
다층퍼셉트론의 오류역전파 학습과 계층별 학습의 비교 분석

곽영태*

Comparative Analysis on Error Back Propagation Learning and Layer By Layer Learning in Multi Layer Perceptrons

Young-Tae Kwak*

요 약

본 논문은 MLP의 학습 방법으로 사용되는 EBP학습, Cross Entropy함수, 계층별 학습을 소개하고, 필기체 숫자인식 문제를 대상으로 각 학습 방법의 장단점을 비교한다. 실험 결과, EBP학습은 학습 초기에 학습 속도가 다른 학습 방법에 비해 느리지만, 일반화 성능이 좋다. 또한, EBP학습의 단점을 보완한 Cross Entropy함수는 학습 속도가 EBP학습보다 빠르다. 그러나, 출력층의 오차 신호가 목표 벡터에 대해 선형적으로 학습하기 때문에, 일반화 성능이 EBP학습보다 낮다. 그리고, 계층별 학습은 학습 초기에, 학습 속도가 가장 빠르다. 그러나, 일정한 시간 후, 더 이상 학습이 진행되지 않기 때문에, 일반화 성능이 가장 낮은 결과를 얻었다. 따라서, 본 논문은 MLP를 응용하고자 할 때, 학습 방법의 선택 기준을 제시한다.

ABSTRACT

This paper surveys the EBP(Error Back Propagation) learning, the Cross Entropy function and the LBL(Layer By Layer) learning, which are used for learning the MLP(Multi Layer Perceptrons). We compare the merits and demerits of each learning method in the handwritten digit recognition. Although the speed of EBP learning is slower than other learning methods in the initial learning process, its generalization capability is better. Also, the speed of Cross Entropy function that makes up for the weak points of EBP learning is faster than that of EBP learning. But its generalization capability is worse because the error signal of the output layer trains the target vector linearly. The speed of LBL learning is the fastest speed among the other learning methods in the initial learning process. However, it can't train for more after a certain time, it has the lowest generalization capability. Therefore, this paper proposes the standard of selecting the learning method when we apply the MLP.

키워드

다층퍼셉트론, 오류역전파 학습, Cross Entropy함수, 계층별 학습

1. 서론

패턴 인식, 연상 기억, 비선형 함수의 근사화, 최적화 등에 응용되고 있는 신경회로망은 수학적 인 모델링이나 해결을 위한 구체적인 지식이 없어

도 학습 기능에 의해 문제를 해결할 수 있기 때문에 연구의 관심이 되고 있다. 그 중, 학습 알고리즘이 용이하고 구현이 쉽기 때문에 여러 응용 분야에서 이용되고 있는 다층퍼셉트론(Multi Layer Perceptron)은 외부의 입력을 받아들이는 입력층

* 익산대학 컴퓨터학과
접수일자 : 2003. 7. 18

과 신경회로망의 출력을 내는 출력층 그리고 입력층과 출력층 사이에 있는 은닉층으로 구성되어 있다[1,2].

이런 MLP를 학습하기 위해, 오차 함수를 가중치로 1차 미분하는 경사 강하법(Gradient Descent Method)을 이용하는 오류역전파(Error Back Propagation) 학습은 간단하고, 구현이 쉬운 장점이 있다. 그러나, 학습시간이 오래 걸리고, 지역 최소점에 빠질 수 있는 단점이 있다[3,4]. 이런 EBP의 학습 속도를 개선하기 위해 여러 가지 방법이 제안되어 왔다.

첫 번째, 학습 상수와 활성화 함수의 기울기를 조정하는 방법이 있다[5,6,7]. 학습 상수를 조정하는 방법은 학습하는 동안 오차가 계속 감소할 경우, 학습 상수를 증가시켜 학습 속도를 가속화시킨다. 반면에, 오차가 증감을 반복하는 경우, 학습 상수를 감소시켜 가중치 공간에서 오차의 최소 위치를 찾아가는 방법이다. 또한, 활성화 함수의 기울기를 조정하는 방법은 오차 함수를 활성화 함수의 기울기로 미분하여, 기울기를 조정하는 것이다. 즉, 활성화 함수의 기울기도 하나의 가중치로 생각하여, 학습하는 과정에서 조정한다. 이런 방법들은 어느 정도 학습 속도를 단축시킬 수 있지만, 기본적으로 경사 강하법의 한계를 극복하지는 못한다.

두 번째, 새로운 오차 함수를 사용하는 방법은 오차 함수를 평균제곱 오차(Mean Squared Error) 함수 대신 Cross Entropy 함수를 사용한다[8]. 이 방법은 출력노드의 오차 신호가 목표값에 선형적으로 변하므로, 학습 속도는 빠르다. 그러나, 학습 말기에 오차 신호가 너무 강하게 작용하여, EBP 학습보다 일반화 성능이 낮은 단점이 있다.

세 번째, 계층별(Layer By Layer) 학습이 있다[9,10]. 계층별 학습은 출력층의 가중치를 목표 벡터와 은닉층의 출력 벡터를 이용하여 최소 제곱법(Least Squares Method)으로 구한다. 또한, 은닉층의 목표 벡터도 최소 제곱법으로 구한 후, 은닉층의 MSE 함수를 정의하고, 최소 제곱법으로 은닉층의 가중치를 구한다. 이런 계층별 학습은 최소 제곱법을 사용하기 때문에, 학습 속도가 빠르다. 그러나, 은닉층의 목표 벡터가 선형 분리가 불

가능한 경우, 일정한 시간 후에도 더 이상 학습이 진행되지 않는 단점이 있다[11].

이외에도, 최근에는 오차 함수의 2차 미분을 이용한 CGM(Conjugate Gradient Method)과 뉴턴 방법 등이 이용되고 있다[12,13]. CGM은 가중치 공간상에서 오차 함수에 대한 현재의 경사 벡터를 이전의 가중치 변경 벡터들과 직교하도록 하여, 가중치 공간 차원만큼 가중치를 변경한다. 하지만, CGM은 가중치 변경 벡터의 크기를 결정하는데, 많은 계산량이 필요하다. 또한, 뉴턴 방법은 현재의 가중치의 경사도에 Hessian 역행렬을 곱하여 가중치를 변경한다. 뉴턴 방법도 Hessian 행렬과 Hessian 역행렬의 계산에 많은 시간이 필요하다. 이러한 2차 미분을 사용하는 방법은 EBP 학습보다 학습 속도는 빠르지만, 계산량이 많은 단점이 있다.

따라서, 본 논문은 MLP의 학습 방법으로 사용되는 EBP 학습, 오차 함수로 Cross Entropy 함수를 사용하는 방법, 계층별 학습 등을 소개하고, 실험을 통하여 각 학습 방법의 장단점을 비교 분석한다. 또한, MLP의 응용에 적합한 학습 방법을 선택하는 기준을 제시한다.

논문의 구성으로, II장에서는 EBP 학습, Cross Entropy 함수와 계층별 학습에 대해 간단히 소개한다. 그리고, III장의 실험에는 각 학습 방법의 확인을 위해서 입력 문자에 상당한 왜곡과 변형이 많은 필기체 숫자(CEDAR Database)[14]를 대상으로 실험한 결과가 있다. 실험 결과, 학습 속도는 계층별 학습, Cross Entropy 함수, EBP 학습 순으로 빨랐으며, 일반화 성능은 EBP 학습, Cross Entropy 함수, 계층별 학습 순으로 우수했다. IV장의 결론은 실험 결과를 분석하여 각 학습 방법의 장단점 및 응용 방법을 제시한다.

II. 본 론

1. MLP와 EBP 학습

대표적인 전방향 신경회로망의 모델 중에서, 많이 사용되는 모델이 MLP이다. 그리고, 본 논문에서 사용한 MLP는 그림 1과 같은 은닉층이 하나

있는 2층 퍼셉트론 구조이다. 이런 2층 퍼셉트론은 일반적으로 2층 이상의 MLP보다 지역 최소점에 빠질 확률이 낮고, 일반화 성능이 좋다[15,16]. 또한, 2층 퍼셉트론은 어떠한 형태의 사상도 나타낼 수 있는 일반적인 근사화 함수임이 증명되어 있다[17].

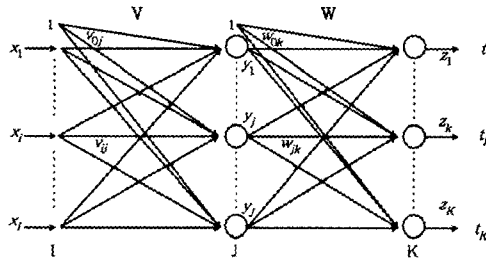


그림 1. MLP의 구조
Fig. 1 Architecture of MLP

그림 1에서, 입력 노드(\$x_i\$)는 단지 입력 패턴을 전달만 하며, 노드의 역할에 따라 은닉 노드(\$y_j\$)와 출력 노드(\$z_k\$)로 나누어진다. 은닉 노드는 입력값에 가중치를 취하여 합한 뒤에, 그 결과를 시그모이드 함수로 출력한다. 그리고, 출력 노드는 은닉층의 출력값을 대상으로 은닉 노드와 유사한 일을 수행한다. 여기서, 입력 노드의 수와 출력 노드의 수는 학습 패턴의 차원과 목표값의 차원에 따라 결정되지만, 은닉 노드의 수는 학습할 문제의 복잡도에 따라 다르다. 이러한 MLP의 구조는 비선형 분리 함수(nonlinearly separable function)와 같은 복잡한 기능을 수행할 수 있다[18].

본 논문에서 전체적으로 사용할 표기법은 다음과 같다.

입력층 벡터: $\mathbf{x}_p = [1, x_{p1}, \dots, x_{pi}, \dots, x_{pI}]^T$
 은닉층 벡터: $\mathbf{y}_p = [1, y_{p1}, \dots, y_{pj}, \dots, y_{pJ}]^T$
 출력층 벡터: $\mathbf{z}_p = [z_{p1}, \dots, z_{pk}, \dots, z_{pK}]^T$
 목표 벡터: $\mathbf{t}_p = [t_{p1}, \dots, t_{pk}, \dots, t_{pK}]^T$

여기서, 각 벡터는 열벡터이고, \$p\$는 학습 패턴의 인덱스이며, 각 벡터의 첫 번째 요소 중 1은 바이어스를 위한 입력값이다.

전체 학습 패턴에 대한 각 출력 벡터를 행렬로

표시하면, 입력 패턴, 은닉층의 출력 패턴, 출력층의 출력 패턴, 목표 패턴은 다음과 같다.

$$\begin{aligned} \mathbf{X} &= [\mathbf{x}_1, \dots, \mathbf{x}_p, \dots, \mathbf{x}_P] \in R^{(I+1) \times P} \\ \mathbf{Y} &= [\mathbf{y}_1, \dots, \mathbf{y}_p, \dots, \mathbf{y}_P] \in R^{(J+1) \times P} \\ \mathbf{Z} &= [\mathbf{z}_1, \dots, \mathbf{z}_p, \dots, \mathbf{z}_P] \in R^{K \times P} \\ \mathbf{T} &= [\mathbf{t}_1, \dots, \mathbf{t}_p, \dots, \mathbf{t}_P] \in R^{K \times P} \end{aligned}$$

또한, \$j\$번째 은닉 노드로 연결된 가중치 벡터는 $\mathbf{v}_j = [v_{0j}, \dots, v_{ij}, \dots, v_{Ij}]^T$ 이며, \$k\$번째 출력 노드에 연결된 가중치 벡터는 $\mathbf{w}_k = [w_{0k}, \dots, w_{jk}, \dots, w_{Jk}]^T$ 이다. 가중치 벡터도 열벡터이며, \$v_{0j}\$와 \$w_{0k}\$는 각 노드의 바이어스이다.

이런 가중치 벡터를 행렬로 나타내면,

$$\begin{aligned} \mathbf{V} &= [\mathbf{v}_1, \dots, \mathbf{v}_j, \dots, \mathbf{v}_J] \in R^{(I+1) \times J} \\ \mathbf{W} &= [\mathbf{w}_1, \dots, \mathbf{w}_k, \dots, \mathbf{w}_K] \in R^{(J+1) \times K} \end{aligned}$$

EBP학습에서 사용하는 오차 함수는 식(1)과 같은 MSE함수이다. 이 오차 함수는 각 출력 노드의 출력값과 그 노드에 대한 목표값의 오차를 제공해서 모두 더한다. 식(1)에서 \$P\$는 전체 학습 패턴의 수이고, \$K\$는 출력 노드의 수이다.

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

경사 강하법을 사용하는 EBP학습은 두 단계로 나눌 수 있다. 우선, 가중치를 임의로 초기화 하고 학습 패턴과 목표 벡터를 지정해 준 다음, 식(2)와 식(3)과 같이, 순방향 연산으로 출력 벡터를 계산한다. 여기서, \$f(\cdot)\$는 -1~1의 시그모이드 함수이다.

$$net_{pi} = \sum_{j=1}^J v_{ij} x_{pi} + v_{0i}, \quad y_{pi} = f(net_{pi}) \quad (2)$$

$$net_{pk} = \sum_{j=1}^J w_{jk} y_{pj} + w_{0k}, \quad z_{pk} = f(net_{pk}) \quad (3)$$

$v_{0j}, w_{0k} : j, k$ 번째 노드의 바이어스

$$\begin{aligned} \Delta w_{jk} &= -\frac{\partial E}{\partial w_{jk}} & (4) \\ &= -\frac{\partial E}{\partial z_{pk}} \frac{\partial z_{pk}}{\partial net_{pk}} \frac{\partial net_{pk}}{\partial w_{jk}} \\ &= (t_{pk} - z_{pk}) f'(net_{pk}) y_{pj} \\ w_{jk}(t+1) &= w_{jk}(t) + \eta \Delta w_{jk} \end{aligned}$$

$$\begin{aligned} \Delta v_{ij} &= -\frac{\partial E}{\partial v_{ij}} & (5) \\ &= -\frac{\partial E}{\partial y_{pj}} \frac{\partial y_{pj}}{\partial net_{pj}} \frac{\partial net_{pj}}{\partial v_{ij}} \\ &= \sum_{k=1}^K \delta_{pk} w_{jk} f'(net_{pj}) x_{pi} \\ v_{ij}(t+1) &= v_{ij}(t) + \eta \Delta v_{ij} \end{aligned}$$

그 다음, 역방향 연산으로 가중치 공간에서 오차가 감소하도록 가중치를 수정하는 과정을 반복한다. 출력층의 오차는 식(4)처럼, 목표 벡터와 출력 벡터의 값을 이용한다. 그리고, 오차값을 알 수 없는 은닉층의 경우는 식(5)처럼 chain rule을 이용하여, 출력층의 오차값을 역전파하고, 이 오차에 따라 가중치의 변화량을 구한다[3].

식(4)와 식(5)는 하나의 학습 패턴(p)에 대한 가중치 조정이다. 또한, E는 식(1)의 오차 함수이고, $v_{ij}(t)$ 와 $w_{jk}(t)$ 는 현재의 가중치이며, $v_{ij}(t+1)$ 와 $w_{jk}(t+1)$ 는 새로 조정되는 가중치이다. η 는 학습 상수이고, δ_{pk} 와 δ_{pj} 는 각각 출력 노드와 은닉 노드의 오차 신호이다. 이런 오차 신호에는 시그모이드 함수의 미분이 포함되어 있어, 현재의 학습 패턴에 대한 오차는 크지만, 전체 오차가 충분히 작으면, 현재의 오차를 감소시키지 않고 오차 신호를 약화시킨다.

2. Cross Entropy 함수

Ooyen와 Nienhuis는 EBP학습 과정 중에, 출력 노드의 부적절한 포화가 학습을 지연시키는 원인이 되고, 이런 포화는 출력 노드의 오차 신호에 포함되어 가중치의 변화량을 작게 한다 것에 착안했다. 따라서, 출력 노드의 오차 신호에서 시그모이드 함수의 기울기 향이 제거되는 식(6)과 같은 Cross Entropy 함수를 MSE함수 대신 사용한다 [8]. 이 방법은 출력 노드의 오차 신호가 선형적으로 학습하여, 학습 속도는 빠르다. 그러나, 학습

말기에도 오차신호가 너무 강하게 작용하여, EBP 학습보다 일반화 성능이 낮은 단점이 있다.

$$E = -\sum_{p=1}^P \sum_{k=1}^K [(1+t_{pk}) \ln(1+z_{pk}) + (1-t_{pk}) \ln(1-z_{pk})] \quad (6)$$

$$\Delta w_{jk} = -\frac{\partial E}{\partial w_{jk}} = (t_{pk} - z_{pk}) y_{pj} = \delta_{pk} y_{pj}$$

3. 계층별 학습

Wang과 Chen은 MLP의 계층별 학습을 제안하였다[9,10]. 우선, 전체 학습 패턴에 대한 은닉층 출력 행렬(Y)을 구한다. 그리고, 하나의 학습 패턴(q)에 대한 최적의 출력층 가중치(\mathbf{W}_q^*)를 식(7)과 같이 최소 제곱법으로 구한다. 그리고, 구한 출력층 가중치로 은닉층 목표 벡터(\mathbf{y}_q^*)를 식(8)처럼 최소 제곱법으로 구한다. 이 때, 구한 은닉층의 목표 벡터가 활성화 함수의 범위를 넘어가면 절삭한다. 이런 은닉층 목표 벡터를 이용하여 은닉층의 MSE함수를 정의하고, 정의된 MSE함수를 이용하여 은닉층 가중치(\mathbf{V}^*)를 다시 최소 제곱법으로 식(9)처럼 구한다.

$$\mathbf{W}_q^* = \left(\sum_{p=1}^{P-1} \mathbf{y}_p \mathbf{y}_p^T \right)^{-1} \left(\sum_{p=1}^{P-1} \mathbf{y}_p \mathbf{d}_p^T \right) \quad (7)$$

$$\mathbf{y}_q^* = (\mathbf{W}_q^*)^+ \mathbf{d}_p \quad (8)$$

$$\mathbf{V}^* = \left(\sum_{p=1}^P \mathbf{x}_p \mathbf{x}_p^T \right)^{-1} \left(\sum_{p=1}^P \mathbf{x}_p \mathbf{s}_p^T \right) \quad (9)$$

여기서, 식(7)~식(9)는 최소 제곱법으로 의사역(pseudoinverse) 행렬을 이용한다[19]. 식(7)과 식(8)의 $\mathbf{d}_p = [f^{-1}(t_{p1}), \dots, f^{-1}(t_{pk}), \dots, f^{-1}(t_{pK})]^T$ 는 시그모이드 역함수에 목표 벡터를 입력한 값이고, $\mathbf{s}_p = [f^{-1}(y_{p1}^*), \dots, f^{-1}(y_{pj}^*), \dots, f^{-1}(y_{pJ}^*)]^T$ 는 식(8)에서 구한 은닉층 목표 벡터를 시그모이드 역함수에 입력한 것이다.

계층별 학습의 최소 제곱법은 가중치의 조정을 오차에 대해 선형적으로 조정한다. 따라서, 학습 속도는 빠르다. 그러나, 은닉층의 목표 벡터가 활성화 함수의 범위를 벗어나는 경우, 그 값을 절삭하는 변환 함수를 사용한다. 이런 변환 함수는 은닉층 목표 벡터를 선형 분리가 불가능한 패턴으로

만들기 때문에, 일정한 학습 시간 후, 더 이상 학습이 진행되지 않는 현상이 나타난다[11]. 이것은 계층별 학습의 일반화 성능을 떨어지게 한다. 그리고, 학습 정보가 출력층의 가중치에 편중되는 조기 포화 현상이 일어난다.

III. 실험

본 논문의 실험은 필기체 숫자 인식 문제를 대상으로 실험하였다. 필기체 숫자는 CEDAR Database[14]의 숫자 중, 각 숫자에 대해 100개씩, 모두 1,000개를 학습에 사용하였고, 시험 패턴은 각 숫자에 대해 50개씩, 500개의 숫자를 이용하였다. 한 숫자의 영상은 12x12 픽셀로 구성되며, 각 픽셀은 16 레벨의 그레이 이미지로 바꾼 후, [-1~+1]로 정규화 한 다음, MLP에 입력시켰다. 그림 2는 학습에 사용된 패턴을 그레이 이미지로 나타낸 것이다.

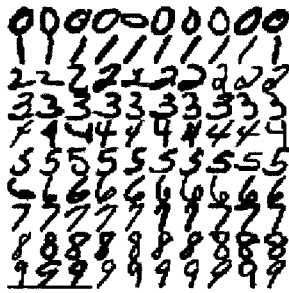


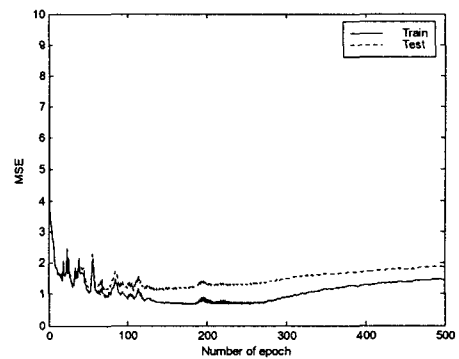
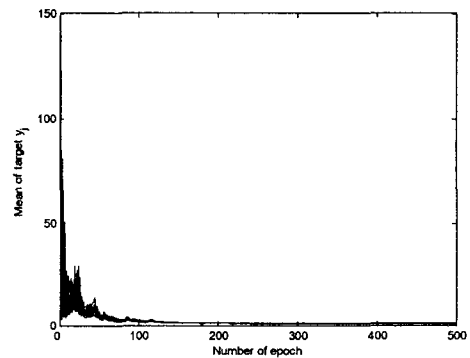
그림 2. 학습 패턴
Fig. 2 Learning Patterns

실험을 위한 MLP는 입력 노드 144개, 출력 노드 10개로 구성되고, 은닉 노드의 수는 10개와 20개로 구성하였다. 그리고, 각 학습 방법은 은닉 노드의 수에 따라, 가중치의 초기화를 3회씩 다르게 한 후, 학습 결과를 식(10)과 같은 MSE함수로 표현했다. 이렇게 표현된 학습 결과는 3회에 대한 평균값으로 그 은닉 노드의 대표 학습 결과를 나타냈다.

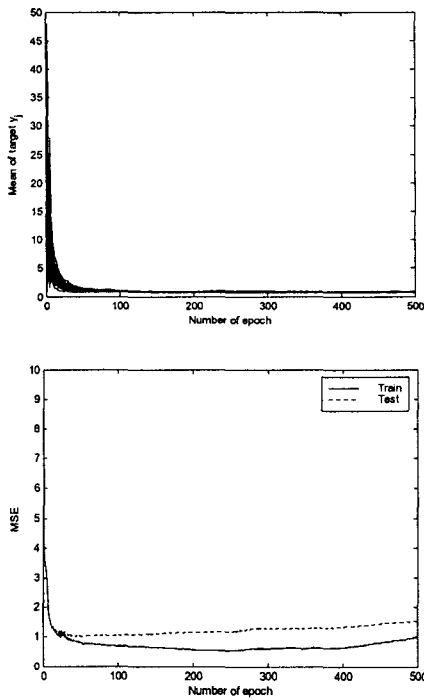
$$MSE = \frac{1}{P} \sum_{p=1}^P \sum_{k=1}^K (t_{pk} - z_{pk})^2 \quad (10)$$

학습의 완료 조건은 학습 시간을 500 epoch로 했다. 그리고, 학습 상수는 모두 0.001로 했으며, 각 학습 방법은 C로 구현하고, 의사역 행렬에 대한 계산은 MATLAB을 결합시켜 구현했다.

먼저, 계층별 학습을 분석해 보자. 계층별 학습에서 전체 학습 패턴에 대한 은닉층의 목표 벡터가 시그모이드 함수의 범위 내에 존재한다면, 한번의 최소 제곱법으로 가중치를 구하여 학습을 종료할 수 있다. 이런 경우는 MLP가 필요 없으며, 단층퍼셉트론만으로도 학습이 가능하다. 그러나, 대부분의 응용에서 학습 패턴은 비선형 문제이며 이런 응용에서 계층별 학습은 은닉층의 목표 벡터가 선형 분리되지 않으면, 더 이상의 오차를 축소할 수 없는 선형 학습이 된다.



(a) 10개의 은닉 노드



(b) 20개의 은닉 노드

그림 3. 은닉층 목표 벡터와 오차 함수
Fig. 3 Target vector of hidden layer and error function

그림 3은 각 은닉 노드에 대해 학습이 진행됨에 따라, 은닉층 목표 벡터와 학습(시험) 패턴에 대한 MSE를 나타낸다. 여기서, 은닉층 목표 벡터는 각 epoch에서 전체 학습 패턴에 대한 은닉층 목표 벡터의 절대값 평균으로 식(11)과 같다.

$$y_{축} = \frac{1}{P} \sum_{j=1}^P |y_{mj}^*| \quad (11)$$

각 그림에서, 은닉 노드의 출력이 작은 학습 초기에는 은닉층 목표 벡터가 시그모이드 함수를 벗어난다. 이것을 절삭하여 은닉층의 가중치를 조정하는 방법으로 학습을 계속 진행하면, 은닉층의 목표 벡터가 시그모이드 함수의 범위 내에 근사하게 된다. 그러나, 이렇게 근사화된 은닉층 목표 벡터가 선형 분리가 불가능하면, 더 이상 학습이 진행되지 않는 현상이 나타난다.

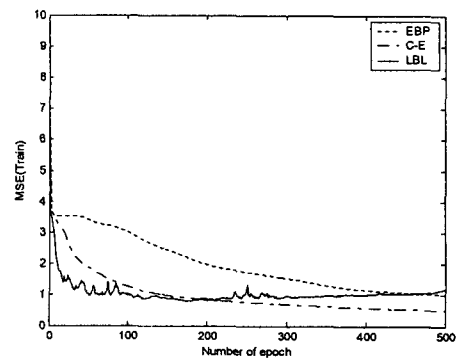
또한, 은닉 노드의 수가 증가할수록 은닉층 공간은 커진다. 이것은 은닉층 목표 벡터가 선형적으로 분포

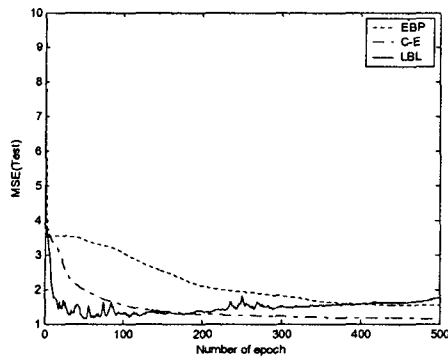
할 수 있는 가능성이 크기 때문에, 수렴 속도가 빠르다. 즉, 은닉 노드가 10개인 (a)의 경우는 작은 은닉층 공간에서 최적의 은닉층 목표 벡터를 찾기 위해, 진동이 발생한다. 그러나, 은닉 노드가 20개인 (b)의 경우는 은닉층 공간이 크기 때문에, 은닉층 목표 벡터를 쉽게 찾을 수 있다.

그림 4는 EBP학습, Cross Entropy함수, 계층별 학습의 학습 결과로서, 학습 패턴과 시험 패턴에 대한 오차 함수의 변화를 나타낸다. 여기서, x축은 학습 시간의 단위인 epoch이며, y축은 식(10)에서 정의한 MSE 함수이다. 그리고, 'LBL'은 계층별 학습을 나타내고, 'C-E'는 Cross Entropy함수를 나타낸다.

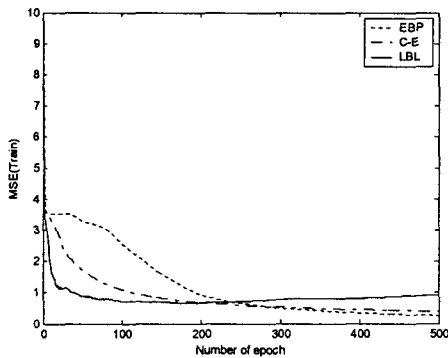
먼저, EBP학습은 은닉 노드의 수가 증가할수록, 학습 속도와 일반화 성능이 우수하다. 그러나, 은닉 노드의 수가 적은 경우, 학습이 제대로 이루어지지 않는다. 또한, 학습 초기에 학습 속도가 다른 학습 방법에 비해 느리다. EBP학습의 단점을 보완한 Cross Entropy함수는 학습 속도가 EBP학습보다 빠르다. 그러나, 은닉 노드의 수가 충분한 경우, 출력층의 오차 신호가 목표 벡터에 대해 선형적으로 학습하기 때문에, 일반화 성능이 EBP학습보다 낮다.

계층별 학습은 학습 초기에, 학습 속도가 가장 빠르다. 그러나, 일정한 시간 후, 더 이상 학습이 진행되지 않는다. 따라서, 수렴성이나 일반화 성능이 가장 낮다. 계층별 학습의 학습 속도가 빠른 원인은 출력층의 최소 제곱법에서 선형적인 학습을 하기 때문이다. 또한, 학습이 더 이상 진행되지 않는 것은 은닉층의 목표 벡터가 선형 분리를 할 수 없기 때문이다. 즉, 계층별 학습은 은닉층 목표 벡터가 선형 분리가 불가능할 때까지, 빠르게 선형적으로 학습한다.





(a) 10개의 은닉 노드



(b) 20개의 은닉 노드

그림 4. 각 학습 방법의 오차 함수 비교

Fig. 4 Comparison on the error function of each learning method

IV. 결론

본 논문은 MLP의 학습 방법으로 사용되는

EBP학습, Cross Entropy함수, 계층별 학습의 학습 방법을 소개하고, 필기체 숫자인식 문제를 대상으로 각 학습 방법의 장단점을 비교하였다.

학습 결과, EBP학습은 학습 초기에 학습 속도가 다른 학습 방법에 비해 느리지만, 일반화 성능은 좋은 결과를 얻었다. 또한, EBP학습의 단점을 보완한 Cross Entropy함수는 학습 속도가 EBP학습보다 빠르다. 그러나, 출력층의 오차 신호가 목표 벡터에 대해 선형적으로 학습하기 때문에, 일반화 성능이 EBP학습보다 낮다. 그리고, 계층별 학습은 학습 초기에, 학습 속도가 가장 빠르다. 그러나, 일정한 시간 후, 더 이상 학습이 진행되지 않는다. 따라서, 수렴성이나 일반화 성능이 가장 낮은 결과를 얻었다.

따라서, MLP을 응용하고자 할 때, MLP의 크기가 작고 학습 패턴의 수가 적은 경우는 좋은 일반화 성능을 위하여, EBP학습, Cross Entropy함수, 계층별 학습 순으로 적합하다. 그리고, MLP의 크기가 크고 학습 패턴의 수가 충분한 경우는 학습 속도를 위하여 계층별 학습, Cross Entropy함수, EBP학습 순으로 적합하다.

참고 문헌

- [1] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing*, MIT Press, Cambridge, MA, pp. 318-362, 1986.
- [2] R. P. Lippmann, "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine*, vol. 4, no. 2, pp. 4-22, April 1987.
- [3] J. M. Zurada, *Introduction to Artificial Neural Systems*, West Publishing Co., 1992.
- [4] Simon Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan College Publishing Co., 1994
- [5] J. R. Chen and P. Mars, "Stepsize variation methods for accelerating the backpropagation algorithm," *Proc. IJCNN* Jan. 15-19, 1990, Washington, DC, USA, vol. I, pp. 601-604
- [6] Ali Rezgui and Nazif Tepedelenlioglu, "The effect of the slope of the activation function on

- the back propagation algorithm," *Proceeding of IJCNN'90* Washington D.C., vol. 1, pp. 707-710
- [7] Plagianakos, V.P., Magoulas, G.D., Vrahatis, M.N., "Deterministic nonmonotone strategies for effective training of multilayer perceptrons," *IEEE Trans. Neural Networks*, vol. 13, pp. 1268-1284, 2002.
- [8] A. Van Ooyen and B. Nienhuis, "Improving the convergence of the back-propagation algorithm," *Neural Networks*, vol. 78, pp. 465-471, 1992.
- [9] G.-J. Wang and C.-C. Chen, "A Fast Multilayer Neural-Network Training Algorithm Based on the Layer-By-Layer Optimizing Procedures," *IEEE Trans. Neural Networks*, vol. 7, pp. 768-775, May, 1996.
- [10] Lengelle, R., and Denoeux, T., "Training MLPs Layer by Layer Using an Objective Function for Internal Representations," *Neural Networks*, vol. 9, January, 1996.
- [11] Jim. Y. F. Yam and Tommy W. S. Chow, "Extended Least Squares Based Algorithm for Training Feedforward Networks," *IEEE Trans. Neural Networks*, vol. 8, pp. 806-810, May, 1997.
- [12] C. M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1997.
- [13] Ampazis, N., Perantonis, S.J., "Two highly efficient second-order algorithms for training feedforward networks," *IEEE Trans. Neural Networks*, vol. 13, pp. 1064-1074, Sep., 2002.
- [14] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern and Machine Intell.*, vol. 16, pp. 550-554, 1994.
- [15] J. Villiers and E. Barnard, "Backpropagation Neural Nets with One and Two Hidden Layers," *IEEE Trans. Neural Networks*, vol. 4, no. 1, pp. 136-141, 1993.
- [16] M M Islam and K Murase, "A new algorithm to design compact two-hidden-layer artificial neural networks," *Neural Networks*, vol. 14, 2001.
- [17] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359-366, 1989.
- [18] Shah, J.V., Chi-Sang Poon, "Linear independence of internal representations in multilayer perceptrons," *IEEE Trans. Neural Networks*, vol. 10, no. 1, pp. 10-18, 1999.
- [19] David J. Winter, *Matrix Algebra*, Macmillan Publishing Company, 1992.

저자 소개



곽영태(Young-Tae Kwak)

1993년 충남대학교 컴퓨터공학과 공학사, 1995년 충남대학교 컴퓨터공학과 공학 석사, 2001년 충남대학교 컴퓨터공학과 공학 박사.

※ 관심분야 : Neural Networks, 컴퓨터 비전, 영상 처리