

오류역전파 알고리즘의 Local maxima를 탈출하기 위한 방법에 관한 연구

서원택⁰ 조범준

조선대학교

wontagi@ai.chosun.ac.kr bjcho@mail.chosun.ac.kr

The Study on the Method which escape from Local maxima of Error- Backpropagation Algorithm

Won-Taek Seo⁰ Beom-Joon Cho
Chosun University

요약

본 논문에서 소개하는 알고리즘은 은닉층의 뉴런의 수를 학습하는 동안 동적으로 변화시켜 역전파 알고리즘의 단점인 Local maxima를 탈출하고 또한 은닉층의 뉴런의 수를 결정하는 과정을 없애기 위해 연구되었다. 본 알고리즘의 성능을 평가하기 위해 두 가지 실험에 적용하였는데 첫번째는 Exclusive-OR 문제이고 두번째는 7×8 한글 자음과 모음의 폰트 학습에 적용하였다. 이 실험의 결과로 네트워크가 local maxima에 빠져드는 확률이 줄어드는 것을 알수 있었고 학습속도 또한 일반적인 역전파 알고리즘보다 빠른것으로 증명되었다.

1. 서 론

오류역전파 알고리즘이라고도 불리는 역전파(BP) 알고리즘은 계층적 구조를 가지는 신경망에서 학습 알고리즘으로 효과적으로 사용되고 있다[1]. 그러나 이 역전파 알고리즘은 학습시에 발생하는 문제점이 있는데 Local maxima에 빠져서 학습이 이루어지지 않는 경우가 생기는 것이다.[2] 또한 최적의 은닉층의 뉴런의 개수를 이론적으로 결정할수 없기 때문에 많은 시행착오를 거쳐서 가장 적절한 수를 선택하는 것이 일반적이다. 이러한 문제를 해결하기 위해서 본 논문에서는 역전파 알고리즘의 학습 과정에서 은닉층의 뉴런의 개수를 동적으로 변화시켜 보았는데 Local maxima에 빠졌다고 판단되면 은닉층 뉴런의 개수를 증가시켜 Local maxima를 탈출 할 수 있게 유도하고 또한 그 과정에서 최적의 은닉층의 뉴런의 개수를 결정할 수 있도록 구성해보았다. 이 알고리즘의 성능을 검사하기 위해 두가지 경우의 문제에 적용시켜보았는데, 첫번째로 Exclusive-OR 문제에 적용하였고 두번째로 7×8 의 한글 자모음 폰트의 학습에 적용시켜 보았다.

2장에서는 역전파 알고리즘에 대해서 간략히 소개하고 3장에서는 본 알고리즘을 실제로 XOR 연산과 한글폰트 학습에 적용하여 제안된 알고리즘과 일반적인 역전파 알고리즘과 비교해서 성능을 평가하였고 4장에서는 결론 및 고찰로 맺었다.

2. 알고리즘

2.1 일반적인 역전파 알고리즘

역전파 알고리즘은 입력층과 은닉층의 연결강도를 변경

시키는 신호로 델타 신호를 사용하는데 델타신호는 연결 강도와 출력층에서의 목표치와 출력치와의 오차인 E 와의 관계로부터 구할 수 있다.[3]

$$\Delta w_{ij} = -\alpha \nabla E = -\alpha \frac{\partial E}{\partial w_{ij}} \quad (1)$$

$$\delta_j = -\frac{\partial E}{\partial (NET_j)} \quad (2)$$

여기서 Δw_{ij} 는 i번쨰 입력과 j번쨰 출력사이의 연결 강도 변화량이고 α 는 양수이다. 전체 오차 E 는 다음과 같이 구할 수 있다.

$$E = E + \frac{1}{2} \|d - y\|^2 \quad (3)$$

여기에서 d 는 목표치이고 y 는 출력치이다.

다층 신경망에서 입력층과 은닉층에 전파되는 오차신호는 다음과 같다.

$$\delta_j = \frac{1}{2} (d - y)(1 - y^2) \quad (4)$$

$$\delta_i = \frac{1}{2} (1 - z^2) \sum_{j=1}^m \delta_j w_{ij} \quad (5)$$

역전파 알고리즘은 Steepest descent 알고리즘을 기초로 하고 있기 때문에 Local maxima에서 탈출하지 못하는 단점이 있다. Local maxima란 연결강도의 공간에서 오차와의 관계에서 표현할 수 있는데 그림 1에서 보는 바와 같이 연결강도가 변경되다가 W_2, W_3 에서 더 이상 변경할수 없게 되는 상태이다. 본 논문에서 이러한 Local maxima를 효과적으로 탈출하기 위해 제안하였다.

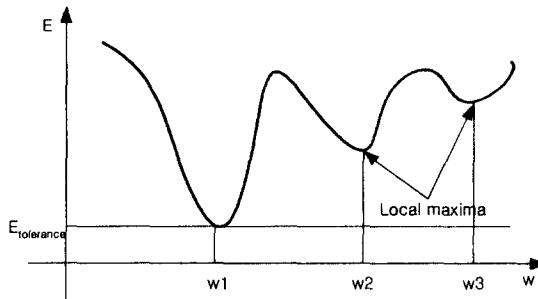


그림 1. 연결강도 공간에서의 Local maxima

2.2. 제안된 알고리즘

제안된 알고리즘의 핵심요지는 네트워크가 Local maxima에 빠졌다고 판단되었을 경우 은닉층의 뉴런의 개수를 추가해서 연결강도를 더 개선할 수 있는 조건을 만들어 준다.

그러면 언제 어떻게 뉴런을 추가할 것인가를 살펴보겠다. 네트워크가 Local maxima에 빠졌다는 것을 정확히 판단해내기는 어렵다. 여기에서 오류 E를 가지고 판단하는데 매 100회의 연결강도를 조정한 후 E를 검사하여 만약에 E의 차이가 1% 미만이라면 Local maxima에 빠진 것으로 판단하고 은닉층의 유닛을 추가한다. 만약에 E가 1% 이상 감소하였다면 그 다음 100회를 시행한 후 다시 검사한다. 유닛을 추가하는 방법으로는 두 가지를 사용하였는데 한 가지는 연결강도를 0으로 설정하는 방법과 또 한 가지는 1과 -1 사이의 랜덤한 값으로 설정하는 것이다. 이렇게 하여 최소 오차의 허용한도에 도달하면 네트워크의 학습이 성공한 것으로 판단한다. 그런데 이렇게 은닉층 뉴런의 개수를 증가하다 보면 네트워크의 크기가 커져 버리는 단점이 있다. 그러한 단점을 보완하기 위해서 한 번 네트워크가 수렴하게 되면 이번에는 다시 은닉층의 뉴런을 하나씩 제거하면서 다시 학습을 하여 수렴하는지를 판단한다. 마지막으로 수렴하지 못할 때까지 학습하여 그 전단계의 네트워크를 최종적으로 결정하게 된다.

아래의 그림 2와 3으로 exclusive-OR 문제의 학습과정에 대해서 나타내었다. 이 네트워크는 2개의 입력과 1개의 출력으로 구성하였다.

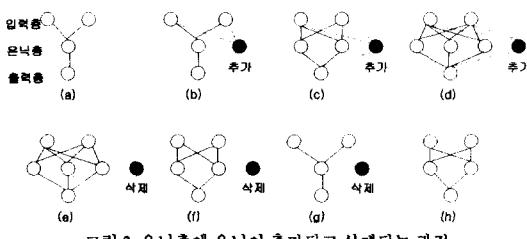


그림 2. 은닉층에 유닛이 추가되고 삭제되는 과정

그림 2은 학습과정 동안 네트워크의 구조의 변화를 나타내고 그림 3은 학습과정동안 학습횟수에 따른 E의 변화를 나타내었다.

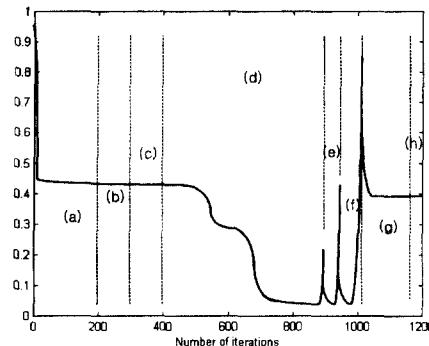


그림 3. 네트워크 구조 변화에 따른 E의 변화

초기 연결강도는 랜덤하게 정하였는데 처음 50번의 학습이 진행되면서 E가 0.5 감소하였고 100번의 학습 후에는 E가 1% 이상 감소하였다. 그래서 은닉층의 뉴런이 추가되지 않았다. 그 다음 100번의 학습 후에는 E가 1% 미만으로 감소하였고 그래서 그림 2(b)와 같은 모양으로 유닛이 한 개 추가되었다. 채워진 동그라미가 새로 추가된 유닛을 표현하고 점선으로 표시된 것은 추가된 유닛에 의해 연결된 연결강도이다. 그 다음 100번의 학습이 있었으나 E가 1% 미만으로 감소하고 E의 최소치에 도달하지 못했기 때문에 그림 2(c)와 같이 유닛이 추가되었다. 400번의 연결강도가 조정 후 또 하나의 은닉층 뉴런이 추가되었고 그림 2(d)에 표시하였다. 그후 E는 완만하게 감소하다가 810번의 학습 후에 수렴하였다. 한번 네트워크가 수렴하고 난 후 다시 유닛의 삭제 과정에 들어가게 되어서 그림 2(e)와 같이 뉴런을 삭제하였다. 처음에는 E가 증가하였지만 40번의 학습 후에는 수렴하여 다시 그림 2(f)와 같이 또 하나의 뉴런을 삭제하였다. 여러 E는 0.5까지 증가하였지만 다시 수렴하였고 다시 뉴런을 제거했을 때는 끝내 수렴하지 않았다. 그래서 그림 2(h)의 모양으로 네트워크를 결정하고 연결강도는 그림 2(f)에서 성공한 연결강도로 결정하였다.

3. 실험

본 알고리즘의 실험을 위해서 두 가지의 문제를 결정하였다. 간단한 문제로는 XOR 문제이고 좀 더 복잡한 문제로 7x8 한글 자모음 폰트의 학습문제로 결정하였다. 입력층과 은닉층 사이의 활성화함수에는 다음과 같은 식을 사용하였다.

$$n = \frac{2}{1 + \exp(-2n)} + 1 \quad (6)$$

은닉층과 출력층 사이에는 항등함수(identity function)을 사용하였다.

3.1 Exclusive-OR

Exclusive-OR 학습 문제는 간단하면서도 비선형적이기 때문에 신경망을 실험하는데 자주 사용되는 문제이다

본 실험에서는 학습률을 0.1로 설정하였고 최소 허용오차 E는 0.001로 하였고 최대 학습반복 시기를 10000번으로 설정하였다. 초기의 네트워크의 구성으로는 입력이 2개, 은닉층의 뉴런이 1개, 출력층의 뉴런을 1개로 하였다. 초기의 연결강도는 -1에서 1까지의 랜덤한 수로 하였다.

첫째로, 초기의 뉴런의 수와 마지막 학습이 끝났을 때의 뉴런의 수를 비교해 보았다. 이 실험을 위해서 네트워크의 초기 뉴런의 수를 1, 2, 3, 5, 10개로 해서 각각 10번씩의 실험을 하여 평균을 구해 보았는데 결과는 그림 4와 같다.

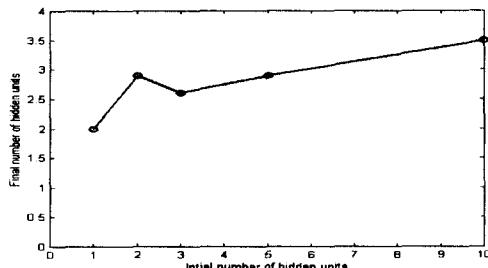


그림 4. 초기 은닉층의 뉴런수과 최종 은닉층의 뉴런수와의 관계

위의 그림에서 알 수 있듯이 작은 네트워크를 구성하려고 할 때는 은닉층의 초기 뉴런의 개수를 크게 설정하는 것이 좋다. 또한 뉴런을 감소시키는 과정에서 제거하는 순서에 관해서 실험해 보았는데 최초의 뉴런과 차례로 더해진 뉴런에 순서를 붙여서 제거해 보았다. 먼저 최초의 뉴런을 삭제하였을 때는 네트워크가 수렴하지 못하는 것을 알 수 있었다. 그리고 마지막에 추가된 뉴런부터 삭제를 해보니 네트워크가 수렴하였다. 추가된 뉴런들은 순서에 관계 없이 제거하여도 네트워크가 수렴하였으나 최초의 뉴런을 제거 시에는 수렴하지 않는 것을 확인 할 수 있었다.

3.2 7x8의 한글 자모음 폰트 학습

Exclusive-OR의 문제에 적용한 네트워크는 최대의 뉴런의 개수가 10개 정도 밖에 되지 않는 작은 네트워크였다. 그래서 본 알고리즘을 좀 더 큰 문제에 적용하고자 7x8의 한글 자모음 폰트의 학습을 선택하였다. 그림 5에 네트워크의 구성을 간략하게 나타내었다.

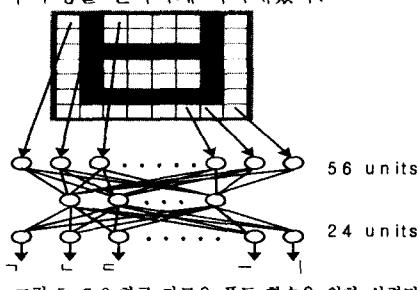
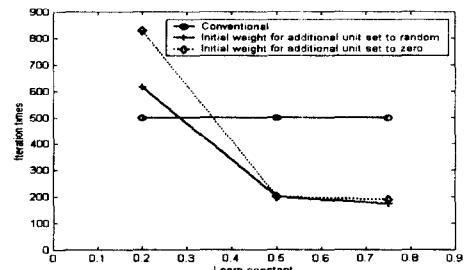


그림 5. 7x8 한글 자모음 폰트 학습을 위한 신경망

입력층에 56개의 유닛이 존재하고 출력층에는 ㄱ-ㅎ 까지의 자음과 ㅏ-ㅣ 까지의 모음을 나타낼 수 있게 24개의 뉴런으로 구성하였다. 입력층에서는 하얀색의 도트는 -1의 값으로 입력되고 검정색의 도트는 1로 입력된다.

초기의 학습률은 0.2로 설정하였고 E는 0.001로 설정하였고 최대 학습횟수는 10000번으로 설정하였다. 매 100회마다 E를 검사하여 뉴런의 추가 여부를 결정하였다.

추가되는 뉴런에 연결강도를 0으로 설정하였을 때는 대부분의 학습이 수렴하였는데 수렴되지 않은 것을 살펴보면 변화의 폭이 작아서 계속 뉴런이 추가가 되는 경우였다. 그래서 연결강도를 랜덤한 값으로 설정하였더니, E가 급격히 증가하였지만 조정이 빨리 되어지는 것을 확인할 수 있었다. 이 학습에서 최종 은닉층의 수는 대략 6~7개였다. 우리는 학습률에 따른 계산시간을 일반적인 역전파 알고리즘과 비교해보고 다음 그림과 같은 결과를 얻을 수 있었다.



학습률이 클수록 빠르게 수렴하는 효과가 있었지만 학습률이 작을 때 보다 학습이 이루어 지지 않는 경우가 더 많았다.

4. 결론 및 고찰

본 알고리즘이 Local maxima에서 효과적으로 잘 벗어 났는가 하는 것을 일반적인 알고리즘의 결과와 본 알고리즘의 결과를 비교해보고 판단할 수 있었다. 일반적인 알고리즘은 10번 중에 2~3회 정도는 수렴하지 못하였지만 본 알고리즘은 거의 모든 학습에서 수렴하였다.

본 논문에서는 은닉층의 뉴런의 개수를 변경시키는 역전파 알고리즘에 대해서 소개하였다. Local maxima에 빠졌을 때 은닉층에 새로운 유닛을 추가하고 연결강도로 랜덤한 값을 주면 Local maxima에서 탈출할 수 있었다. 본 알고리즘을 사용하면 더 이상 은닉층의 뉴런의 개수를 결정할 필요도 없게 된다. 7x8 한글 자모음 폰트 학습을 하였을 때는 일반적인 역전파 알고리즘보다 1.5배 정도 빠르게 학습하였다.

우리는 앞으로 이 알고리즘을 음성인식 분야로 확대 적용해볼 계획이다.

5. 참고문헌

- [1] Rumelhart, D. E., Hinton, G. E., & Williams, R. J.: Learning representations by error propagation. Parallel distributed processing vol. I, pp. 318~362, 1986
- [2] Robert Hecht-Nielsen: Theory of the Backpropagation Neural Network, IJCNN, Vol 1, pp. 593~605, 1989
- [3] Sejnowski, T. J. & Rosenberg, C. R.: Parallel networks that learn to pronounce English text. Complex Systems, 1, 145~168, 1987