韓國컴퓨터情報學會 論文誌
第16卷 第8號, 2011. 8.

2011-16-8-1-5

상관 계수를 이용한 다층퍼셉트론의 계층별 학습

곽 영 태*

A Layer-by-Layer Learning Algorithm using Correlation Coefficient for Multilayer Perceptrons

Young-Tae Kwak*

요 약

다층퍼셉트론의 계층별 학습 방법의 하나인 Ergezinger 방법은 출력 노드가 1개로 구성되어 있고, 출력층의 가중치를 최소자승법으로 학습하기 때문에 출력층의 가중치에 조기 포화 현상이 발생할 수 있다. 이런 조기 포화 현상은 학습 시간과 수렴 속도에 장애가 된다. 따라서, 본 논문은 Ergezinger의 학습 방법을 출력층에서 벡터 형태로 학습할 수 있는 알고리즘으로 확대하고 학습 시간과 수렴 속도를 개선하기 위해서 학습 상수를 도입한다. 학습 상수는 은닉층 가중치 조정 시, 새로이 계산된 가중치와 기존 가중치의 상관 관계를 계산하여 학습 상수에 반영하는 가변적인 방법이다. 실험은 제안된 방법과 기존 방법의 비교를 위해서 iris 문제와 비선형 근사화 문제를 대상으로 실험하였다. 실험에서, 제안 방법은 기존 Ergezinger 방법보다 학습 시간과 수렴 속도에서 우수한 결과를 얻었으며, 상관 관계를 고려한 CPU time 측정에서도 제안한 방법이 기존 방법보다 약 35%의 시간을 절약할 수 있었다.

▸ Keyword : 다층퍼셉트론, 계층별 학습, 최소자승법, 상관 계수

Abstract

Ergezinger's method, one of the layer-by-layer algorithms used for multilyer perceptrons, consists of an output node and can make premature saturations in the output's weight because of using linear least squared method in the output layer. These saturations are obstacles to learning time and covergence. Therefore, this paper expands Ergezinger's method to be able to use an output vector instead of an output node and introduces a learning rate to improve learning time and convergence. The learning rate is a variable rate that reflects the

• 제1저자 : 곽영태
• 투고일 : 2011. 04. 12, 심사일 : 2011. 05. 09, 게재확정일 : 2011. 05. 11.
* 전북대학교 IT정보공학부(Division of Information Technology, Chonbuk National University)

correlation coefficient between new weight and previous weight while updating hidden's weight. To compare the proposed method with Ergezinger's method, we tested iris recognition and nonlinear approximation. It was found that the proposed method showed better results than Ergezinger's method in learning convergence. In the CPU time considering correlation coefficient computation, the proposed method saved about 35% time than the previous method.

▸ Keyword : Multilayer perceptrons, Layer-by-layer learning, Least squared method, Correlation coefficient

I. 서 론

다층퍼셉트론(MultiLayer Perceptrons)의 학습 알고리즘으로 많이 사용되는 오류역전파(Error Back Propagation) 학습은 기본적으로 오차 함수의 1차 미분을 이용한 최급 강하법을 사용하기 때문에 수렴 속도가 느리고 학습 시간이 오래 걸리는 단점이 있다[1,2]. 이런 단점을 해결하기 위해, 학습 상수의 조정, 모멘텀 항 추가, 은닉층의 선형 근사화 등 다양한 방법들이 제안되어 왔다[3,4,5,6].

오차 함수의 1차 미분 한계를 벗어나 2차 미분을 이용한 방법으로는 Gauss-Newton 방법[7], Levernberg-Marquardt 방법[8,9], Quasi-Newton 방법[7], Conjugate Gradient 방법[10,11] 등이 있다. 이런 2차 미분을 이용하는 방법은 Hessian 행렬이나 Jacobian 행렬을 계산해야 하거나, 선형 검색이나 지역 검색을 학습 알고리즘에 구현해야 한다. 비록 오차 함수의 2차 미분 계산과 검색 알고리즘에 계산 시간과 저장 공간이 많이 필요하지만, 이런 학습 방법은 학습 속도 면에서는 오류역전파 학습보다 빠르다고 알려져 있다[10].

MLP(Multilayer Perceptrons)의 또 다른 학습 방법으로는 계층별 학습이 있다[6,12,13]. 계층별 학습은 오차 함수로 먼저 출력층을 학습하고 은닉층의 오차 함수를 재 정의한 다음 은닉층을 학습한다. 은닉층의 오차 함수를 재 정의하는 방법에는 여러 가지가 있다. 본 논문은 Ergezinger[6,12]가 제안한 계층별 학습법에서 출력층의 학습 방법을 개선하고, 은닉층에 학습 상수를 도입하여 학습 속도와 계산 시간을 단축하고자 한다.

Ergezinger의 계층별 학습은 출력층을 최소자승법(Least Square Method)[6]으로 학습한다. 은닉층을 위한 오차 함수는 두 개의 항으로 정의되는데, 첫 번째 항은 은닉층을 테일러 1차 근사화한 MLP에 대한 오차 함수이며, 두 번째 항은 은닉층의 2차 미분을 반영한 항으로 구성되어 있다. 이런 두 번째 항을 페널티(penalty) 항이라고 부른다. 학습은 선형화된 MLP에서 은닉층의 가중치 변화에 대한 출력층의 변화률을 계산하여 은닉층의 가중치를 변경하는데, 이때 은닉층의 선형화가 더 이상 오차를 감소시키지 못하면 페널티 항을 강화하여 선형화된 은닉층에 비선형 특성을 반영하는 학습 방법이다.

Ergezinger가 제안한 계층별 학습은 신호 예측을 위한 알고리즘으로 출력층의 노드가 1개로 구성되어 있다. 따라서 본 논문에서는 여러 개의 출력 노드를 가질 수 있는 학습 알고리즘으로 확대한다. 그리고 은닉층의 가중치 변경시, 학습 상수를 상관 계수를 이용하여 학습 속도를 향상시키고자 한다. 학습 상수는 기존 은닉층의 가중치와 새로 계산된 가중치 사이의 상관 관계를 사용하여, 가중치의 변화를 가변적으로 적용한다.

제안된 방법과 Ergezinger의 방법에 대한 비교를 위하여 iris 문제[14]와 비선형 근사화 문제[15]를 대상으로 실험하였다. 실험 결과, 제안된 방법은 상관 계수에 대한 계산이 필요하지만 학습 시간과 수렴 속도 면에서 기존 방법보다 우수한 결과를 얻었다. 논문의 순서는 다음과 같이, 2장에서 Ergezinger의 계층별 학습법과 제안된 학습법을 설명한다. 여기서, Ergezinger의 방법을 설명하면서 본 논문에서 제안한 방법을 추가적으로 설명하고자 한다. 그리고 3장에서 실험 결과를 나타내며, 4장에서 결론을 맺는다.

II. 본론

1. Ergezinger의 계층별 학습

논문에서 사용한 MLP의 구조는 그림 1과 같이 은닉층이 하나인 MLP이며, 은닉층은 0~1의 시그모이드 함수를 사용하고 출력층은 선형 함수를 사용한다. 사용된 표기 및 기호는 다음과 같다.

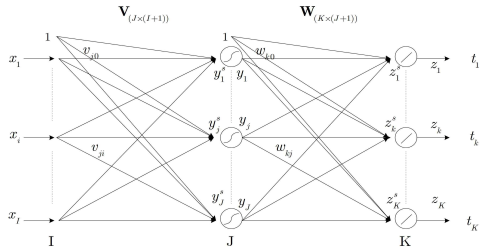

그림 1. 다층퍼셉트론의 구조
Fig. 1. Structure of MultiLayer Perceptrons

$$\mathbf{x}_p = [1, x_{1p}, x_{2p}, \cdots, x_{IP}]^T, \quad p = 1, \cdots, P$$
$$\mathbf{t}_p = [t_{1p}, t_{2p}, \cdots, t_{KP}]^T$$
$$y^s_{jp} = \sum_{i=0}^{I} v_{ji} x_{ip}, \quad \mathbf{Y}^s_{(J\times P)} = \mathbf{V}_{(J\times(I+1))}\mathbf{X}_{((I+1)\times P)}$$
$$y_{jp} = f(y^s_{jp}), \quad \mathbf{Y}_{(J\times P)} = f(\mathbf{Y}^s_{(J\times P)})$$
$$z^s_{kp} = \sum_{j=0}^{J} w_{kj} y_{jp}, \quad \mathbf{Z}^s_{(K\times P)} = \mathbf{W}_{(K\times(J+1))}\mathbf{Y}_{((J+1)\times P)} \leftarrow \text{add a bias}$$
$$z_{kp} = z^s_{kp}, \quad \mathbf{Z}_{(K\times P)} = \mathbf{Z}^s_{(K\times P)}$$

여기서, P는 학습 패턴의 수이며 t_p는 목표 패턴을 나타낸다. 또한 입력층과 은닉층의 첫 번째 원소에는 바이어스로써 1을 추가한다. MLP의 오차 함수는 식(1)과 같은 평균 제곱 오차(Mean Squared Error)를 사용한다.

$$E = \frac{1}{P}\sum_{p=1}^{P}\sum_{k=1}^{K}(t_{kp} - z_{kp})^2 = \frac{1}{P}\sum_{p=1}^{P}\mathbf{t}_p^T\mathbf{z}_p \quad \cdots\cdots (2.1)$$

우선, 출력층의 가중치(W)를 구하기 위해 식(2.2)와 같은 출력층의 오차 함수를 정의한다.

$$E^{out} = \frac{1}{P}\sum_{p=1}^{P}\sum_{k=1}^{K}(t_{kp} - z_{kp})^2 \quad \cdots\cdots (2.2)$$

Ergezinger는 출력 노드가 1개인 경우, 오차 함수의 기울기가 0이 되는 가중치를 최적의 가중치로 결정하는 LSM(Least Squared Method)을 사용하는데, 여기서 한 개의 출력 노드를 k라고 가정하고 계산하면 식(2.3)을 구할 수 있다. 하지만 본 논문에서는 출력층이 벡터로 된 경우에 적용할 수 있는 학습 알고리즘을 위하여 Ergezinger의 방법을 식(2.4)와 같이 확대 수정한다. 식(2.3)과 식(2.4)에서 행렬 A는 대칭 행렬이며 양의 정부호(positive definite) 행렬이다. 그리고 b는 열벡터이며 B는 b의 열벡터가 모인 행렬이다.

$$E^{out}_k = \frac{1}{P}\sum_{p=1}^{P}(t_{kp} - \sum_{j=0}^{J} w_{kj} y_{jp})^2$$
$$= \frac{1}{P}\sum_{p=1}^{P}(t_{kp} - \mathbf{w}_k\mathbf{y}_p)^2, \begin{cases}\mathbf{w}_k = [w_{k0}, w_{k1}, \cdots, w_{kj}, \cdots, w_{kJ}] \\ \mathbf{y}_p = [1, y_{1p}, \cdots, y_{jp}, \cdots, y_{Jp}]^T\end{cases}$$

$$\nabla\mathbf{w}_k E^{out}_k = \frac{1}{P}\sum_{p=1}^{P}(t_{kp} - \mathbf{w}_k\mathbf{y}_p)\mathbf{y}_p^T = 0$$
$$\frac{1}{P}\sum_{p=1}^{P}\mathbf{w}_k\mathbf{y}_p\mathbf{y}_p^T = \frac{1}{P}\sum_{p=1}^{P}t_{kp}\mathbf{y}_p^T \quad \leftarrow \text{transpose}$$
$$\frac{1}{P}\sum_{p=1}^{P}\mathbf{y}_p\mathbf{y}_p^T\mathbf{w}_k^T = \frac{1}{P}\sum_{p=1}^{P}t_{kp}\mathbf{y}_p$$

$$\mathbf{A}\mathbf{w}_k^T = \mathbf{b}_k, \quad \begin{cases}\mathbf{A} = \sum_{p=1}^{P}\mathbf{y}_p\mathbf{y}_p^T \\ \mathbf{b}_k = \sum_{p=1}^{P}t_{kp}\mathbf{y}_p\end{cases}$$
$$\mathbf{w}_k^T = \mathbf{A}^{-1}\mathbf{b}_k, \quad \begin{cases}\mathbf{A}_{(J+1)\times(J+1)} \\ \mathbf{b}_{k,(J+1)\times 1}\end{cases} \quad \cdots\cdots (2.3)$$

$$\mathbf{A}\mathbf{W}^T = \mathbf{B}, \quad \begin{cases}\mathbf{A} = \sum_{p=1}^{P}\mathbf{y}_p\mathbf{y}_p^T \\ \mathbf{B} = \{\mathbf{b}_1, \cdots, \mathbf{b}_k, \cdots, \mathbf{b}_K\}, \quad \mathbf{b}_k = \sum_{p=1}^{P}t_{kp}\mathbf{y}_p\end{cases}$$
$$\mathbf{W}^T = \mathbf{A}^{-1}\mathbf{B}, \quad \begin{cases}\mathbf{A}_{(J+1)\times(J+1)} \\ \mathbf{B}_{(J+1)\times K}\end{cases} .$$
$$\cdots\cdots (2.4)$$

Ergezinger는 은닉층의 가중치(V)를 계산하기 위하여, 은닉층의 가중치 변화량($\triangle v_{ji}$)에 대한 출력층의 변화량($\triangle z_{kp}$)을 구하여 최적의 가중치를 구한다. 이 과정에서 은닉층의 활성화 함수가 비선형 함수이므로 이 함수를 테일러 전개의 1차 근사화를 통해 은닉층을 식(2.7)과 같이 근사화 한다. 먼저, 식(2.5)에서처럼 $\triangle v_{ji}$를 최적의 가중치 변화량이라 하자. 그리고 이 변화량이 은닉층의 가중치 합에 적용되면, 식(2.6)과 같다. 식(2.6)의 가중치 합이 선형 근사화된 은닉층에 적용되면 식(2.7)이 되고 이런 은닉층이 출력층에서 출력되면 식(2.8)이 된다.

$$v^{new}_{ji} = v_{ji} + \Delta v_{ji}, \quad \text{Let's } \Delta v_{ji} \text{ an optimal change } \cdots$$
$$\cdots\cdots (2.5)$$

$$y_{jp}^{s,new} = y_{jp}^{s} + \Delta y_{jp}^{s}, \qquad \Delta y_{jp}^{s} = \sum_{i=0}^{I} \Delta v_{ji} x_{ip} \quad \cdots\cdots (2.6)$$

$$\begin{cases} y_{jp}^{new} = f(y_{jp}^{s}) + f'(y_{jp}^{s})\Delta y_{jp}^{s}, & j = 1,\cdots,J \\ \quad = y_{jp} + \Delta y_{jp}, & \Delta y_{jp} = f'(y_{jp}^{s}) \sum_{i=0}^{I} \Delta v_{ji} x_{ip} \\ y_{0p}^{new} = 1, & j = 0 \end{cases} \quad \cdots\cdots (2.7)$$

$$\begin{aligned} z_{kp} &= \sum_{j=0}^{J} w_{kj} y_{jp}^{new} = \sum_{j=0}^{J} w_{kj} \left(y_{jp} + \Delta y_{jp} \right) \\ &= \sum_{j=0}^{J} w_{kj} y_{jp} + \sum_{j=1}^{J} w_{kj} \Delta y_{jp} \\ &= z_{kp} + \sum_{j=1}^{J} w_{kj} \left(f'(y_{jp}^{s}) \Delta y_{jp}^{s} \right) \\ &= z_{kp} + \Delta z_{kp} \end{aligned} \quad \cdots\cdots (2.8)$$

식(2.9)는 식(2.5)의 가중치 변화량($\triangle v_{ji}$)에 대한 출력층의 변화량($\triangle z_{kp}$)를 나타낸다.

$$\begin{aligned} \frac{\partial \Delta z_{kp}}{\partial \Delta v_{ji}} &= \frac{\partial \sum_{j=1}^{J} w_{kj} f'(y_{jp}^{s}) \sum_{i=0}^{I} \Delta v_{ji} x_{ip}}{\partial \Delta v_{ji}} \\ &= w_{kj} f'(y_{jp}^{s}) x_{ip} \\ &= w_{kjp}^{lin} x_{ip}, \qquad w_{kjp}^{lin} = w_{kj} f'(y_{jp}^{s}) \end{aligned} \quad \cdots\cdots (2.9)$$

이제, 은닉층에 대한 오차 함수를 재 정의하면 식(2.10)과 같다.

$$\begin{aligned} E^{hid} &= E^{lin} + \mu E^{pen} \\ &= \frac{1}{P} \sum_{p=1}^{P} \sum_{k=1}^{K} \left((t_{kp} - z_{kp}) - \Delta z_{kp} \right)^2 + \\ &\quad \mu \frac{1}{PJK} \sum_{p=1}^{P} \sum_{k=1}^{K} \sum_{j=1}^{J} \left| \frac{1}{2} f''(y_{jp}^{s}) (\Delta y_{jp}^{s})^2 w_{kj} \right| \\ &= \begin{cases} \frac{1}{P} \sum_{p=1}^{P} \sum_{k=1}^{K} \left(d_{kp} - \Delta z_{kp} \right)^2 + \mu \frac{1}{PJK} \sum_{p=1}^{P} \sum_{k=1}^{K} \sum_{j=1}^{J} \left| \varepsilon_{jp} w_{kj} \right| \\ \varepsilon_{jp} = \frac{1}{2} f''(y_{jp}^{s}) (\Delta y_{jp}^{s})^2 \end{cases} \end{aligned} \quad \cdots\cdots (2.10)$$

여기서, E^{lin}은 선형화된 MLP의 선형 오차가 식(2.1)에서 정의된 오차 함수와 같아지도록 정의하고 있다. 즉, MLP의 선형 특성을 평가하고 있다. 그러나 MLP는 비선형 문제도 해결해야 하므로 은닉층의 오차 함수에 비선형 특성을 나타낸 것이 E^{Pen}이다. E^{Pen}는 선형 근사화에 대한 평가 척도로써 선형 오차의 영향력을 나타낸다. 학습시, MLP가 선형적인 학습을 하면, μ를 작게하여 페널티 역할을 제한하고 은닉층의 선형 변화를 확대한다. 반대로 MLP가 비선형적인 학습을 하면, μ를 크게하여 선형 오차에 대한 페널티 역할을 강화하여 선형 변화를 억제한다.

은닉층의 최적 가중치 변화량($\triangle v_{ji}$)을 구하기 위해서는 $\triangle v_{ji}$의 변화량에 대한 식(2.10)에 있는 오차 함수의 기울기가 식(2.11)과 같이 0이 되어야한다.

$$\frac{\partial E^{hid}}{\partial \Delta v_{ji}} = 0 = \frac{\partial E^{lin}}{\partial \Delta v_{ji}} + \mu \frac{\partial E^{pen}}{\partial \Delta v_{ji}} \quad \cdots\cdots (2.11)$$

식(2.11)에서 먼저 $\partial E^{lin} / \partial \triangle v_{ji}$를 계산하면 식(2.12)와 같다.

$$\begin{aligned} \frac{\partial E^{lin}}{\partial \Delta v_{ji}} &= \frac{1}{P} \sum_{p=1}^{P} \sum_{k=1}^{K} \left(\Delta z_{kp} - d_{kp} \right) \frac{\partial \Delta z_{kp}}{\partial \Delta v_{ji}} \\ &= \frac{1}{P} \sum_{p=1}^{P} \sum_{k=1}^{K} \left(\Delta z_{kp} - d_{kp} \right) w_{kjp}^{lin} x_{ip} \\ &= \frac{1}{P} \sum_{p=1}^{P} \sum_{k=1}^{K} \left(\sum_{h=1}^{H} w_{kh} f'(y_{hp}^{s}) \sum_{q=0}^{Q} \Delta v_{hq} x_{qp} - d_{kp} \right) w_{kjp}^{lin} x_{ip} \\ &= \frac{1}{P} \sum_{p=1}^{P} \sum_{k=1}^{K} \left(\sum_{h=1}^{H} w_{khp}^{lin} \sum_{q=0}^{Q} \Delta v_{hq} x_{qp} - d_{kp} \right) w_{kjp}^{lin} x_{ip} \end{aligned} \quad \cdots\cdots (2.12)$$

식(2.12)에서 첨자 h는 은닉 노드를 나타내며 q는 입력 패턴의 인덱스이다. 여기서 출력층의 노드 수와 학습 패턴의 수를 고려하고 $\sum$의 순서를 바꾸면 식(2.13)과 같다.

$$\begin{aligned} \frac{\partial E^{lin}}{\partial \Delta v_{ji}} &= \sum_{h=1}^{H} \sum_{q=0}^{Q} \Delta v_{hq} \frac{1}{P} \sum_{p=1}^{P} \sum_{k=1}^{K} w_{khp}^{lin} x_{qp} w_{kjp}^{lin} x_{ip} \\ &\quad - \frac{1}{P} \sum_{p=1}^{P} \sum_{k=1}^{K} d_{kp} w_{kjp}^{lin} x_{ip} \\ &= \sum_{h=1}^{H} \sum_{q=0}^{Q} \Delta v_{hq} a_{hq,ji} - b_{ji} \end{aligned}$$

$$a_{hq,ji} = \frac{1}{P} \sum_{p=1}^{P} \sum_{k=1}^{K} w_{khp}^{lin} x_{qp} w_{kjp}^{lin} x_{ip}, \; b_{ji} = \frac{1}{P} \sum_{p=1}^{P} \sum_{k=1}^{K} d_{kp} w_{kjp}^{lin} x_{ip} \quad \cdots\cdots (2.13)$$

$\partial E^{lin}/\partial \triangle v_{ji}$을 $a_{hq,ji}$와 b_{ji}의 계수로 나열하면 식(2.14)가 된다.

$$\frac{\partial E^{lin}}{\partial \Delta v_{ji}} = a_{10,ji}\Delta v_{10} + a_{11,ji}\Delta v_{11} + \cdots + a_{20,ji}\Delta v_{20} + \cdots + a_{HQ,ji}\Delta v_{HQ} - b_{ji} \quad \cdots\cdots (2.14)$$

다음은 식(2.11)의 $\partial E^{pen}/\partial \triangle v_{ji}$를 계산하면 식(2.15)와 같다. 여기서도 식(2.13)과 같이 출력층의 노드 수와 학습 패턴의 수를 고려하고 $\sum$의 순서를 바꾼다.

$$\begin{aligned}\frac{\partial E^{pen}}{\partial \Delta v_{ji}} &= \frac{1}{PJK}\sum_{p=1}^{P}\sum_{k=1}^{K}\sum_{j=1}^{J}\left|f^{"}(y_{jp}^{s})w_{kj}\right|(\Delta y_{jp}^{s})\frac{\partial \Delta y_{jp}^{s}}{\partial \Delta v_{ji}} \\ &= \frac{1}{PJK}\sum_{p=1}^{P}\sum_{k=1}^{K}\left|f^{"}(y_{jp}^{s})w_{kj}\right|(\sum_{q=0}^{Q}\Delta v_{jq}x_{qp})x_{ip} \\ &= \frac{1}{PJK}\sum_{q=0}^{Q}\Delta v_{jq}\sum_{p=1}^{P}\sum_{k=1}^{K}\left|f^{"}(y_{jp}^{s})w_{kj}\right|x_{ip}x_{qp} \\ &= \frac{1}{PJK}\sum_{q=0}^{Q}\Delta v_{jq}c_{q,ji}\end{aligned}$$

$$c_{q,ji} = \sum_{p=1}^{P}\sum_{k=1}^{K}\left|f^{"}(y_{jp}^{s})w_{kj}\right|x_{ip}x_{qp} \quad \cdots\cdots (2.15)$$

식(2.15)에서 $\partial E^{pen}/\partial \triangle v_{ji}$을 계수 $c_{q,ji}$로 나열하면 식(2.16)이 된다.

$$\frac{\partial E^{pen}}{\partial \Delta v_{ji}} = c_{0,ji}\Delta v_{j0} + c_{1,ji}\Delta v_{j1} + \cdots + c_{q,ji}\Delta v_{jq} + \cdots + c_{Q,ji}\Delta v_{jQ} \quad \cdots\cdots (2.16)$$

식(2.11)을 (2.13)과 식(2.15)로 나타내면 식(2.17)이 되며 식(2.17)을 다시 정리하면 식(2.18)로 표현할 수 있다. 식(2.18)에서 $c_{q,ji}$는 $j=h$인 경우에만 값을 구할 수 있으므로 식(2.18)의 $a_{hq,ji}$와 $c_{q,ji}$을 하나의 행렬로 나타내면 식(2.19)가 된다.

$$\frac{\partial E^{lin}}{\partial \Delta v_{ji}} = 0 = \sum_{h=1}^{H}\sum_{q=0}^{Q}\Delta v_{hq}a_{hq,ji} - b_{ji} + \frac{\mu}{PJK}\sum_{q=0}^{Q}\Delta v_{jq}c_{q,ji} \quad \cdots\cdots (2.17)$$

$$\sum_{h=1}^{H}\sum_{q=0}^{Q}\Delta v_{hq}a_{hq,ji} + \frac{\mu}{PJK}\sum_{q=0}^{Q}\Delta v_{jq}c_{q,ji} = b_{ji} \quad \cdots\cdots (2.18)$$

$$a_{hq,ji} = \begin{cases} a_{hq,ji} & j \neq h \\ a_{hq,ji} + \dfrac{\mu}{PJK}c_{q,ji} & j = h \end{cases} \quad \cdots\cdots (2.19)$$

식(2.19)에서 계산된 행렬 원소를 가진 행렬 $\tilde{\mathrm{A}}$와 b_{ji}로 표현된 열벡터 $\tilde{\mathrm{b}}$로 식(2.18)를 다시 나타내면 식(2.20)을 얻을 수 있으며, 식(2.20)에서 은닉층을 위한 최적의 가중치 변화량을 구할 수 있다. 여기서 $\triangle \mathrm{v}^{opt}$는 최적의 은닉층 가중치로써 가중치를 열벡터로 나타내고 있다. 그림 2는 행렬 $\tilde{\mathrm{A}}$을 식(2.14) 형태로 나타낸 것이며, 여기서 위쪽 밑줄이 있는 부분에 $c_{q,ji}$의 원소가 더해지는 부분이다.

$$\begin{aligned}&\tilde{\mathbf{A}}\Delta\mathbf{v}^{opt} = \tilde{\mathbf{b}} \qquad \begin{cases}\tilde{\mathbf{A}} = a_{hq,ji}, & \tilde{\mathbf{b}} = b_{ji}\end{cases} \\ &\Delta\mathbf{v}^{opt} = (\tilde{\mathbf{A}})^{-1}\tilde{\mathbf{b}} \qquad \begin{cases}\Delta\mathbf{v}^{opt} = (\Delta v_{10}, \Delta v_{11}, \cdots, \Delta v_{HQ})^{T}\end{cases} \quad \cdots\cdots\end{aligned} \quad \cdots\cdots (2.20)$$

$$J\times(I+1) \left\{ \begin{array}{c} \overline{a_{10,10}\Delta v_{10} + a_{11,10}\Delta v_{11} + \cdots + a_{1Q,10}\Delta v_{1Q}} + a_{20,10}\Delta v_{20} + \cdots + a_{2Q,10}\Delta v_{2Q} + \cdots + a_{HQ,10}\Delta v_{HQ} \\ \overline{a_{10,11}\Delta v_{10} + a_{11,11}\Delta v_{11} + \cdots + a_{1Q,11}\Delta v_{1Q}} + a_{20,11}\Delta v_{20} + \cdots + a_{2Q,11}\Delta v_{2Q} + \cdots + a_{HQ,11}\Delta v_{HQ} \\ \vdots \\ \overline{a_{10,1Q}\Delta v_{10} + a_{11,1Q}\Delta v_{11} + \cdots + a_{1Q,1Q}\Delta v_{1Q}} + a_{20,1Q}\Delta v_{20} + \cdots + a_{2Q,1Q}\Delta v_{2Q} + \cdots + a_{HQ,1Q}\Delta v_{HQ} \\ a_{10,20}\Delta v_{10} + a_{11,20}\Delta v_{11} + \cdots + a_{1Q,20}\Delta v_{1Q} + \overline{a_{20,20}\Delta v_{20} + \cdots + a_{2Q,20}\Delta v_{2Q}} + \cdots + a_{HQ,20}\Delta v_{HQ} \\ \vdots \\ a_{10,HQ}\Delta v_{10} + a_{11,HQ}\Delta v_{11} + \cdots + a_{1Q,HQ}\Delta v_{1Q} + a_{20,HQ}\Delta v_{20} + \cdots + a_{2Q,HQ}\Delta v_{2Q} + \overline{\cdots + a_{HQ,HQ}\Delta v_{HQ}} \end{array} \right\} H\times Q = J\times(I+1)$$

그림 2. 행렬 $\tilde{\mathrm{A}}$
Fig. 2. Matrix $\tilde{\mathrm{A}}$

지금까지는 Ergezinger가 제안한 계층별 학습에서 1개의 출력 노드를 출력 벡터로 사용할 수 있도록 학습 알고리즘을 수정 확대시켰다. 이런 Ergezinger의 방법은 출력층을 LSM으로 학습하기 때문에 출력층의 가중치에 조기 포화 현상이 일어날 수 있다. 그리고 조기 포화된 출력층의 가중치는 은닉층의 가중치 변경 시, 영향을 주어 학습 속도를 지연시킨다.

따라서, 논문은 은닉층의 가중치 변경에 대한 학습 상수를 추가적으로 제안한다. 이런 학습 상수는 식(2.21)과 같은 상관 계수의 값에 따라 학습 상수를 조정하는 가변적인 학습 상수이다. 식(2.21)에서 v^{cur}는 현재의 은닉층 가중치를 벡터로 표현한 것이며, $\overline{\triangle v_{ji}^{opt}}$와 $\overline{\triangle v_{ji}^{cur}}$은 각각 $\triangle \mathrm{v}^{opt}$의 평균과 v^{cur}의 평균을 나타낸다.

$$cf = \mathrm{corr}\left(\Delta \mathbf{v}^{opt}, \mathbf{v}^{cur}\right), \qquad v^{cur} = \mathbf{V}$$
$$= \frac{\sum_{j=1}^{J}\sum_{i=0}^{I}\left(\left(\Delta v_{ji}^{opt} - \overline{\Delta v_{ji}^{opt}}\right)\left(v_{ji}^{cur} - \overline{v_{ji}^{cur}}\right)\right)}{\sqrt{\sum_{j=1}^{J}\sum_{i=0}^{I}\left(\Delta v_{ji}^{opt} - \overline{\Delta v_{ji}^{opt}}\right)^2}\sqrt{\sum_{j=1}^{J}\sum_{i=0}^{I}\left(v_{ji}^{cur} - \overline{v_{ji}^{cur}}\right)^2}} \quad \cdots\cdots \quad (2.21)$$

식(2.21)의 상관 계수는 가중치 변경에서 다음과 같은 특징을 나타낸다. ① $cf \approx 1$: 상관 계수가 1에 가까운 경우는 새로 계산한 가중치와 기존 가중치 사이에 오차의 최소화 방향이 일치하고 있어 가중치의 조정을 더 확대할 수 있다. ② $cf \approx -1$: 상관 계수의 값이 -1에 가까운 경우는 새로운 가중치와 기존 가중치의 방향이 반대로 향하고 있어 가중치의 조정을 축소해야 한다. ③ $cf \approx 0$: 만약 상관 계수의 값이 0에 가까우면 가중치 방향들 사이에 관계가 일정하지 않으므로 가중치의 조정을 확대하거나 축소하지 않고 현 상태를 유지한다. 이와 같은 학습 상수 조정 방법은 Ergezinger의 방법에서 사용되지 않았으므로 논문에서 제안한 방법은 Ergezinger의 학습 시간과 수렴 속도를 개선할 수 있다.

2. 제안된 계층별 학습 알고리즘

1 절에서 설명한 Ergezinger의 학습법과 상관 계수를 이용한 학습 상수 조정법을 적용한 전체적인 알고리즘의 구성은 다음과 같다.

1. Initialize weights and parameters
   $\mu = 10^{-3}, \quad \gamma = 1.2, \quad \beta = 0.8$
2. Update the output-hidden weight (W) with eq. (2.4)
   Compute MSE using eq. (2.1)
3. Solve $\triangle \mathrm{v}^{opt}$ in eq. (2.20)
4. Evaluate correlation coefficient cf in eq. (2.21)
5. Compute the trial weight $(\mathrm{V}^{\mathrm{trial}})$
   $\mathrm{V}^{\mathrm{trial}} = \mathrm{V} + (1 + cf)\triangle \mathrm{v}^{\mathrm{opt}}$
6. Recompute $\mathrm{MSE}^{\mathrm{trial}}$ with (W) and $(\mathrm{V}^{\mathrm{trial}})$
7. If $\mathrm{MSE}^{\mathrm{trial}}$ < MSE then
   update the hidden-input weight $(\mathrm{V} = \mathrm{V}^{\mathrm{trial}})$
   MSE = $\mathrm{MSE}^{\mathrm{trial}}$
   $\mu = \mu \cdot \beta, \quad 0 < \beta < 1$
   go to step 9
8. else
   $\mu = \mu \cdot \gamma, \quad \gamma > 1$
   go to step 3
9. Test for completion
   If not satisfied with ending condition,
   go to step 2

III. 실험 결과

제안한 학습 알고리즘과 Ergezinger의 학습 방법을 비교 실험하기 위하여 iris 문제[14]와 비선형 근사화 문제[15]를 대상으로 실험하였다. 또한 각 알고리즘은 모두 Matlab으로 구현하였으며, 학습을 평가하는 오차 함수는 식(2.1)과 같은 MSE(Mean Squared Error) 함수를 사용하였다.

우선, iris 문제는 꽃받침과 꽃잎의 모양에 따른 4개의 관측값을 입력하여 3종류의 iris로 분류하는 문제이다. 따라서 MLP는 입력 노드 4개, 출력 노드 3개, 그리고 은닉 노드는 5, 7, 9개로 변경하면서 학습하였다. 학습에 사용된 iris 관측값은 각 iris 종류별로 50개씩 총 150개를 사용하였고, 학습의 종료 조건으로는 최대 epoch가 300에 이르던지, MSE가 0.1이하가 될 때까지 학습하였다. 제안한 방법과 Ergezinger 방법의 동등한 비교를 위하여 초기 가중치를 동일하게 설정하였다.

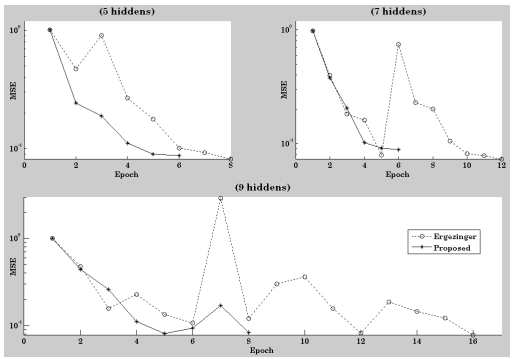

그림 3. Iris 학습 결과
Fig. 3. Learning results of Iris

그림 3은 각 은닉 노드의 수의 변경에 따른 iris의 학습 결과이다. 여기서, x축은 학습 시간을 epoch 단위로 나타내며 y축은 학습 MSE를 지수 비율로 나타내고 있다. 그림에서 알 수 있듯이, Ergezinger의 방법은 학습 시간이 제안된 방법보다 많이 소요되며, 학습 하는 동안 많은 오차의 진동이 발생되고 있다. 이에 반하여, 제안된 방법은 Ergezinger의 방법보다 오차의 진동이 적고 수렴 속도도 빨랐다. 그림 3은 2.2절에서 제안된 방법에 따라 실험한 결과이다. 따라서 학습 시간에 MSE^{trial}이 포함되어 있지 않다. 그러나 그림 4는 각 epoch의 계산에 MSE^{trial}도 포함시킨 결과이다. 그림 4의 결과에서도 그림 3의 결과와 마찬가지로 학습 속도와 오차의 진동 횟수 면에서 제안된 방법이 Ergezinger의 방법보다 우수한 결과를 얻었다. 특히, 은닉 노드가 5개인 경우는 기존 방법보다 대략 5배 정도의 학습 속도의 개선이 있었다.

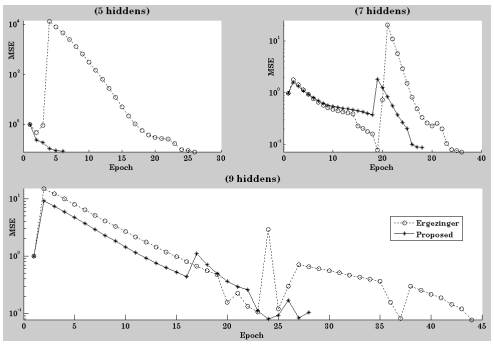

그림 4. Iris 학습 결과 (MSE^{trial} 포함)
Fig. 4. Learning results of Iris (including MSE^{trial})

두 번째 실험으로, 비선형 근사화 문제는 식(3.1)과 같이 여덟 개의 입력값을 세 개의 출력값으로 출력하는 함수 근사화 문제로써 입력값과 출력값의 범위는 0~1의 값을 가지며 총 50개의 샘플을 학습 패턴으로 사용하였다. 또한 은닉 노드의 수를 5, 10, 15개로 변경하면서 학습하였다. 학습 조건은 iris 문제와 동일하게 적용하였고, 좀 더 최소화된 오차에 대한 두 방법의 비교를 위하여 목표 MSE를 0.001로 사용하였다.

$$t_1 = \frac{x_1 \cdot x_2 + x_3 \cdot x_4 + x_5 \cdot x_6 + x_7 \cdot x_8}{4}$$

$$t_2 = \frac{x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8}{8} \qquad \cdots\cdots\cdots\cdots (3.1)$$

$$t_3 = \sqrt{1 - t_1}$$

그림 5는 2.2절의 MSE^{trial}을 epoch의 횟수에 포함시키지 않은 결과로써, 은닉 노드의 수에 관계없이 제안한 방법과 Ergezinger의 방법이 비슷한 결과를 얻었다. 즉, 학습 시간과 수렴 속도 면에서 유사한 결과를 얻었다. 하지만, MSE^{trial}을 포함하는 그림 6의 결과를 보면 학습 시간과 수렴 속도 면에서 제안한 방법이 Ergezinger의 방법보다 월등한 결과를 보여주고 있다.

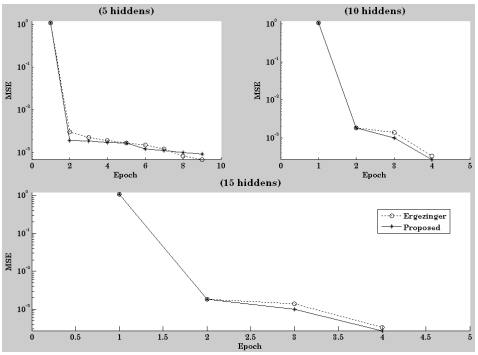

그림 5. 비선형 근사화 학습 결과
Fig. 5. Learning results of nonlinear approximation

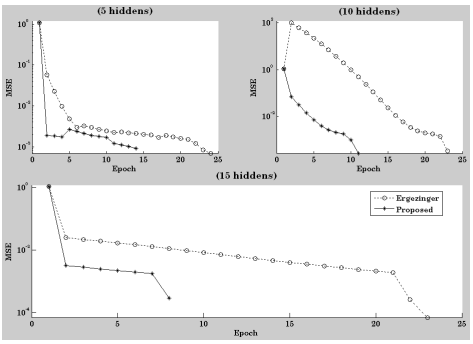

그림 6. 비선형 근사화 학습 결과 (MSE^{trial} 포함)
Fig. 6. Learning results of nonlinear approximation (including MSE^{trial})

제안된 방법은 기존 Ergezinger의 방법보다 상관 계수를 계산해야 하는 추가적인 시간이 필요하다. 표 1은 제안된 방법과 기존 방법의 CPU time을 초 단위로 측정한 것이다. 표 1에서 알 수 있듯이, 제안된 방법이 전체적으로 기존 방법보다 CPU time이 적게 소모되었다. 제안된 방법은 기존 방법보다 iris 문제에서는 33%, 비선형 근사화 문제에서는 35%의 CPU time을 절약할 수 있었다. 따라서, 논문은 제안된 방법이 기존 Ergezinger의 방법보다 CPU time은 적게 걸리면서 학습 시간과 수렴 속도면에서 더 좋은 결과를 얻을 수 있음을 실험을 통해 확인하였다.

표 1. CPU 소요 시간
Table 1. CPU time required

(단위:sec)

Iris 문제	은닉 노드의 수			평균 시간
	5	7	9	
Ergezin.	1.45	2.38	2.98	2.27
제안 방법	0.53	1.89	2.16	1.53
비선형 근사화	은닉 노드의 수			평균 시간
	5	10	15	
Ergezin.	1.86	1.72	2.22	1.93
제안 방법	1.31	1.19	1.27	1.26

IV. 결 론

MLP의 계층별 학습에 사용되는 Ergezinger 학습은 출력 노드가 1 개인 경우로 제한되어 있고 학습 상수를 사용하지 않으므로 학습 시간이 오래 걸리고 학습 오차의 진동이 발생할 수 있다. 따라서, 논문에서는 출력층이 벡터 형태로 학습할 수 있도록 Ergezinger의 학습 방법을 확대하였고, 은닉층 가중치 사이의 상관 계수를 학습 상수로 사용함으로써 학습 시간과 수렴 속도를 개선할 수 있는 알고리즘을 제안하였다.

제안한 학습 상수는 은닉층의 새로이 계산된 가중치와 기존 가중치 사이의 상관 계수에 따라 조정되는데, 상관 계수가 1에 근사하면 가중치의 조정을 확대하고, -1에 근사하면 축소하며, 0에 근사한 값을 보이면 현재의 가중치를 유지한다.

Iris 문제와 비선형 근사화 문제를 대상으로 하는 실험에서,제안된 방법은 기존 Ergezinger의 학습법보다 학습 시간과 수렴 속도에서 우수한 결과를 얻었다. 또한, CPU time을 고려한 학습 결과에서도 제안한 방법은 기존 방법보다 약 35%의 CPU time을 절약할 수 있었다.

참고문헌

[1] D. E. Rumelhart, and J. L. McClelland, Parallel Distributed Processing, MIT Press, Cambridge, MA, pp. 318-362, 1986.

[2] Kyong Ho Lee, “A Study on the Implementation of Serious Game Learning Multiplication Table using Back Propagation Neural Network on Divided Interconnection Weights Table,” Journal of The Korea Society of Computer and Information, Vol. 14, No. 10, pp. 233-240, Oct. 2009.

[3] T. P. Vogal, J. K. Mangis, A. K. Zigler, W. T. Zink ,and D. L. Alkon, "Accelerating the convergence of the backpropa gation method," Biological Cybernetics, Vol. 59, pp. 256-264, Sept. 1988.

[4] T. Tollenaere, "SuperSAB: Fast adaptive back propa gation with good scaling properties," Neural Networks, Vol. 3, No. 5, pp. 561-573, 1990.

[5] M. Kordos and W. Duch, "Variable step search algorit hm for feedforward networks," Neurocomputing, Vol. 71, pp. 2470-2480, April 2008.

[6] S. Ergezinger, and E. Thomsen, "An accelerated lear ning algorithm for multilayer perceptrons optimizati on Layer by Layer," IEEE Trans. on Neural Networks, Vol. 6, No. 1, pp. 31-42, June 1995.

[7] Rolnald E. Miller, "Optimization," John Wiley & Son, INC. pp. 358-362, 2000.

[8] M. T. Hagan, and M. Menhaj, "Training feedforward net works with the Marquardt algorithm," IEEE Trans. on Neural Networks, Vol. 5, No. 6, pp. 989-993, Nov. 1994.

[9] Young-Tae Kwak, “Accelerating Levenberg- Marqua rdt Algorithm using Variable Damping Parameter,” Journal of The Korea Society of Computer and Information, Vol. 15, No. 4, pp. 57-63, April 2010.

[10] C. Charalambous, "Conjugate gradient algorithm for efficient training of artificial neural networks," IEEE Proceedings, Vol. 139, No. 3, pp. 301-310, 1992.

[11] Wei Chu, Chong Jin Ong, and Keerthi S.S., "An imp roved conjugate gradient scheme to the solution of least squares

SVM," IEEE Trans. on Neural Networks, Vol. 16, No. 2, pp. 498-501, March 2005.

[12] B. Ph. van Milligen, V. Tribaldos, J. A. Jimenez, an d C. Santa Cruz, ''Comments on "An accelerated learning algorithm for multilayer perceptrons optimization layer by layer"," IEEE Trans. on Neural Networks, Vol. 9, No. 2, pp. 339-341, March 1998.

[13] Jim Y. F. Yam, and Tommy W. S. Chow, ''Extended least squares based algorithms for training feedforward networks," IEEE Trans. on Neural Networks, Vol. 8, No. 3, pp. 806-810, May 1997.

[14] UCI Machine Learning Repository
http://archive.ics.uci.edu/ml/

[15] F. Biegler-Kong, and F. Barmann, "A learning algo rithm for multilayered neural networks based on linear squares problems," Neural Networks, Vol. 6, pp. 127-131, 1993.

저 자 소 개

곽 영 태

1993년 충남대학교 컴퓨터공학과 공학사
1995년 충남대학교 컴퓨터공학과 공학석사
2001년 충남대학교 컴퓨터공학과 공학박사
2001년 - 2002년 한국전자통신연구원 선임연구원
2002년 3월 - 현재 전북대학교 IT정보공학부 부교수
관심분야 : 신경회로망, 패턴인식, 영상처리
Email : ytkwak@jbnu.ac.kr