

# Local maxima 를 해결하기 위해 개선된 오류역전파 알고리즘

서원택\*, 조범준\*\*  
\*조선대학교 컴퓨터공학과  
\*\*조선대학교 컴퓨터공학부  
e-mail : wontagi@ai.chosun.ac.kr

## Developed BackPropagation which solve the problem of Local maxima

Won-Taek Seo\*, Beom-Joon Cho\*\*  
\*Dept. of Computer Science, Chosun University  
\*\*School. of Computer Engineering, Chosun University

### 요 약

다층 신경망의 학습에 쓰이는 오류 역전파 학습은 매우 효과적이지만 학습 속도가 너무 느리고 최적의 은닉층의 뉴런의 수를 결정하는 해답은 아직 없는 실정이다. 또한 가끔은 국부 최소점(Local maxima)에 빠져 학습이 끝내 이루어지지 않는 경우가 있다. 이에 본 논문에서는 이러한 Local maxima 를 효과적으로 탈출 할수 있는 방법에 대해서 연구해 보았다. 국부 최소점은 연결강도와 전체 오차 사이의 이차원 공간에서 표현할수 있는데 본 알고리즘은 이러한 연결강도와 오차와의 관계를 인위적으로 변화시켜 결론적으로 Local maxima 를 탈출하게 하는 방법을 소개한다. 본 연구에서 사용된 방법은 네트워크 학습중에 Local maxima 에 빠졌을 때 은닉층의 뉴런의 수를 추가하여 인위적으로 연결강도 평면의 위상을 변조시킨다. 또한 은닉층의 뉴런의 수를 동적으로 변화 시키면서 최적의 뉴런의 수를 결정할 수 있게 하였다. 위 알고리즘의 성능을 평가하기 위해서 XOR 문제와 10x8 영문폰트와 숫자의 학습에 적용하여 일반적인 역전파 학습과 비교평가하였다.

### 1. 서론

오류 역전파(Error-BackPropagation) 학습 알고리즘은 다층 신경망의 학습방법으로 매우 효과적인 것으로 알려져 있다.[1] 그러나 이 방법은 Steepest decent 알고리즘을 기본으로 하고 있기 때문에 학습속도가 매우 느리고 국부 최소치(Local Maxima)에 빠질 위험성이 있어 문제시 되고 있다. 이를 위해 많은 연구 결과가 나오고 있다[2,4]. 이를 위한 연구에서는 여러가지 방법이 사용되었는데 경험적 방법으로 경사치의 재사용과 동적 학습법등이 있고 수치적인 방법으로는 학습률을 조정하는 법등이 있고 간접적인 방법으로 학습의 순서를 조정한다든가 자료의 표현방법을 조정하는 방법등이 있다. 이러한 방법들을 비교분석한 연구도 진행되고 있다[3]. 본 논문에서 제안하는 방법은 위에서 설명한 방법중 첫번째인 경험적 방법에 속한다고 할 수 있는데 네트워크 Local Maxima 에 빠졌을 때 은닉층의 뉴런의 수를 동적으로 변화시켜 오류를 유

도함으로써 Local maxima 를 탈출할수 있게 제안하였다. 또한 최적의 은닉층 뉴런의 수를 결정할수 있도록 구성해보았다. 위의 알고리즘의 성능을 검사하기 위하여 두가지의 문제에 적용하여 보았는데, 첫번째 문제는 간단하면서도 비선형인 XOR 문제이고, 두번째 문제는 좀더 큰 부피를 가지는 문제로 10\*8 의 영문알파벳과 숫자 폰트의 학습에 적용하여 일반적인 역전파 알고리즘과 성능을 비교해 보았다.

2 장에서는 오류역전파 알고리즘에 대해서 간략한 소개와 제안된 알고리즘에 대해서 설명하고 3 장에서 실제로 본알고리즘을 XOR 문제와 10x8 알파벳과 숫자 폰트의 학습문제에 적용하여본 결과를 나타내었다. 4 장에서 결론 및 고찰로 맺었다.

### 2. 알고리즘

#### 2.1 일반적인 오류 역전파 알고리즘

BP(BackPropagation) 알고리즘은 이미 언급한 바와

같이 순방향 다층 신경망의 일반적인 응용에 활용되는 오류 역전파 학습 알고리즘이다. 출력층의 오차 신호를 이용하여 은닉층과 출력층간의 연결강도를 변경하고, 또한 출력층의 오차 신호를 은닉층에 역전파하여 입력층과 은닉층간의 연결강도를 변경하는 학습 방법이다. 역전파 알고리즘은 입력층과 은닉층의 연결강도를 변경 시키는 신호로 델타 신호를 사용하는데 델타신호는 연결강도와 출력층에서의 목표치와 출력치와의 오차인 E와의 관계로부터 구할수 있다[5].

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

$$\delta_{yi} \equiv -\frac{\partial E}{\partial (NET_i)} = (d_i - y_i) f'(NET_i) \quad \text{-----}(2)$$

여기서  $\Delta w_{ij}$ 는 i 번째 입력과 j 번째 출력사이의 연결강도 변화량이고  $\alpha$ 는 양수이다. 전체 오차 E는 다음과 같이 구할수 있다.

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

여기에서 d는 목표치이고 y는 출력치이다.

다층 신경망에서 양극성 입력데이터의 입력층과 은닉층에 전파되는 오차신호는 다음과 같다.

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

$$\delta_z = \frac{1}{2} (1 - z^2) \sum_{i=1}^m \delta_y w_{ij} \quad \text{-----}(5)$$

여기에서  $\delta_y$ 는 은닉층과 출력층 사이에 전달되는 학습신호이고  $\delta_z$ 는 입력층과 은닉층 사이에 전달되는 학습 신호이다.

위의 식(1)에서 보듯이 역전파 알고리즘은 연결강도 변화량을 에러가 작아지는 방향으로 유도하는 Steepest descent 알고리즘을 기초로 하고 있기 때문에 Local maxima 에서 탈출하지 못하는 단점이 있다. Local maxima란 연결강도의 공간에서 오차와의 관계에서 표현할수 있는데 그림 1에서 나타내었다.

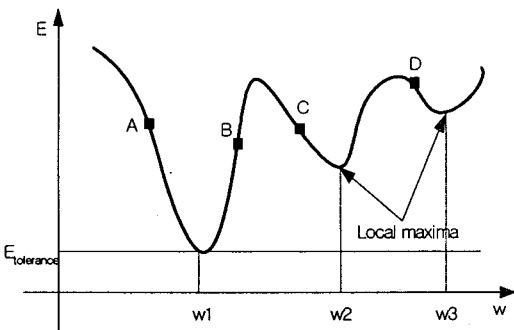


그림 1. 연결강도 공간에서의 Local maxima

그림 1에서 보는 바와 같이 만약 초기 연결강도를 A 점 또는 B 점에 설정하였다면 학습이 진행되면서 E가 감소하는 방향으로 연결강도가 W1의 상태로 변해가 되면서 E가 E\_tolerance까지 줄어들어 결국 수렴하게 된다. 그러나 만약 초기 연결강도를 C나 D로 설정되었다면 각각 W2와 W3로 연결강도가 조정되어 국부최소점인 Local maxima에 빠져 더 이상 학습을 진행하더라도 E가 줄어들지 않는 현상이 발생하여 결국 잘못된 결과를 초래하게 된다. 또한 연결강도를 갱신할 때 활성화 함수의 미분값이 0이 되지 않도록 초기 연결강도를 잘 설정하여야 한다. 한편 연결강도를 너무 적은 값으로 설정하면 학습시간이 길어지게 되는 점도 생각하여 보통 -1에서 1사이의 값으로 설정하는 것이 좋다. 이러한 초기 연결강도를 설정하는 방법에 관한 연구도 진행되었다. 한편 은닉층의 뉴런의 개수를 결정하는 방법에 대해 알아보면 아직까지 이 문제에 대한 해답은 없는 실정이고 타당성있게 받아들여지는 방법을 살펴보면 입력 패턴 공간이 n 차원이고 은닉층의 뉴런수가 p 이면 선형 분리 가능한 영역의 최대수 M은 아래와 같다.

$$M = \sum_{k=0}^n {}_p C_k, \quad k < p \quad \text{-----}(6)$$

M개의 선형 분리 가능한 영역의 문제를 해결하기 위한 은닉층의 뉴런수 p는 다음과 같이 구할수 있다.

$$p = \log_2^M \quad \text{-----}(7)$$

은닉층의 뉴런의 개수는 학습시간과 밀접한 관계가 있고 또한 수가 많다고 학습이 잘되는 것도 아니기 때문에 보통은 여러 번의 실험을 통한 결과를 적용하는 것이 일반적이다.

## 2.2 제안된 알고리즘

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

그러면 언제 어떻게 뉴런을 추가할 것인가를 살펴 보겠다. 네트워크가 Local maxima에 빠졌다는 것을 정확히 판단해 내기는 어렵다. 여기에서 오류 E를 가지고 판단하는데 매 100회의 연결강도를 조정후 E를 검사하여 만약에 E의 차이가 1% 미만이라면 Local maxima에 빠진것으로 판단하고 은닉층의 유닛을 추가한다. 만약에 E가 1%이상 감소하였다면 그 다음 100회를 시행한 후 다시 검사한다. 이때 유닛을 추가하는 방법으로는 두 가지를 사용하였는데 한가지는 연결강도를 0으로 설정하는 방법과 또 한가지는 1과 -1사이의 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 한번 네트워크가 수렴하게 되면 이번에는 다시 은닉층의 뉴런을 하나씩 제거하면서 다시 학습을 하여 수렴하는지를 판단한다. 마지막으로 수렴하지 못할때까지

학습하여 그 전단계의 네트워크를 최종적으로 결정하게 된다.

아래의 그림 2,3 으로 exclusive-OR 문제의 학습과정에 대해서 나타내었다. 이 네트워크는 2 개의 입력과 한 개의 출력으로 구성 하였다.

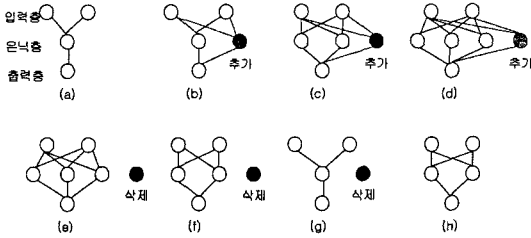


그림 2. 은닉층에 유닛이 추가되고 삭제되는 과정

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

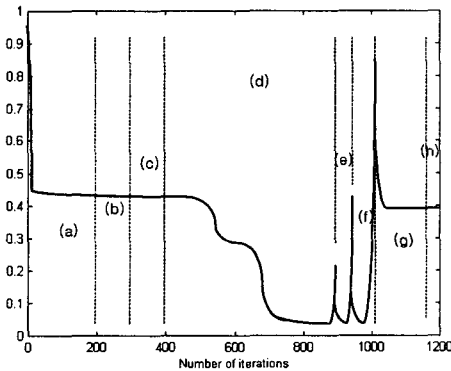


그림 3. 네트워크구조 변경에 따른 E의 변화

초기 연결강도는 랜덤하게 정하였는데 처음 50 번의 학습이 진행되면서 E 가 0.5 감소하였고 100 번의 학습 후에는 E 가 1% 이상 감소하였다. 그래서 은닉층의 뉴런이 추가 되지 않았다. 그다음 100 번의 학습 후에는 E 가 1% 미만으로 감소하였고 그래서 그림 2(b)와 같은 모양으로 유닛이 한 개 추가 되었다. 채워진 동그라미가 새로 추가된 유닛을 표현하고 점선으로 표시된 것은 추가된 유닛에 의해 연결된 연결강도이다. 그 다음 100 번의 학습이 있었으나 E 가 1% 미만으로 감소하고 E 의 최소치에 도달하지 못했기 때문에 그림 2(c)와 같이 유닛이 추가되었다. 400 번의 연결강도가 조정후 또 하나의 은닉층 뉴런이 추가되었고 그림 2(d)에 표시하였다. 그후 E 는 완만하게 감소하다가 810 번의 학습 후에 수렴하였다. 한번 네트워크가 수렴하고 난후 다시 유닛의 삭제 과정에 들어가게 되어서 그림 2(e)와 같이 뉴런을 삭제 하였다. 처음에는 E 가 증가하였지만 40 번의 학습 후에는 수렴하여서 다시 그림 2(f)와 같이 또 하나의 뉴런을 삭제 하였다. 여러 E 는 0.5 까지 증가하였지만 다시 수렴하였고 다시 뉴런을 제거했을 때는 끝내 수렴하지 않았

다. 그래서 그림 2(h)의 모양으로 네트워크를 결정하고 연결강도는 그림 2(f)에서 성공한 연결강도로 결정하였다.

아래의 그림에서 제안한 알고리즘의 FlowChart 를 나타내었다.

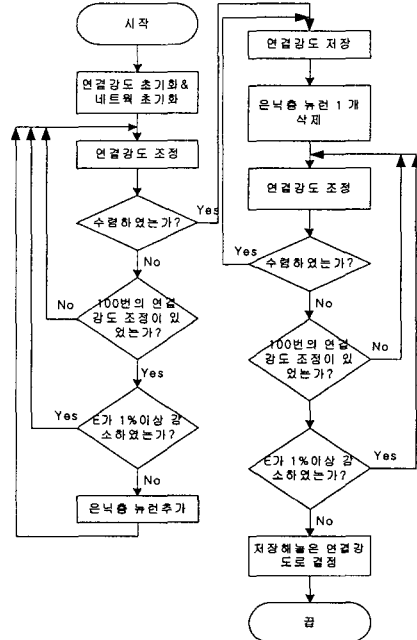


그림 4. 제안된 알고리즘의 FlowChart

### 3. 실험

본 알고리즘의 실험을 위해서 두 가지의 문제를 결정하였다. 간단한 문제로는 XOR 문제이고 좀더 복잡한 문제로 10x8 알파벳과 숫자의 학습문제로 결정하였다. 입력층과 은닉층 사이의 활성화함수에는 다음과 같은 식을 사용하였다.

$$n = \frac{2}{1 + \exp(-2n)} + 1 \quad \text{-----}(8)$$

은닉층과 출력층 사이에는 항등함수(identity function)을 사용 하였다.

#### 3.1 Exclusive-OR

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

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

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

림 5와 같다.

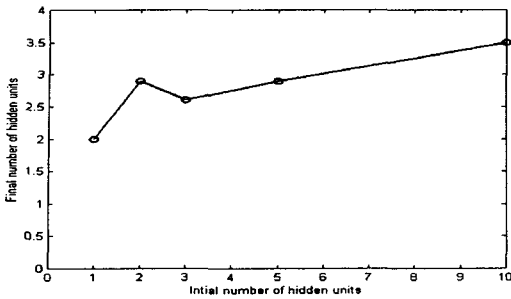


그림 5. 초기 은닉층의 뉴런수와 최종 은닉층의 뉴런수와의 관계

위의 그림에서 알수있듯이 작은 네트워크를 구성하려고 할때는 은닉층의 초기뉴런의 개수를 작게 설정하는 것이 좋다. 또한 뉴런을 감소시키는 과정에서 제거하는 순서에 관해서 실험해 보았는데 최초의 뉴런과 차례로 더해진 뉴런에 순서를 붙여서 제거해 보았다. 먼저 최초의 뉴런을 삭제 하였을 때는 네트워크가 수렴하지 못하는 것을 알수 있었다. 그리고 마지막에 추가된 뉴런부터 삭제를 해보니 네트워크가 수렴하였다. 추가된 뉴런들은 순서에 관계없이 제거 하여도 네트워크가 수렴하였으나 최초의 뉴런을 제거시에는 수렴하지 않는 것을 확인 할 수 있었다.

### 3.2 10x8의 알파벳과 숫자 폰트 학습

Exclusive-OR의 문제에 적용한 네트워크는 최대의 뉴런의 개수가 10 개 정도 밖에 되지 않는 작은 네트워크이었다. 그래서 본 알고리즘을 좀더 큰 문제에 적용하고자 10x8의 알파벳과 숫자 폰트의 학습을 선택하였다. 그림 6에 네트워크의 구성을 간략하게 나타내었다.

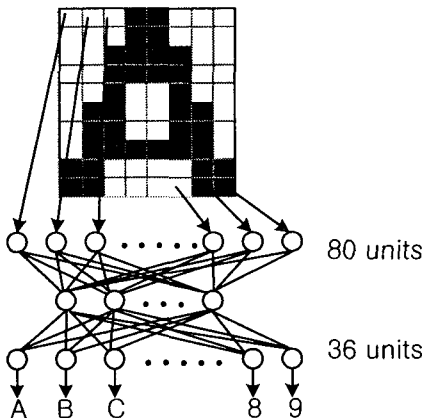


그림 6. 10x8 알파벳과 숫자 폰트 학습을 위한 신경망

입력층에 80 개의 유닛이 존재하고 출력층에는 A-Z까지의 알파벳과 0-9까지 나타낼 수 있게 36개의 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 100 회마다 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 비교해 보고 다음 그림과 같은 결과를 얻을 수 있었다.

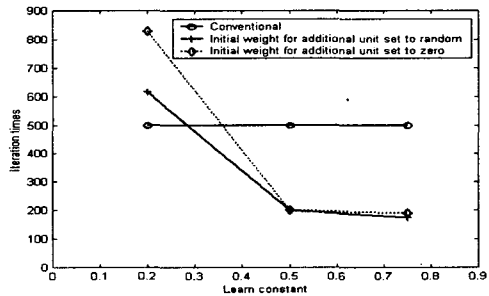


그림 7. 학습률의 변화에 따른 학습능력

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

## 4. 결론 및 고찰

본 논문에서는 은닉층의 뉴런의 개수를 변경시키는 역전파 알고리즘에 대해서 소개하였다. Local maxima에 빠졌을 때 은닉층에 새로운 유닛을 추가하고 연결강도를 랜덤한 값을 주면 Local maxima에서 탈출할수 있었다. 본 알고리즘을 사용하면 더 이상 은닉층의 뉴런의 개수를 결정할 필요도 없게 된다. 10x8 알파벳과 숫자폰트 학습을 하였을 때는 일반적인 역전파 알고리즘보다 1.5배정도 빠르게 학습하였다. 우리는 앞으로 이 알고리즘을 음성인식 분야로 확대 적용해볼 계획이다.

### 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] Ali A. Minai and Ronald D. Williams, "BackPropagation Heuristics : A study of the Delta-Bar-Delta Algorithm", IJCNN, Vol 1, pp. 595-600, 1990, Jun  
 [3] 조성배, 김진형, "오류역전파 학습 알고리즘의 속도 개선방법 : 분류 및 비교", 한국정보과학회 논문지 '91.11 vol.18, No.6 November, pp.649-660  
 [4] Robert Hecht-Nielsen: Theory of the Backpropagation Neural Network, IJCNN, Vol 1, pp. 593-605, 1989  
 [5] Sejnowski. T. J. & Rosenberg. C. R.: Parallel networks that learn to pronounce English text. Complex Systems. 1, 145-168. 1987