필요한 개체수를 결정해야 하며, 이들 세가지 요인들의 증가는 더욱 많은 개체수의 증가를 필요로 하게 됨을 알 수 있다.

4. 결론

비교적 간략한 두가지 문제에 대하여, 전방향 다층 신경망을 유전자 알고리즘으로 학습함으로써 역전파 알고리즘의 약점인 지역 최소점에서 벗어나지 못하는 문제를 쉽게 해결할 수 있다는 것을 확인하였다. 또한 역전파 알고리즘이 해를 찾아가는 데에 긴 반복을 필요로 하지만 유전자 알고리즘은 비교적 적은 세대 이내에 해를 찾을 수 있었다.

하지만 탐색구간의 범위와 탐색의 정밀도, 적합도 함수를 나타내는 신경망의 구조, 학습 데이터의 복잡도들이 증가함에 따라서 더욱 많은 개체수의 증가를 필요로 하였다. 이는 많은 계산량을 유발시켰고 복잡한 문제에 대한 실제에의 적용을 어렵게 만드는 요인중의 하나로 작용하였다.

그러므로, 다음 논문에서는 신경망과 유전자 알고리즘을 하드웨어에 적용시킬 수 있는 구조로 변경시키고, 이들을 이용한 하드웨어를 구현하고자 한다. 그리고 이를 더욱 복잡한 문제에 적용하고자 한다.

참고 문헌

[1] H. Guo and S. B. Gelfand, "Analysis of Gradient Descent Learning Algorithms for Multilayer Feedforward Neural Networks," *IEEE Trans. Circuits and Systems*, Vol. 38, No. 8, 1991.

[2] M. Gori and A. Tesi, "On the Problem of Local Minima in Backpropagation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 1, 1992.

[3] D. E. Rumelhart, G. E. Hinton, and R. J.

Williams, "Learning Internal Representation by Error Propagation," *Parallel Distributed Processing*, Vol. 1, 1986.

[4] W. M. Jenkins, "A Neural Network Trained by Genetic Algorithm," *Procs. of the Advances in Computational Structures Technology*, pp. 77-84, 1996.

[5] D. J. Montana and L. Davis, "Training Feedforward Neural Networks Using Genetic Algorithms," *Proc. of the IJCAI-89*, pp. 762-767, 1989.

[6] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 2nd Edition, Springer-Verlag, 1992

[7] G. Winter, J. Periauz, M. Galan, and P. Cuesta, *Genetic Algorithms in Engineering and Computer Science*, John Wiley & Sons, 1995.

[8] R. Setiono, "A Neural Network Construction Algorithm which Maximizes the Likelihood Function," *Connection Science*, Vol. 7, No. 2, 1995.