

새로운 다층 신경망 학습 알고리즘

고 진욱, 이 철희

연세대학교 전자공학과

Tel: (02)361-2779, E-mail: gojw@feature.yonsei.ac.kr

A new learning algorithm for multilayer neural networks

Jinwook Go and Chulhee Lee

Department of Electronic Engineering, Yonsei University

ABSTRACT

In this paper, we propose a new learning algorithm for multilayer neural networks. In the error backpropagation that is widely used for training multilayer neural networks, weights are adjusted to reduce the error function that is sum of squared error for all the neurons in the output layer of the network. In the proposed learning algorithm, we consider each output of the output layer as a function of weights and adjust the weights directly so that the output neurons produce the desired outputs. Experiments show that the proposed algorithm outperforms the backpropagation learning algorithm.

1. 서론

다층 신경망은 지도 학습(supervised learning)방법을 통해 많은 난해한 문제들을 해결하는데 성공적으로 적용되었다 [1]. 특히 오류 역전파 알고리즘은 다층 신경망을 학습시키는 가장 보편화된 방법이다. 오류 역전파 알고리즘은 각 출력 뉴런(neuron)의 자승 오차의 합을 최소화하기 위해 출력층(output layer) 오차 신호를 이용하여 은닉층(hidden layer)과 출력층간의 연결 강도를 변경하고, 또한, 출력층 오차 신호를 은닉층에 역전파하여 입력층(input layer)과 은닉층간의 연결 강도를 변경하는 학습 방법이다.

본 논문에서는 학습 과정을 다른 관점에서 생각하여 새로운 방법을 제안한다. 제안한 학습 알고리즘은 출력층의 오차 합수를 최소화하는 대신 출력 뉴런들이

원하는 출력을 나타내도록 연결 강도를 변경시킨다. 이는 각 출력 뉴런에 대해 출력 오차를 가중한 gradient 벡터를 구하고, 이 벡터들의 합의 방향으로 연결 강도를 이동시킴으로써 이루어진다.

제안한 방법의 성능을 평가하기 위해 패턴 분류 문제에 대해 실험한 결과 오류 역전파 알고리즘과 비교하여 학습의 정확도에서 향상을 보였다.

2. 본론

2.1 오류 역전파 학습 알고리즘

일반적으로 다층 신경망은 입력층, 여러 개의 은닉층, 그리고 출력층으로 구성된다. 각 층은 각각 N_c 개의 뉴런으로 이루어진다고 가정하며 $c=0, \dots, m$ 은 각 층(layer)을 나타낸다. $c=0$ 은 입력층, $c=1, \dots, m-1$ 은 은닉층, 그리고 $c=m$ 은 출력층을 나타낸다. 그럼 1은 3층으로 이루어진 신경망의 예이다. 각 층은 i, j, k 로 표시하고 층 i 와 층 j 를 연결하는 연결 강도를 w_{ji} , 층 j 와 층 k 를 연결하는 연결 강도를 w_{kj} 로 나타낸다. 입력층은 N_0 , 은닉층은 N_1 , 출력층은 N_2 개의 뉴런으로 구성된다. PE(processing element)는 비선형 활성 함수(activation function)로서 시그모이드(sigmoid) 함수나 tanh 함수를 주로 사용한다. 입력 벡터 X 가 입력층에 가해지고 연결 강도와 곱해져서 은닉층의 각 뉴런에 전달되고 전달된 값은 활성 함수를 거쳐 각 뉴런의 출력이 된다. 이러한 과정이 출력층까지 이어져 신경망의 출력이 만들어진다. 이 때 적절하게 학습되지 않은 신경망은 잘못된 출력을 준다. 하나의 입력이 가해졌을 때 신경망에서 만들어지는 오차 합수는 다음과 같다.

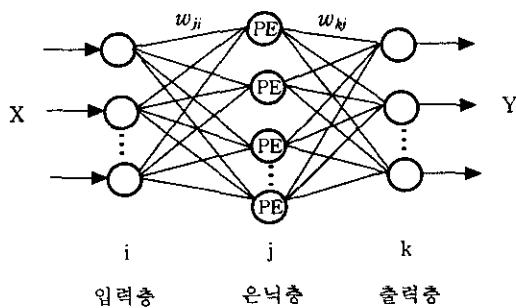


그림 1. 3층 순방향 신경망.

$$E = \frac{1}{2} \sum_k (t_k - o_k)^2 \quad (1)$$

t_k 는 원하는 값(desired value)이고 o_k 는 입력에 대한 출력층의 활성함수로부터의 실제 출력을 나타낸다. 합은 모든 출력층 뉴런의 오차에 대한 것이다. 오류 역전과 알고리즘은 식 (1)의 합수를 최소화하기 위해 함수의 인자인 연결 강도 값을 함수의 편미분의 음의 값에 비례하도록 조정한다.

$$\Delta w_{ji} \propto -\frac{\partial E}{\partial w_{ji}}$$

$$\Delta w_{kj} \propto -\frac{\partial E}{\partial w_{kj}}$$

2.2 제안한 학습 알고리즘

그림 1에서 입력 벡터는 $X = (x_1, x_2, \dots, x_{N_i})^T$ 이고,

출력 벡터는 $Y = (y_1, y_2, \dots, y_{N_o})^T$ 이라고 생각하자.

입력층 i 와 은닉층 j , 은닉층 j 와 출력층 k 를 연결하는 연결 강도 행렬을 각각 W_1, W_2 라고 정의하면 이를

$$W_1 = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N_0} \\ w_{21} & w_{22} & \cdots & w_{2N_0} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N_1-11} & w_{N_1-12} & \cdots & w_{N_1-1N_0} \\ w_{N_11} & w_{N_12} & \cdots & w_{N_1N_0} \end{bmatrix} \quad (2-a)$$

$$W_2 = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N_1} \\ w_{21} & w_{22} & \cdots & w_{2N_1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N_2-11} & w_{N_2-12} & \cdots & w_{N_2-1N_1} \\ w_{N_21} & w_{N_22} & \cdots & w_{N_2N_1} \end{bmatrix} \quad (2-b)$$

와 같이 나타낼 수 있다. 식 (2-a), (2-b)의 연결 강도 행렬의 모든 원소(element)의 개수를 L 로 표시하면 $L = (N_1N_0 + N_2N_1)$ 이 된다. W_1 과 W_2 를 L 개의 구성 요소(component)를 갖는 L 차원 연결 강도 벡터 W 로 나타

내면 다음과 같다.

$$W = (w_{11}^1, w_{12}^1, \dots, w_{N_0N_1}^1, w_{11}^2, w_{12}^2, \dots, w_{N_0N_1}^2)^T$$

$$= (w_1, w_2, \dots, w_{L-1}, w_L)^T$$

여기서 W 를 L 차원 공간의 한 점으로 생각한다면 학습을 통해 L 차원 공간상에서 일련의 입력 패턴에 대한 원하는 출력을 나타내는 점으로 W 를 이동시킬 수 있다.

출력 벡터 Y 는 X 와 W 의 합수로 나타낼 수 있다.

$$Y = F(W, X)$$

Y 를 각각의 뉴런 출력으로 나타내면 다음과 같다.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{N_1} \end{bmatrix} = \begin{bmatrix} f_1(W, X) \\ f_2(W, X) \\ f_3(W, X) \\ \vdots \\ f_{N_1}(W, X) \end{bmatrix} \quad (3)$$

지도 학습 방법에서는 신경망을 학습시키는데 있어서 입력 X 와 원하는 목표치 d 의 쌍 (X, d) 이 필요하며, 이를 학습 패턴쌍(training pattern pair)이라고 한다. 일반적으로 신경망의 활성 함수로 시그모이드 함수를 주로 사용하므로 학습 패턴쌍을 입력 X 의 클래스(class)에 속하는 출력의 목표치는 1로 나머지 출력의 목표치는 0으로 만들어줄 수 있다. 이러한 학습 패턴쌍을 고려한다면 입력 X 가 클래스 w_i ($i = 1, \dots, N_2$)에 속하면 식 (3)의 y_i 는 증가시키고 y_j 를 제외한 나머지 출력은 감소시키는 방향으로 연결 강도를 변화시켜 줄 수 있다. 이는 각 출력의 gradient를 이용하여 구할 수 있다.

$$\nabla Y = \frac{\partial Y}{\partial W} \quad (4)$$

하지만 식 (4)의 Y 는 많은 비선형 활성 함수가 포함되어 있으므로 각 출력의 gradient를 분석적으로 구하는 것은 어렵다. 그러므로 gradient를 근사화하여 다음과 같이 나타낼 수 있다.

$$\nabla y_i = \frac{\Delta y_i}{\Delta w_1} \vec{w}_1 + \frac{\Delta y_i}{\Delta w_2} \vec{w}_2 + \dots + \frac{\Delta y_i}{\Delta w_N} \vec{w}_N$$

이때 $i = 1, 2, \dots, N_2$ 이고, $\{\vec{w}_1, \vec{w}_2, \dots, \vec{w}_N\}$ 는 각각 L 차원 공간의 한 기저(basis)를 이룬다.

예를 들어 학습 패턴이 2개의 클래스 w_1, w_2 로 구성되었을 때, 입력 X 가 클래스 w_1 에 속한다면 그림 2의 (a)와 같이 연결 강도 벡터 W 를 $\alpha \nabla y_1 - \beta \nabla y_2$ 의 방향으로 이동시켜 y_1 은 증가시키고 y_2 는 감소시킬 수 있다. 반대의 경우에는 (b)와 같이 W 를 $\beta \nabla y_2 - \alpha \nabla y_1$ 의 방향으로 움직일 수 있다. 여기서 α

α 와 β 는 양의 상수이다. α 와 β 는 각 출력의 gradient 벡터의 크기를 결정하는 계수로서 출력값과 목표치의 오차의 절대 값으로 구할 수 있다. y_1 에 대한 오차가 y_2 에 대한 오차보다 크다면 y_1 의 gradient 벡터에 더 큰 값이 곱해져 W 는 ∇y_1 의 방향으로 더 큰 비중을 두며 이동한다.

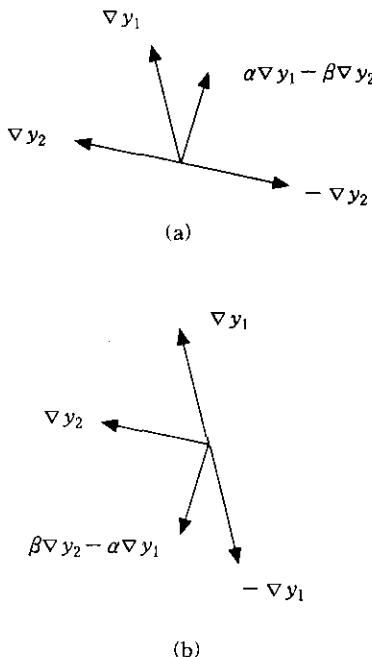


그림 2. gradient들의 합에 의한 연결 강도 변화.

- (a) 입력 X 가 클래스 w_1 일 경우.
- (b) 입력 X 가 클래스 w_2 일 경우.

학습 패턴이 P 개의 클래스로 구성된다면 연결 강도 벡터 W 는 다음과 같이 갱신할 수 있다.

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

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

γ 는 학습률(learning rate)이다.

3. 실험 결과 및 고찰

실험은 3층 신경망을 사용하고 은닉층 뉴런의 개수는 입력층 뉴런수의 2배를 사용하였다. 입력 데이터로는 리모트 센싱 시스템(LACIE) [2]에 의해 채집된 실제 데이터 중 각 클래스 당 200개를 랜덤하게 선택하

여 학습시켰고 나머지 데이터는 시험용으로 사용하였다. 본 논문의 실험 결과로는 8 차원의 8 클래스를 분리하는 문제에 대해 분석한다.

제안된 알고리즘을 위해식 (4)의 gradient 계산을 위한 Δw_i ($i = 1, 2, \dots, L$)의 크기를 결정하여야 한다. 본 논문에서는 Δw_i 의 각각에 대해 같은 크기를 사용하여 gradient를 구한다. 실험 결과 Δw_i 의 크기는 0.1 이하일 경우에 유사하게 좋은 학습 성능을 보인 반면 크기가 0.5보다 큰 값에 대해서는 성능의 저하나 일관성 없는 수렴 특성을 보였다.

그림 3은 제안된 알고리즘에 대해 Δw_i 의 크기를 0.1로 고정하였을 때, 학습률(learning rate)의 변화에 대한 학습 성능을 보여준다. 일반적으로 크기가 큰 학습률은 수렴 속도는 빠르지만 학습이 진행될수록 학습 성능의 저하가 발생하기도 한다. 제안된 알고리즘에서는 1이하의 학습률에 대해서 학습 성능의 저하는 없음을 알 수 있다. 그러므로 학습률은 0.5~1의 값을 사용하면 적합하다.

제안한 알고리즘의 비교를 위해 오류 역전파 알고리즘을 사용한다. 먼저 두 알고리즘의 차이를 확인하기 위해 오류 역전파 알고리즘으로 현재 연결 강도를 구하고 이 연결 강도로 새로운 알고리즘과 오류 역전파 알고리즘의 연결 강도 변화량을 구한다. 그림 4는 첫 번째 iteration에서의 각 학습 데이터에 대한 두 알고리즘의 연결 강도 변화량의 각도 차이를 보여준다. 그림 5는 각 iteration마다의 총 연결 강도 변화량의 각도 차이를 나타낸다. 두 알고리즘의 연결 강도가 서로 다른 방향으로 이동하고 있음을 알 수 있다.

오류 역전파 알고리즘은 학습률이 0.01, 0.05, 0.1, 0.3, 0.5일 경우에 학습하여 가장 좋은 성능을 보여주는 학습률을 선택하여 나타내었다. 제안한 알고리즘은 학습률을 1로 놓고, Δw_i 의 값만을 0.001, 0.01, 0.05, 0.1, 0.5로 변화시키며 학습시키고, 그 중에서 가장 좋은 성능을 나타내는 경우를 선택한다. 그림 6-7은 오류 역전파 알고리즘의 학습률은 0.3, 제안한 알고리즘의 Δw_i 의 값은 0.05일 때의 성능 비교를 보여 준다. 학습 데이터와 시험 데이터 모두 오류 역전파 알고리즘이 더 높은 정확도를 보여준다. iteration이 많아질수록 오류 역전파 알고리즘은 정확도의 변동이 조금씩 발생하지만 제안된 알고리즘은 일정한 증가를 보인다.

제안한 알고리즘은 은닉층의 뉴런 개수가 적은 경우에도 좋은 성능을 보였고 동일한 신경망 구조에서 클래스의 수가 많은 경우에 오류 역전파 알고리즘과 비교하여 더 큰 성능차이를 나타냈다.

4. 결론

본 논문에서는 신경망의 각 연결 강도를 L차원 벡터의 한 구성 요소로 보고 학습을 L차원 공간상에서 주어진 입력에 대한 출력이 원하는 학습 패턴상을 갖도록 하는 한 점을 찾는 과정으로 생각한다. 이러한 점을 찾기 위해 각 출력의 출력 오차가 가중된 gradient 를 구하고 구해진 gradient의 전체 합에 의해 최종 연결 강도의 이동 방향을 결정한다. 실험을 통해 기준의 오류 역전과 알고리즘에 의해 학습의 정확도에서 좋은 성능을 보여주었다.

참고문헌

- [1] S. Haykin, *Neural Networks*, New York: Macmillan, 1994.
- [2] L.L. Biehl and e. al., "A Crops and Soils Data Base For Scene Radiation Research," *Proc. Machine Process. of Remotely Sensed Data Symp.*, West Lafayette, Indiana, 1982.

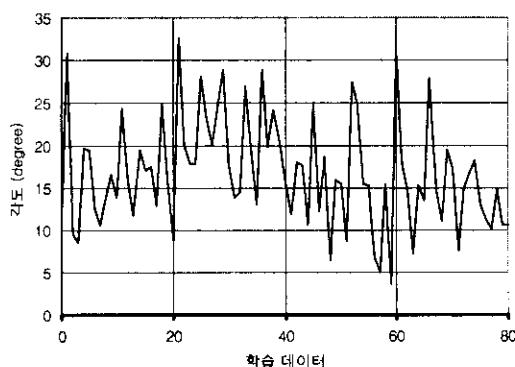


그림 4. 각 학습 데이터에 대한 각도 차이.

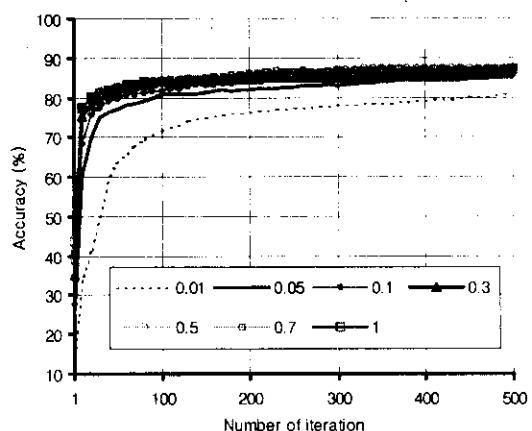


그림 3. $\Delta w_i = 1$ 일 때, 제안된 알고리즘의 다른 학습률에 대한 학습 성능 비교.

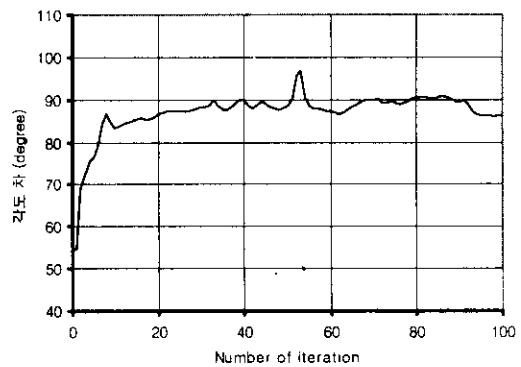


그림 5. 각 iteration에서의 각도 차이.

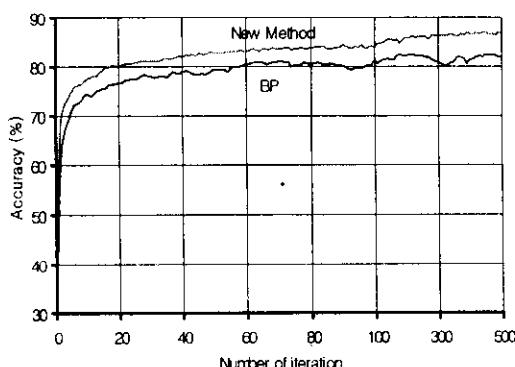


그림 6. 학습 데이터에 대한 성능 비교.

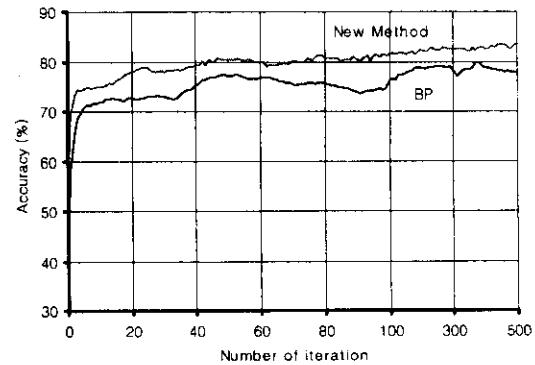


그림 7. 시험 데이터에 대한 성능 비교.