1999 年度 大韓電子工學會 夏季綜合學術大會 論文集 第 22 卷 第 1 號 99/6

다층 신경망을 위한 Multi-gradient 학습 알고리즘

고 진 욱, 이 철 희
연세대학교 전기 · 컴퓨터 공학과
Tel: (02) 361-2779, E-mail: chulhee@bubble.yonsei.ac.kr

Multi-gradient learning algorithm for multilayer neural networks

Jinwook Go and Chulhee Lee
Department of Electrical and Computer Engineering, Yonsei University

ABSTRACT

Recently, a new learning algorithm for multilayer neural networks has been proposed [1]. In the new learning algorithm, each output neuron is considered as a function of weights and the weights are adjusted so that the output neurons produce desired outputs. And the adjustment is accomplished by taking gradients. However, the gradient computation was performed numerically, resulting in a long computation time. In this paper, we derive the all necessary equations so that the gradient computation is performed analytically, resulting in a much faster learning time comparable to the backpropagation. Since the weight adjustments are accomplished by summing the gradients of the output neurons, we will call the new learning algorithm "multi-gradient." Experiments show that the multi-gradient consistently outperforms the backpropagation.

1. 서론

다층 신경망은 패턴 인식, 동적 시스템의 제어와 같은 많은 비선형 신호 처리 문제에 성공적으로 적용되었다 [2]. 특히, 오류 역전파(error backpropagation) 학습 알고리즘의 발견과 보편화는 신경망과 관계된 연구를 활성화시키는 계기를 마련하였다. 오류 역전파 알고리즘은 각 출력 뉴런(neuron)의 자승 오차의 합인 오차 함수를 정의하여 이 오차 함수를 감소시키도록 웨이트(weight)를 변경시키는 학습 방법이다. 그러나 오류 역전파 알고리즘은 일반적인 성공에도 불구하고 긴 수렴시간과 국부 최소(local minima)와 같은 문제점을 지니고 있다. 이러한 문제점을 해결하기 위해 학습 파라미터(parameter)의 최적화[4], 오차함수의 변형과 같은 많은 향상된 학습 알고리즘들이 발표되었다.

본 논문에서는 학습 과정을 다른 관점에서 생각하여 새로운 학습 방법을 제안한다. 제안한 학습 알고리즘은 출력층의 오차 함수를 최소화하는 대신 출력층이 주어진 입력 패턴에 대한 목적하는 패턴을 출력하도록 웨이트를 변경시킨다. 이는 신경망의 출력을 입력과 웨이트의 함수로 보고, 각 출력의 gradient를 구하여 웨이트를 갱신한다.

2. 본론

2.1 순방향 다층 신경망

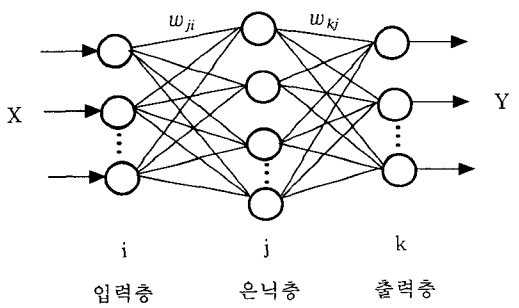

그림 1. 3층 순방향 신경망.

신경망은 뉴런이라는 연산 단위들의 상호 연결로서 일반적으로 입력층, 다수의 은닉층, 그리고 출력층으로 구성된다. 그림 1은 순방향 3층 신경망의 예이다. 각

층은 각각 N_0, N_1, N_2개의 뉴런으로 구성되고, 입력층과 은닉층을 연결하는 웨이트를 w_{ji}, 은닉층과 출력층을 연결하는 웨이트를 w_{kj}로 나타냈다. 입력 신호를 $x_j\,(j=1,2,\cdots,N_0)$ 라고 할 때 일반적인 뉴런의 출력 o_j는 다음과 같은 함수로 나타낼 수 있다.

$$o_j = K\Phi\left(\sum_{i=1}^{N_0} w_{ji}x_i - \theta_j\right) \qquad (1)$$

이때, K는 상수, Φ는 비선형 함수, θ_j는 threshold 값을 나타낸다.

그림 1에서 입력 벡터는 $X=(x_1,x_2,\cdots,x_{N_0})^T$ 이고, 출력 벡터는 $Y=(y_1,y_2,\cdots,y_{N_2})^T$ 이다. 신경망의 전체 웨이트 개수를 N로 표시하면 $N=(N_1N_0+N_2N_1)$이 되고, 입력층과 은닉층, 은닉층과 출력층을 연결하는 웨이트 행렬들을 하나의 N 차원 웨이트 벡터 W로 다음과 같이 나타낼 수 있다.

$$\begin{aligned} W &= (w^1_{11}, w^1_{12}, \cdots, w^1_{N_1N_0}, w^2_{11}, w^2_{12}, \cdots, w^2_{N_2N_1})^T \\ &= (w_1, w_2, \cdots, w_{N-1}, w_N)^T \quad . \end{aligned} \qquad (2)$$

2.2 제안된 학습 알고리즘

출력 벡터 Y는 X와 W의 함수로 나타낼 수 있다.

$$Y = F(W,X) \qquad (3)$$

Y를 각각의 뉴런 출력으로 나타내면 다음과 같다.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{N_2} \end{bmatrix} = \begin{bmatrix} f_1(W,X) \\ f_2(W,X) \\ f_3(W,X) \\ \vdots \\ f_{N_2}(W,X) \end{bmatrix} \qquad (4)$$

입력 패턴에 대한 클래스 결정 규칙(decision rule)이 가장 큰 출력값을 갖는 출력 뉴런에 대응하는 클래스를 선택하는 것이면, 학습 패턴쌍을 입력 X의 클래스에 속하는 출력의 목표치는 0.9로 나머지 출력의 목표치는 0.1로 설정할 수 있다. 이러한 학습 패턴쌍을 고려하여 입력 X가 클래스 $\omega_i\,(i=1,\cdots,N_2)$에 속하면 식 (4)의 y_i는 증가시키고 y_i를 제외한 나머지 출력은 감소시키는 방향으로 웨이트를 변화시켜 줄 수 있다. 이는 각 출력의 W에 대한 gradient를 이용하여 구할 수 있고 y_i의 gradient는 다음과 같다.

$$\nabla y_i = \frac{\partial y_i}{\partial w_1}\overrightarrow{w_1} + \frac{\partial y_i}{\partial w_2}\overrightarrow{w_2} + \cdots + \frac{\partial y_i}{\partial w_N}\overrightarrow{w_N} \qquad (5)$$

이때 $i=1,2,\cdots,N_2$ 이고, $\{\overrightarrow{\omega_1}, \overrightarrow{\omega_2}, \cdots, \overrightarrow{\omega_N}\}$ 는 각각 N차원 공간의 한 기저(basis)를 이룬다. 각 출력의 웨이트에 대한 편미분은 다음과 같이 구할 수 있다. 먼저, 은닉층과 출력층간의 웨이트에 대한 편미분은 다음과 같다.

$$\frac{\partial y_k}{\partial w_{j,i}} = \begin{cases} y_k(1-y_k)z_i & \text{if } j=k \\ 0 & otherwise \end{cases} \qquad (6)$$

이때 $w_{j,i}$는 은닉층의 뉴런 i와 출력층의 뉴런 j간의 웨이트이고, z_i는 은닉층 뉴런 i의 출력이다. 또한 입력층과 은닉층간의 웨이트에 대한 편미분은 다음과 같다.

$$\frac{\partial y_k}{\partial v_{j,i}} = y_k(1-y_k)w_{kj}z_j(1-z_j)x_i \qquad (7)$$

이때 $v_{j,i}$는 입력층의 뉴런 i와 은닉층의 뉴런 j간의 웨이트이고, w_{kj}는 은닉층의 뉴런 j와 출력층의 뉴런 k간의 웨이트이다.

예를 들어, 학습 패턴이 2개의 클래스 ω_1,ω_2 로 구성되었을 때, 입력 X가 클래스 ω_1에 속한다면 웨이트 벡터 W를 $c_1\nabla y_1 - c_2\nabla y_2$ 의 방향으로 이동시켜 y_1은 증가시키고 y_2는 감소시킬 수 있다. 반대의 경우에는 W를 $c_1\nabla y_2 - c_2\nabla y_1$ 의 방향으로 이동시켜 y_2는 증가시키고 y_1는 감소시킬 수 있다. 이때, c_1, c_2 는 양의 상수이다.

일반적으로 학습 패턴이 P개의 클래스로 구성될 때, multi-gradient 알고리즘의 웨이트 벡터 W는 다음과 같이 갱신할 수 있다.

$$W(t+1) = W(t) + \gamma \sum_{i=1}^{P} c_i \nabla y_i \qquad (8)$$

이 때, $c_i = \begin{cases} \geq 0 & \text{if } X \in w_i \\ < 0 & otherwise \end{cases}$,

γ는 학습률(learning rate)이다. 식 (8)의 각 gradient 벡터의 계수 c_i를 구하는 여러 가지 가능성이 존재하지만, 본 논문에서는 c_i를 다음과 같이 계산한다.

$$c_i = \begin{cases} t_i - y_i & \text{if}(t_i=0.9 \text{ and } y_i<0.9) or (t_i=0.1 \text{ and } y_i>0.1) \\ 0 & otherwise \end{cases}$$

만약 c_i를 $t_i - y_i$ 와 같이 구한다면, multi-gradient는 역전파 알고리즘과 동일한 결과를 보여준다.

3. 실험 및 고찰

제안된 multi-gradient 알고리즘의 성능을 평가하기 위해 원격탐사 프로그램(LACIE)에 [3] 의해 채집된 실제 데이터를 사용하였다. 본 논문에서는 7차원의 10 클래스를 분류하는 문제에 대해 은닉층 뉴런 수와 학

습률을 변화시키며 실험하였다. 표 1은 10개의 클래스에 대한 정보를 나타낸다. 각 클래스 당 300개를 무작위로 선택하여 학습 데이터로 사용하였고, 나머지 데이터는 모두 시험 데이터로 사용하였다.

표 1. 10개의 클래스 정보.

Data	Location	Species	No. data
770308	Finney Co. KS	Winter Wheat	691
770626	Finney Co. KS	Winter Wheat	677
771018	Hand Co. KS	Winter Wheat	662
770503	Finney Co. KS	Winter Wheat	658
770626	Finney Co. KS	Summer Fallow	643
780726	Hand Co. KS	Spring Wheat	518
780602	Hand Co. KS	Spring Wheat	517
780515	Hand Co. KS	Spring Wheat	474
780921	Hand Co. KS	Spring Wheat	469
780816	Hand Co. KS	Spring Wheat	464

본 논문의 모든 실험에서 3층 신경망을 학습시켰고, 제안된 알고리즘과의 성능 비교를 위해 오류 역전파 알고리즘을 사용하였다. 그림 2-4에서, 은닉층의 뉴런 개수를 입력 차원의 2배인 14개를 사용하고, learning rate 0.3, momentum rate 0.1을 사용하여 다른 초기 웨이트로 10번 반복 실험한 결과의 평균을 나타내었다. 학습 데이터와 시험 데이터에 대해 1-2% 정도의 성능 향상을 보여준다. 주목할만한 사항은 그림 4에서 보는 바와 같이 두 알고리즘의 학습시의 평균 자승 오차(mean squared error)의 크기이다. 제안된 알고리즘은 같은 iteration 상에서 역전파 알고리즘과 비교하여 오차가 큼에도 불구하고 우수한 학습 성능을 보여주었다. 이는 MSE와 분류 오차(classification error)가 반드시 비례하지는 않음을 재확인시키고 [2], 제안된 알고리즘과 역전파 알고리즘의 학습 결과가 상이함을 보여준다.

그림 5-6은 14개의 은닉층 뉴런을 사용하고 momentum은 사용하지 않은 상태에서, 두 알고리즘 각각에 대해 학습률의 크기 변화에 따른 학습 성능의 변화를 보여준다. 보는 바와 같이, 수렴 속도와 분류 정확도 면에서 제안된 알고리즘의 성능이 우수함을 알 수 있다. 제안된 알고리즘은 0.7과 1정도의 비교적 큰 학습률에 대해서도 수렴 성능에 영향을 주지 않는 것으로 보인다. 반면 역전파 알고리즘은 학습률의 크기가 커짐에 따라 최종 학습 성능이 저하됨을 알 수 있다. 그러므로 제안된 알고리즘은 우수한 학습 성능을 얻으면서 빠른 수렴 속도를 기대할 수 있다.

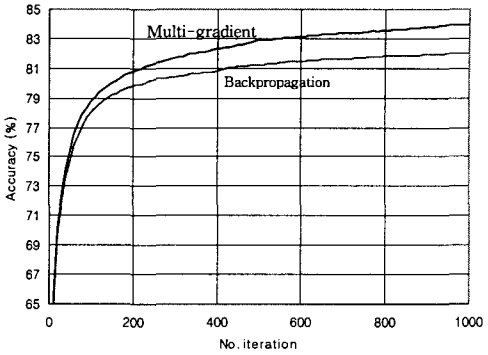

그림 2. 학습 데이터에 대한 성능 비교.

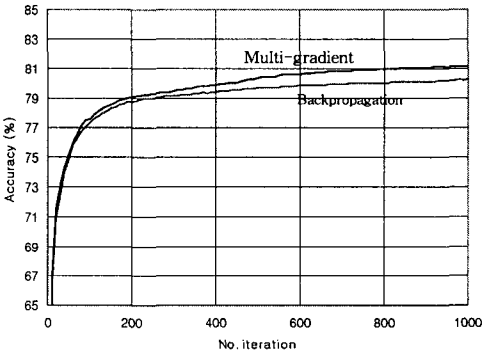

그림 3. 시험 데이터에 대한 성능 비교.

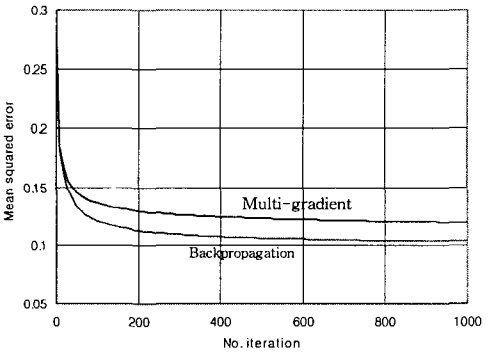

그림 4. 학습시의 MSE 변화.

표 2-3에서, 은닉층의 뉴런의 개수를 1개에서 13개까지 변화시키며 iteration 1000번 동안 학습시켰을 때(동일한 파라미터 사용), 최종 학습 성능을 비교하였다. 제안된 알고리즘은 은닉층 뉴런의 개수가 2개 이하일 때에도 안정적인 학습 곡선을 보여주었다. 반면 역전파 알고리즘은 진동과 성능 저하의 불안정한 모습

을 보여 주었다. 또한 표에서 알 수 있듯이, multi-gradient의 학습 성능이 일관되게 역전파 알고리즘보다 우수하다.

4. 결론

본 논문에서는, 신경망의 각 웨이트를 N차원 공간의 한 점으로 보고 학습 과정을 N차원 공간상에서 주어진 입력에 대한 바람직한 출력을 갖도록 하는 한 점을 찾는 과정으로 생각한다. 이러한 점을 찾기 위해 각 출력의 웨이트에 대한 gradient를 구하고, 이들의 전체 합에 의해 N차원 웨이트 벡터의 이동 방향을 결정하는 multi-gradient 학습 방법을 제안하였다. 실험을 통해 제안된 방법이 기존의 오류 역전파 알고리즘과 비교하여 학습 성능의 향상을 나타냄을 확인하였다.

참고문헌

[1] 고진욱, 이철희, "새로운 다층 신경망 학습 알고리즘", 대한전자공학회 추계 학술대회, pp. 1285-1288, 1998.

[2] S. Haykin, *Neural Networks*, New York: Macmillan, 1994.

[3] L.L. Biehl and et al.,"A Crops and Soils Data Base For Scene Radiation Research," *Proc. Machine Process. of Remotely Sensed Data Symp., West Lafayette, Indiana,* 1982.

[4] R. Battiti, "Accelerating Backpropagation learning, two optimization methods," *Complex Syst,*, vol. 3, pp. 331-342, 1989.

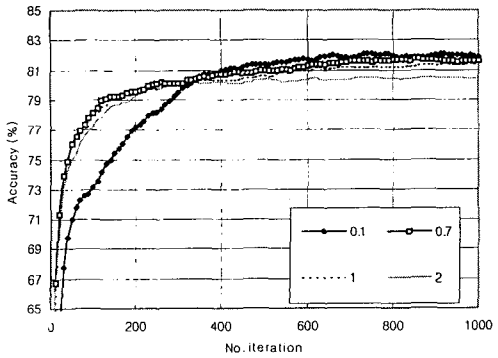

그림 5. 학습률에 따른 성능 비교(BP).

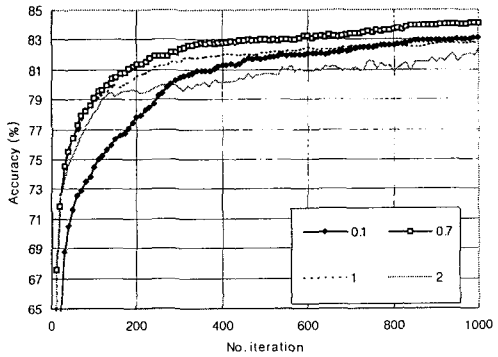

그림 6. 학습률에 따른 성능 비교(multi-gradient).

표 2. 은닉층의 뉴런 개수의 변화에 따른 성능 비교(학습 데이터).

No. hidden neuron	Back propagation	Multi-gradient	Difference of Accuracy
1	27.6%	34.1%	6.5%
2	43.8%	51.4%	7.6%
3	57.5%	59.0%	1.5%
4	69.7%	71.9%	2.2%
5	74.6%	75.4%	0.8%
6	74.3%	76.8%	2.5%
7	76.2%	78.5%	2.3%
8	78.2%	80.4%	2.2%
9	78.6%	80.6%	2.0%
10	79.0%	82.0%	3.0%
11	79.9%	81.8%	1.9%
12	81.1%	82.8%	1.7%
13	81.8%	84.3%	2.5%

표 3. 은닉층의 뉴런 개수의 변화에 따른 성능 비교(시험 데이터).

No. hidden neuron	Back propagation	Multi-gradient	Difference of Accuracy
1	34.4%	39.5%	5.1%
2	53.1%	54.2%	1.1%
3	57.3%	59.0%	1.7%
4	69.6%	71.1%	1.5%
5	74.1%	76.1%	2.0%
6	73.9%	76.9%	3.0%
7	75.2%	79.3%	3.1%
8	76.6%	78.7%	2.1%
9	77.4%	80.2%	2.8%
10	77.3%	79.2%	1.9%
11	79.8%	80.6%	0.8%
12	79.8%	80.9%	1.1%
13	79.2%	81.0%	1.8%