가속화 알고리즘을 이용한 EBP의 학습 속도의 개선에 관한 연구

°최 희창, 권 희용, 황 희융
서울대학교 컴퓨터 공학과

A study on the improvement of the EBP learning speed using an acceleration algorithm

Hee Chang Choi, Hee Yong Kwon, Hee Yeung Hwang
Dept. of Computer Eng., Seoul National Univ.

Abstract

In this paper, an improvement of the EBP(error back propagation) learning speed using an acceleration algorithm is described. Using an acceleration algorithm known as the Partan method in the gradient search algorithm, learnig speed is 25% faster than the original EBP algorithm in the simulaion results.

1. 서론

1950년대 말에 인간의 신경 구조를 모방한 아주 간단한 모형인 퍼셉트론(Perceptron) 이라는 것이 미국의 Rosenbratt에 의하여 만들어졌다. 퍼셉트론을 가지고서 많은 사람들이 실험을 한 결과 어떤 문제에 대해서는 아주 잘 적용되어질 수 있음이 증명되었다. 그러나 그 후에 Minsky와 Papert는 "Perceptron(1969)"이라는 저서를 통해 퍼셉트론의 한계를 지적하였는데 그 대표적인 예가 exclusive or(xor) 문제이다.[1,7] 즉 퍼셉트론을 가지고는 xor문제를 해결할 수 없음이 밝혀졌다. 이 후로 많은 학자들이 이 방면의 연구를 포기하고 소수의 학자들만이 연구를 계속해 왔었다.[4,5] 그러던 중 1980년대에 들어서면서 D.Rumelhrat, J.MaClelland등을 주축으로 하는 일련의 연구 그룹(PDP 그룹)이 형성되어 병렬 분산 처리(Parallel distributed processing) 이라는 슬로건을 내걸고 많은 연구가 다시 진행되기 시작했다. 이 그룹에서 퍼셉트론의 한계로 지적되어진 문제에 대한 해결책들을 비록 수학적으로 완전히 증명되어지진 않았지만 제시하였다.[1,2]

이러한 해결책으로 제시된 것의 하나가 바로 신경회로망의 구조를 다층화하여 입출력 층안에 내부 층을 두어 문제를 해결하는 것으로 이는 Minsky에 의하여도 생각되어졌으나 그 당시에는 내부 층을 학습시킬 수 있는 적절한 학습 방법의 부재가 문제였었다. PDP그룹에서는 내부 층을 학습시키는데 유용한 학습 법칙을 고안했는데 이것이 바로 EBP이다. 본 연구에서는 EBP에 대하여 살펴보고 EBP에 있어서의 학습 속도의 개선에 관하여 알아본다.

2. 본론

EBP는 다름 아닌 단층 구조의 신경 회로망의 유용한 학습 법칙으로 알려진 델타 법칙(Delta rule) 의 일반화이다. 우선 간단히 델타 법칙에 관하여 알아보기로 하자.

1) 델타 법칙(Delta rule)

델타 법칙은 연결된 두 노드의 연결 강도의 변화량에 대한 법칙으로써 다음과 같이 나타낼 수 있다.

$$\Delta W_{ji} = n(T_j - O_j)A_i$$

여기서 W_{ji}는 노드 i에서 노드 j로의 연결 강도(weight)를 표시하고, n은 학습 비율 (learning rate), T_j는 노드 j의 원하는 출력 값, O_j는 노드 j의 실제 출력 값, A_i는 노드 i의 활성화 값을 나타낸다. 이 법칙은 각각의 학습 주기마다 $\delta_j = T_j - O_j$ 의 값을 구한 다음 입력의 활성화 정도를 곱한 값을 입출력 노드간의 연결 강도를 보정해 주는 값으로

사용하는 것으로서, 보정된 연결 강도는 전체 시스템의 에러를 최소화 하는 값이다. 이를 좀더 자세히 살펴보면 우선 전체 시스템의 에러는 다음과 같이 주어진다.

$$E = \sum_p E_p = \sum_p \sum_i (t_{pi} - o_{pi})^2$$
(total sum of square)

여기서 p는 입력 패턴의 갯수를 가리킨다. 이러한 전체 시스템의 에러를 최소화하는 방법의 하나로서 델타 법칙은 gradient-decent 방법을 이용하는데 gradient는 다음과 같이 계산되어진다.[3]

$$\Delta w_{ij} \doteq -k\frac{\partial E}{\partial w_{ij}}$$

이 식을 풀어 보면

$$\Delta w_{ij} = \epsilon \delta_{pi} i_{pj}$$

되어 델타 법칙과 같음을 알 수 있다.

2) 일반화된 델타 법칙

다층 구조의 신경망에서의 유용한 성질은 입출력 계층간의 내부 노드의 존재 및 노드의 활성화 함수(activation function)의 비선형성에 기인한다. 선형 활성 함수를 가지는 다층 구조의 신경망은 같은 성질을 만족하는 단층 구조의 신경망으로 구현되어질 수 있음은 쉽게 증명되어질 수 있다. 비선형 활성 함수를 가지는 다층 구조 신경망의 학습 법칙은 다음과 같다. 한 노드의 입력과 출력은 다음과 같이 정의되어 진다.

입력 —— $net_{pj} = \sum_i w_{ji} o_{pi}$

출력 —— $o_{pj} = f_j(net_{pj})$

여기서 f가 바로 비선형 활성 함수인데 이는 미분 가능해야 하고 증가 함수이어야 한다. 이러한 함수로 많이 쓰이는 것이 logistic 함수로서 다음과 같다.

$$f_j(net_{pj}) = \frac{1}{1+e^{-net_{pj}}}$$

다층 구조 신경망에서 연결 강도의 변화량을 다음과 같이 가정한다.

$$\Delta_p w_{ji} \propto -\frac{\partial E_p}{\partial w_{ji}}$$

이를 풀면 다음과 같이 된다.

$$\Delta w_{ij} = \epsilon \delta_{pi} i_{pj}$$

이는 앞서 살펴보았던 델타 법칙과 동일함을 알 수 있다. 그러나 일반화된 델타 법칙에서는 δj의 계산이 노드 j가 출력 노드일 경우와 내부 노드일 경우에 다르게 계산된다. 즉 노드 j가 출력 노드라면 δj는 다음과 같이 주어지고

$$\delta_{pi} = (t_{pi} - o_{pi}) f'_i(net_{pi})$$

노드 j가 내부 노드일 경우 δj는 재귀적으로 다음과 같이 구해질 수 있다.

$$\delta_{pi} = f'_i(net_{pi}) \sum_k \delta_{pk} w_{ki}$$

이 식을 설명하면 내부 노드에서의 oj는 바로 윗 노드에서 계산된 δk를 이용하여 구해지는데 이 값은 노드 j로 들어오는 입력의 활성 함수의 미분 값,즉 활성의 변화량에 비례하여 조정함을 알 수 있다. 이러한 방법으로 에러를 아래 노드로 전달하면서 시스템의 에러를 최소화하는 방법이 일반화된 델타 법칙이다.

3) 가속화 알고리즘

EBP에 있어서는 몇가지 패턴을 학습시키는데 있어서 상당히 많은 시간이 필요하게 된다. 이러한 문제는 학습시킬 패턴의 수가 굉장히 많아지면 상당한 문제가 된다. [6] 이러한 문제의 개선을 위하여 gradient_decent의 가속화 알고리즘을 EBP에 적용하여 그 학습 속도를 단축시켰다. 이러한 가속 알고리즘은 다음과 같다.

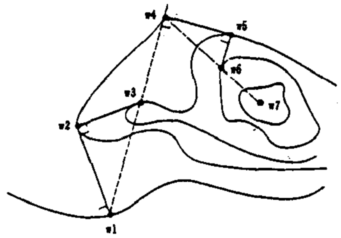
<u>알고리즘</u>

step 1: 원래 방법으로 연결 강도 W_i, W_{i+1}, W_{i+2} 계산

step 2: $W_{i+3} = c(W_{i+2} - W_i)$ (c: 가속화 상수)

step 3: i를 3 증가하고 step 1으로 가서 반복

Gradient_decent에서는 각 학습 주기마다 그 시점에서의 gradient를 구하고 그 반대 방향으로 수렴 방향을 보정하여 에러를 감소시키는데 반하여 가속화 방법에서는 세번째 주기까지는 gradient_decent 방법과 동일하게 gradient를 구하고 네째 주기에서는 앞의 세 단계에서 구한 gradient를 이용하여 복잡한 계산 없이 방향을 잡는 것으로 이를 그림으로 표현하면 다음과 같다.



(그림 1) 가속화 알고리즘의 기하학적 표현

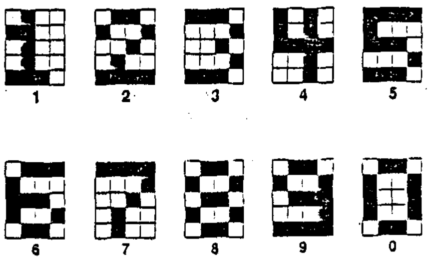
이를 수식으로 표현하면 다음과 같다.

$$\Delta Wji(t+1) = Wji(t-2) + c(Wji(t) - Wji(t-2))$$

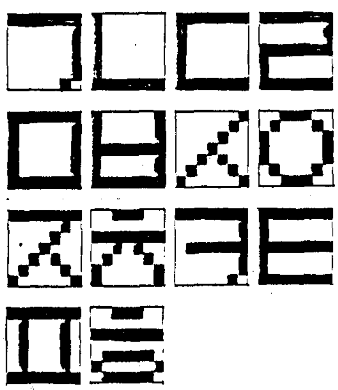
여기서 c는 가속화 상수이다.

4) 실험 및 결과

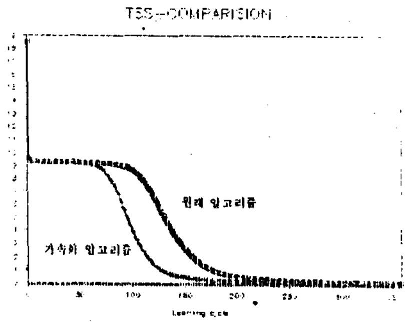
가속화 알고리즘을 이용한 EBP 알고리즘의 구현은 IBM PC/AT 상에서 C언어로 구현되었다. 이 실험은 0-9의 숫자와 ㄱ-ㅎ의 두 종류의 패턴에 대하여 행한 것이다. 숫자의 경우 입력 노드 수는 5*4, 내부 노드 수는 20개이고 문자 입력의 경우는 입력 노드 7*7, 내부 노드는 49개이다. 입력 패턴은 각각 (그림 2)와 (그림 3)에 보이고 있다. 실험에 사용한 학습 파라미터는 학습 비율이 3.0이고 가속화 상수는 1.6으로 하였으며 결과는 각 입력 패턴에 대해 각각 (표 1)과 (표 2)에 보이고 있다.



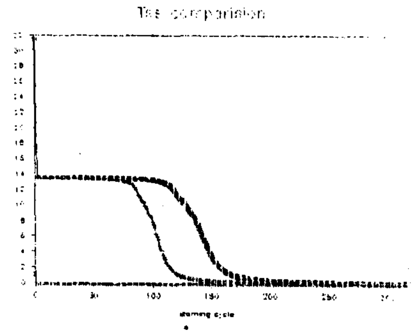
(그림 2) 숫자 입력 패턴



(그림 3) 문자 입력 패턴

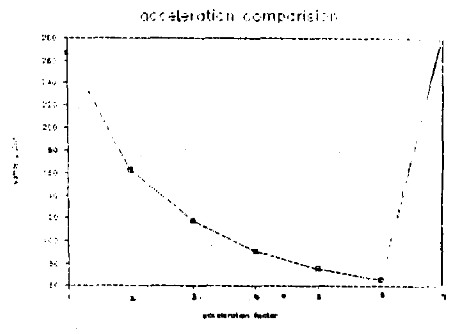


(표 1) 숫자 입력에 대한 학습 속도의 비교



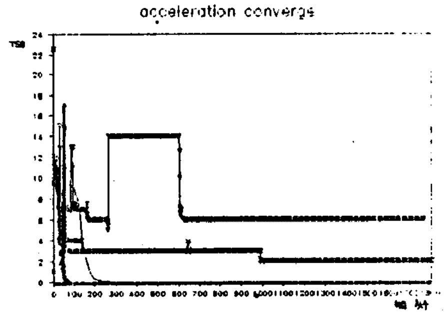
(표 2) 문자 입력에 대한 학습 속도의 비교

가속화 상수의 변화에 따른 학습 속도의 변화를 측정하기 위해 숫자 입력의 경우에 대해 가속화 상수를 1,2,....,6로 했을 때의 학습 속도를 (표 3)에 보였다.



(표 3) 가속화 상수의 변화에 따른 학습 속도의 비교

(가속화 상수가 1인 경우는 실제 가속을 행하지 않은 것과 동일하다.) 또 다른 실험에서 가속화 상수를 7이상으로 했을 때는 실제 학습을 행하는 속도가 매우 늦어 학습을 하지 못하는 것으로 나타났는데 이는 (표 4)에 보이고 있다.



(표 4) 가속화 상수가 너무 큰 경우의 학습 (c => 7)

5. 결론

본 연구는 EBP의 학습 속도는 가속화 알고리즘을 사용하여 개선할 수 있음을 보였다. 앞서 살펴본 바와 같이 다층 구조의 학습 법칙이 간단하지만은 실제 학습시의 계산량은 상당히 많음을 알 수 있다. 이러한 단점을 개선하기 위하여는 다층 구조의 신경 회로망을 학습시킬 수 있는 새로운 학습 법칙이 연구되어져야 할 것이다. 앞으로의 연구에서는 보다 효율적이고 강력한 학습 법칙의 개발이 요구되어진다.

6. 참고 문헌

[1]. D.Rumelhart,G.Hinton,and R.Williams, "Learning internal representations by error back propagation", Parallel Distributed Processing vol.1 pp. 318-362 MIT press, 1986

[2]. D.Rumelhart,G.Hinton,and R.Williams, "Learning representation by Back-proragation errors", Nature 323 pp. 533-536, 1986

[3]. Masanao Aoki, "Introduction to optimization techniques", Macmillian Company, 1971

[4]. Dr Kamal karma and David Breen, "An artificial neural networks tutorial : part 1-basics", Neural Networks, vol 1, No 1, pp. 4-22, january 1989

[5]. R.Lippman, "An Introduction to computing with neural nets", IEEE ASSP, pp. 4-22, April 1989

[6]. D.R.Rush and J.M.Salas, "Improving the learning rate of Back-propagation with the gradient reuse algorithm", IEEE International conference on neural network, pp.441-447, 1988

[7]. M.Minsky and S.Papert, "Perceptrons", The MIT Press, 1969