논문 99-24-9B-21 한국통신학회논문지 '99-9 Vol.24 No.9B

은닉층 다차원공간의 Vertex를 이용한 MLP의 은닉 노드 축소방법

정회원 곽 영 태*, 이 영 직**, 권 오 석*

Reducing the Number of Hidden Nodes in MLP using the Vertex of Hidden Layer's Hypercube

Young-Tae Kwak*, Young-Gik Lee**, Oh-Seok Kwon* *Regular Members*

요 약

본 논문은 학습하는 동안 은닉 노드의 출력에 대한 분산과 평균을 평가하는 새로운 cost function을 이용하여 불필요한 은닉 노드를 축소하는 방법을 제안한다. 제안한 cost function은 필요한 은닉 노드를 활성화 시키고 불필요한 은닉 노드를 상수화 시켜 제거한다. 필기체 숫자인식을 통한 실험에서 제안한 방법은 높은 인식률과 단축된 학습 시간을 나타내며 은닉 노드의 수를 37.2%까지 축소할 수 있었다.

ABSTRACT

This paper proposes a method of removing unnecessary hidden nodes by a new cost function that evaluates the variance and the mean of hidden node outputs during training. The proposed cost function makes necessary hidden nodes be activated and unnecessary hidden nodes be constants. We can remove the constant hidden nodes without performance degradation. Using the CEDAR handwritten digit recognition, we have shown that the proposed method can remove the number of hidden nodes up to 37.2%, with higher recognition rate and shorter learning time.

I. 서 론

신경회로망의 MLP(MultiLayer Perceptrons)는 학습 가능에 의해 임의의 비선형 함수를 근사화 할 수 있기 때문에 패턴인식, 최적화, 비선형 제어, 음성 인식등에 응용되고 있다. 이런 MLP의 학습으로 널리 사용되고 있는 EBP(Error Back Propagation) 학습은 gradient descent방법을 이용하기 때문에 학습 시간이 오래 걸리고 응용에 따라 필요한 은닉 노드의 수를 결정해야 하는 단점이 있다"[1].

EBP학습의 속도를 개선하는 방법은 MSE(Mean Squared Error)함수 대신 entropy함수"[2]를 사용하거나 학습 진행에 따라 학습률을 변경시키는 방법"[3]이 있다. 그리고 MEBP(Modified Error Back Propagation)학습은 학습시 출력층의 부적절한 포화를 방지하기 위해 MSE함수와 entropy함수를 사용한다[4]. 최근에는 2차 미분을 이용한 CGM(Conjugate Gradient Method)과 뉴톤 방법(Newton's method)[5]등이 이용되고 있다.

EBP학습은 은닉 노드수가 충분하다면 임의의 함수를 제한된 오차 내에 근사화 시킬 수 있다[6]. 그러나 은닉 노드수가 너무 많으면 계산량이 많아지며 학습 패턴의 미세한 부분까지 학습하여 MLP의 일반화 성능을 저하시킨다. 반대로 은닉 노드수가 적으면 학습 패턴을 학습할 수 없다[7]. 따라서 본

* 충남대학교 컴퓨터공학과({oskwon, ytkwak}@comeng.chungnam.ac.kr),
** 한국전자통신연구원 멀티모달 I/F팀(ylee@etri.re.kr)
논문번호 : 99053-0218, 접수일자 : 1999년 2월 18일
※ 이 논문은 1997년 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음

논문은 학습시 필요한 은닉 노드 수를 결정하는 MLP의 구조결정 문제를 해결하는 방법을 제시한다.

MLP의 구조결정 문제를 해결하는 방법은 크게 두 가지로 분류된다. 첫째는 cascade correlation learning[8]이나 structure level adaptation[9]등과 같이 초기에 은닉층 및 노드의 수를 작게 한 다음 점차적으로 추가하는 방법이다. 그러나 cascade correlation learning은 은닉층의 증가로 출력층의 fan in이 증가하여 하드웨어 구현이 어려운 단점이 있다. 둘째는 초기에 충분한 은닉 노드로 학습한 다음 불필요한 가중치 및 은닉 노드를 축소하는 방법이 있다[10].

축소 방법은 가중치의 수를 축소하는 방법과 은닉 노드의 수를 축소하는 방법이 있다. 가중치를 축소하는 방법으로, Karnin[10,11]은 학습 후 가중치의 변화량에 대한 오차의 변화량을 구하여 그 값이 작은 가중치를 제거하였다. 이 방법은 EBP학습을 그대로 사용하는 장점이 있으나 가중치 만큼의 추가적인 기억 장소가 필요하다. 또한, Le Cun[10,12]은 오차 함수를 각 가중치로 2차 미분하고 Hessian행렬을 구하여 그 값이 작은 가중치를 제거하였는데, 이 방법은 미분과 행렬 계산으로 인하여 학습 속도가 느려진다.

은닉 노드를 축소하는 방법으로, Hagiwara[13]는 각 은닉 노드마다 전체 학습 패턴에 대하여 역전파된 오차값을 모두 더하여 'badness factor'를 계산하고 가장 큰 노드를 제거하였으나, 각 은닉 노드에 연결된 가중치의 초기화와 재학습이 필요하다. Kruschke[10,14]는 가중치 벡터의 내적을 이용하여 중복된 은닉 노드를 제거하는 방법을 제안하였다. 또한, Chauvin[10,15]은 은닉 노드의 출력이 작은 노드를 상대적으로 더 작도록 학습하여 은닉 노드의 출력이 작은 노드를 제거했다. 그리고 Mozer와 Smolensky[10,16]는 은닉 노드의 출력에 대한 오차 함수의 미분으로 은닉 노드의 중요도를 측정하여 가장 작은 은닉 노드를 제거하는 방법을 제안하였다. 이와 같은 은닉 노드를 축소하는 방법은 가중치를 축소하는 방법보다 MLP의 구조결정 문제를 정확하게 해결하며, 하드웨어 구현 시 효율적이다.

본 논문에서는 은닉 노드를 축소하는 방법으로 오차 함수를 수정한 새로운 cost function을 이용하여 학습이 진행됨에 따라 불필요한 은닉 노드를 점차적으로 제거하는 방법을 제시한다. 제Ⅱ장에서는 기존 은닉 노드를 축소하는 Mozer와 Smolensky의 방법[10,16], Chauvin의 방법[10,15] 그리고 Kruschke의 방법[10,14]을 간단히 소개한다. 또한 MLP의 학습으로 사용한 MEBP학습을 간단히 설명한다[4]. MEBP 학습시 은닉층의 역할을 알아보고 은닉층 공간에서 학습 패턴들의 분포를 이용하여 새로운 cost function을 제안한다. 제안한 cost function은 MEBP학습의 오차 함수와 은닉 노드의 출력에 대한 분산과 평균을 평가하는 함수를 포함한다. 또한 cost function을 이용한 가중치 조정을 batch학습과 standard학습으로 나누어 설명한다.

제Ⅲ장의 실험에서는 제안한 알고리즘의 확인을 위해서 입력 문자에 상당한 왜곡과 변형이 많은 필기체 숫자(CEDAR database)[17]를 대상으로 실험했다. 필기체 숫자는 Mozer와 Smolensky의 방법[10,16], Chauvin의 방법[10,15] 그리고 제안한 방법으로 학습하여 그 결과를 비교했다. 실험 결과, 제안한 방법은 높은 인식률로 학습 시간을 단축시키면서 초기 은닉 노드의 37.2%까지 축소할 수 있었다. 이 결과는 다른 두 방법에 비해 은닉 노드의 축소률, 인식률, 학습 시간면에서 보다 좋은 결과이다. 제Ⅳ장의 결론에서는 실험의 결과를 분석하여 제안한 방법의 장점 및 응용성을 제시한다.

Ⅱ. 본 론

기존 은닉 노드의 축소 방법을 설명하고 은닉 노드의 역할을 분석하기 위해 본 논문에서 사용한 MLP는 그림 1과 같이 은닉층이 하나 있는 two layer perceptrons구조이다. 이런 two layer perceptrons은 일반적으로 two layer이상의 MLP보다 local minima에 빠질 확률이 낮고 일반화 성능이 좋다[18].

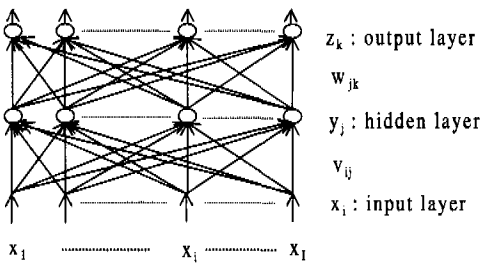


그림 1. MLP의 구조

우선, MLP의 학습을 위해 본 논문에서 사용한 MEBP학습을 간략하게 설명하면 다음과 같다[4]. MLP는 입력 노드(x_i=1…I), 은닉 노드(y_j=1…J), 출력 노드(z_k=1…K)로 구성되고 가중치는 입력층과 은닉층 사이에 v_{ij}와 은닉층과 출력층 사이에 w_{jk}로

되어있다고 하자. 학습은 두 단계로 나눌 수 있으며 출력값을 계산하는 단계

$$y_{pj}=f\left(\sum_{i=1}^{I} v_{ij}x_{pi}+v_{0j}\right) \quad z_{pk}=f\left(\sum_{j=1}^{J} w_{jk}y_{pj}+w_{0k}\right) \quad (1)$$

v_{0j}, w_{0k} : i, k번째 노드의 바이어스

와 오류를 역전파하여 가중치를 조정하는 단계이다.

$$E^{MEBP}=\begin{cases}\frac{1}{2}\sum_{p=1}^{P}\sum_{k=1}^{K}(t_{pk}-z_{pk})^2 & \text{if } k\in|t_{pk}-z_{pk}|<1\\ -\sum_{p=1}^{P}\sum_{k=1}^{K}[(1+t_{pk})\ln(1+z_{pk}) & \text{otherwise}\\ \quad +(1-t_{pk})\ln(1-z_{pk})] & \end{cases} \quad (2)$$

$$\Delta w_{jk}^{MEBP}=-\eta\frac{\delta E^{MEBP}}{\delta w_{jk}}=\eta\sum_{p=1}^{P}\delta_{pk}y_{pj} \quad (3)$$

$$\delta_{pk}=\begin{cases}(t_{pk}-z_{pk})z'_{pk} & \text{if } k\in|t_{pk}-z_{pk}|<1\\ (t_{pk}-z_{pk}) & \text{otherwise}\end{cases}$$

$$\Delta v_{ij}^{MEBP}=-\eta\frac{\delta E^{MEBP}}{\delta v_{ij}}=\eta\sum_{p=1}^{P}\sum_{k=1}^{K}\delta_{pk}w_{jk}y'_{pj}x_{pi}$$

여기서 x_{pi}는 p번째 학습 패턴에 대한 i번째 입력을 나타내고 t_{pk}는 p번째 학습 패턴에 대한 k번째 출력 노드의 목표값(target value)을 나타낸다. 이러한 노드의 출력을 계산하는 활성화 함수는 식(4)과 같은 -1~1범위의 시그모이드 함수를 사용한다. 시그모이드 함수는 기울기 값에 따라 빠른 학습 속도를 보이는 활성화 영역과 학습 속도가 느린 포화 영역으로 나눈다[19].

$$f(net)=\frac{2}{1+e^{-\lambda net}}-1 \quad (4)$$

활성화영역: $|net|\le 3.64$, $|f(net)|\le 0.948$
포화영역: $|net|>3.64$, $|f(net)|>0.948$

1. 기존 은닉 노드 축소방법

1) Mozer와 Smolensky의 방법[10,16]

Mozer와 Smolensky는 은닉 노드의 제거 전과 후에 대한 오차를 이용한다. 오차의 근사적인 계산을 위해 식(5)과 같이 은닉 노드의 출력에 gating term(α_j)을 사용한다.

$$z_k=f\left(\sum_{j=1}^{J} w_{jk}\alpha_j y_j\right) \quad (5)$$

$$\rho_j=-\frac{\delta E}{\delta \alpha_j}\Big|_{\alpha_j=1} \quad \rho_j(t+1)=\rho_j(t)+\rho_j \quad (6)$$

오차에 대한 gating term(α_j)의 미분은 은닉 노드의 출력이 오차를 얼마나 감소시키는가를 나타낸다. α_j의 값은 학습이 진행됨에 따라 증가한다. 따라서 α_j의 값이 크면 필요한 은닉 노드이고 α_j의 값이 상대적으로 작으면 불필요한 은닉 노드가 되어 제거된다. 이 방법은 오차의 감소가 작은 은닉 노드를 제거할 때 그 노드가 학습한 오차의 정보도 동시에 제거되는 문제점이 있다.

2) Chauvin의 방법[10,15]

Chauvin은 은닉 노드의 출력을 이용한다. 즉 식(7)과 같은 cost function을 정의하여 은닉 노드의 출력이 큰 노드는 현 상태의 출력을 유지하도록 가중치를 조정한다. 반대로 은닉 노드의 출력이 작은 노드는 그 출력이 더 작아지도록 가중치를 조정하여 제거하는 방법이다. 여기서 식(7)의 함수$e(y_{pj}^2)$는 식(8)과 같은 단조증가 함수이다. Chauvin의 방법은 은닉 노드의 출력은 크지만 분산이 작아 활동성이 낮은 은닉 노드를 제거하지 못하는 단점이 있다.

$$C=\mu_{er}\sum_{p=1}^{P}\sum_{k=1}^{K}(t_{pk}-z_{pk})^2+\mu_{en}\sum_{p=1}^{P}\sum_{j=1}^{J}e(y_{pj}^2) \quad (7)$$

$$e=\frac{y_{pj}^2}{1+y_{pj}^2} \qquad e'=\frac{1}{(1+y_{pj}^2)^2} \quad (8)$$

3) Kruschke의 방법[10,14]

Kruschke는 가중치 벡터를 이용하여 중복된 은닉 노드를 제거한다. 중복되는 은닉 노드는 가중치 벡터가 평행한 경우이다. 가중치 벡터의 중복에 대한 결정은 식(9)과 같은 벡터의 내적을 이용한다. g_j^s의 값이 작은 은닉 노드는 다른 은닉 노드에 비해 중복된 성질을 많이 포함하고 있으므로, g_j^s의 값이 0에 가까운 노드를 제거하는 방법이다. 여기서 s는 학습 패턴에 대한 인덱스이다. 그리고 r은 양의 실수이다. 이 방법은 학습 패턴이 대칭으로 존재하는 경우 은닉 노드를 부적절하게 축소하며, 상수 r의 값에 따라 은닉 노드의 축소률이 변하는 단점이 있다.

$$\Delta g_j^s=-r\sum_{j\neq i\in J}\cos^2\angle(v_i^s,v_j^s)\cdot g_i^s \quad (9)$$

2. 은닉층의 역할

MLP가 MEBP학습을 하는 동안 은닉층이 학습 패턴을 어떻게 분류하고 어떤 노드가 불필요한 은

닉 노드인지 알아보자. 우선, MEBP학습이 진행됨에 따라 은닉 노드의 출력과 가중치의 크기가 증가함을 알아 보자.

MEBP학습에서 오차를 감소시키는 gradient descent방법에 의해 식(10)이 유도된다.

$$
\begin{aligned}
&E^{MEBP}(t) \geq E^{MEBP}(t+1)\\
&|t_k(t)-z_k(t)| \geq |t_k(t+1)-z_k(t+1)| \quad t_k(t) = t_k(t+1)\\
&|z_k(t)| \leq |z_k(t+1)|\\
&\left|\sum_{j=1}^{J} w_{jk}(t)y_j(t) + w_{0k}(t)\right| \leq \\
&\qquad \left|\sum_{j=1}^{J} w_{jk}(t+1)y_j(t+1) + w_{0k}(t+1)\right|
\end{aligned}
\tag{10}
$$

식(10)을 가중치 벡터와 은닉 노드의 출력 벡터의 내적으로 표현하면 식(11)이다.

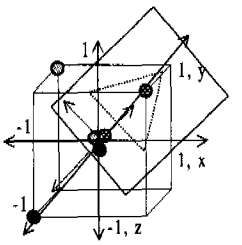
$$
\begin{aligned}
&|\mathbf{w}_{jk}(t)|\,|\mathbf{y}_j(t)|\,|\cos\theta(t)| \leq \\
&\qquad |\mathbf{w}_{jk}(t+1)|\,|\mathbf{y}_j(t+1)|\,|\cos\theta(t+1)|
\end{aligned}
\tag{11}
$$

식(11)에서 하나의 패턴이 학습하는 방향이 일정하다고 가정하여 $|\cos\theta|$를 간소화하면 식(12)이다. 그리고 이 부등식에 대한 해는 식(13)이 된다. 따라서 MEBP학습은 은닉 노드의 출력을 시그모이드 함수의 극값(-1,1)에 근접시킴을 알 수 있다. 이런 MEBP학습의 특성은 gradient descent방법을 사용하는 EBP학습에서도 동일하다.

$$
\begin{aligned}
&|\mathbf{w}_{jk}(t)|\,|\mathbf{y}_j(t)| \leq \\
&\qquad |\mathbf{w}_{jk}(t+1)|\,|\mathbf{y}_j(t+1)|
\end{aligned}
\tag{12}
$$

$$
\begin{cases}
|\mathbf{w}_{jk}(t)| \leq |\mathbf{w}_{jk}(t+1)|, & |\mathbf{y}_j(t)| \leq |\mathbf{y}_j(t+1)| \\
|\mathbf{w}_{jk}(t)| = |\mathbf{w}_{jk}(t+1)|, & |\mathbf{y}_j(t)| \leq |\mathbf{y}_j(t+1)| \\
|\mathbf{w}_{jk}(t)| \leq |\mathbf{w}_{jk}(t+1)|, & |\mathbf{y}_j(t)| = |\mathbf{y}_j(t+1)|
\end{cases}
\tag{13}
$$

은닉층의 역할을 다차원공간에서 설명하면 다음과 같다. MLP의 은닉층은 J개의 은닉 노드(축)로 이루어진 다차원공간을 형성하고 2^J개의 vertex를 가지고 있다. 학습 초기시 학습 패턴은 그림 2와 같이 다차원공간의 중앙(원점)에 위치한다. 그러나 학습이 진행됨에 따라 은닉 노드의 출력이 커지므로 학습 패턴은 vertex쪽으로 이동한다. 즉 원점에 대한 유클리드 거리가 커진다. 이렇게 vertex쪽으로 이동된 학습 패턴은 출력층의 다차원평면에 의해 분리된다. 즉, 은닉층은 출력층의 다차원평면이 학습 패턴을 분리할 수 있도록 각 학습 패턴을 은닉층 공간의 각 vertex쪽으로 비선형 이동시키는 역할을 한다.



출력 노드의 3차원평면

그림 2. 3차원 은닉층 공간과 출력 노드의 3 차원평면

그림 2는 3개의 은닉 노드(x, y, z축)으로 이루어진 3차원 은닉층 공간에서 학습 초기시 원점에 존재하는 학습 패턴들이 MEBP학습에 의해 각 vertex쪽으로 이동하는 것을 나타낸다. 또한 출력층의 3차원평면이 각 verex쪽으로 이동된 학습 패턴을 분할하고 있다.

은닉층 다차원공간에서 은닉 노드의 수가 증가할수록, 전체 vertex의 수는 기하급수적으로 증가하지만 사용되는 vertex의 수는 산술적으로 증가한다. 이것은 은닉 노드의 수가 증가할수록 사용되지 않는 vertex가 늘어나고 불필요한 은닉 노드가 발생할 확률이 높다.

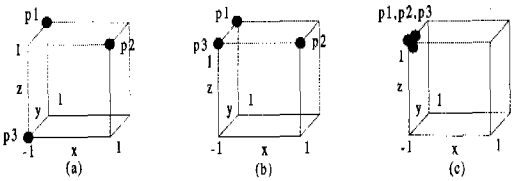


그림 3. 3차원 은닉층 공간에서 패턴의 분포

MEBP학습에 의해 학습 패턴의 분포가 그림 3과 같이 3차원 은닉층 공간에 분포한 경우를 고려해 보자. (a)는 각 학습 패턴들이 x, y, z축에 균일하게 분포하여 어떤 은닉 노드도 제거할 수 없다. 반면에 (b)는 학습 패턴들이 z축에 독립적으로 분포하고 있다. 즉 z축은 모든 학습 패턴에 대해 일정한 출력만을 생성하므로 z축을 제거해도 MLP의 분할 기능을 유지할 수 있다. 따라서 z축을 제거하고 z축의 평균(상수값)을 출력층의 바이어스에 적용할 수 있다. 또한 (c)는 하나의 vertex에 모든 패턴이 존재하는 경우로 xy축, yz축, xz축 중 하나를 제거할 수 있다.

그림 3의 (b)나 (c)와 같은 불필요한 은닉 노드는 모든 학습 패턴에 대해 은닉 노드의 출력이 일정하다. 이것은 은닉 노드의 출력에 대한 분산이 작고,

평균이 활성화 함수의 포화 영역에 속함을 의미한다. 반면, 필요한 은닉 노드는 학습 패턴이 은닉 노드의 양 극값(-1,1)쪽으로 양분되어 분산이 크고 평균이 활성화 함수의 중간값에 근사 한다.

따라서 제안한 방법은 오차 함수에 은닉 노드의 출력에 대한 분산과 평균을 평가하는 함수를 추가한다. 필요한 은닉 노드는 그 노드의 출력에 대한 분산이 크고, 평균이 시그모이드 함수의 중간값에 근사하도록 가중치를 조정한다. 반대로 불필요한 은닉 노드는 그 노드의 출력에 대한 분산이 작고, 평균이 시그모이드 함수의 포화 영역에 속하도록 학습한다. 불필요한 은닉 노드는 그 노드의 평균을 출력층의 바이어스에 weighted summation한 후 제거한다. 이것은 제거되는 은닉 노드가 학습한 정보를 소멸시키지 않고 다른 노드에 전달함을 의미한다.

3. 은닉 노드 축소 알고리즘

은닉 노드 축소 알고리즘을 위하여 은닉 노드의 출력에 대한 평균과 분산은 식(14)과 식(15)이다. 이런 은닉 노드의 분산과 평균을 이용하여 (불)필요한 은닉 노드를 구분하는 함수G(yj)를 식(16)과 같이 정의한다. 함수G(yj)는 전체 학습 패턴에 대한 은닉 노드의 출력이 vertex에서 얼마나 떨어져 분포하는가를 나타낸다. 식(16)의 G(yj)에서 mid는 시그모이드 함수의 중간값이다. 그리고 식(17)은 본 논문에서 사용한 -1~1 시그모이드 함수에 대한 G(yj)이다. 그림 4는 분산과 평균에 대한 함수G(yj)를 나타낸다.

$$M(y_j)=\frac{1}{P}\sum_{p=1}^{P}y_{pj} \qquad -1\le M(y_j)\le 1 \tag{14}$$

$$S(y_j)=\frac{1}{P}\sum_{p=1}^{P}(y_{pj})^2-M^2(y_j) \qquad 0\le S(y_j)\le 1 \tag{15}$$

$$G(y_j)=S(y_j)(2-(M(y_j)-mid)^2) \qquad 0\le G(y_j)\le 2 \tag{16}$$

$$G(y_j)=S(y_j)(2-M^2(y_j)) \tag{17}$$

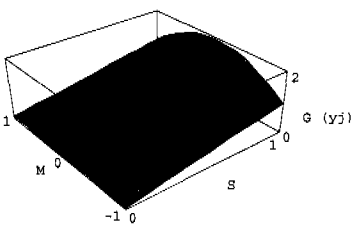


그림 4. 함수 G(yj)

G(yj)의 최소화(gradient descent)는 분산S(yj)를 작게 평균M(yj)를 시그모이드 함수의 포화영역에 속하게 한다. 반면, G(yj)의 최대화(gradient ascent)는 분산S(yj)를 크게 평균M(yj)를 시그모이드 함수의 중간값에 근접시킨다. 그러므로 학습이 진행됨에 따라, G(yj)의 특성을 입력층과 은닉층 사이의 가중치에 적용하기 위하여 식(2)의 MEBP의 오차함수에 G(yj)를 추가한 식(18)과 같은 cost function을 정의한다

$$C=E^{MEBP}\pm G(y_j) \tag{18}$$

정의된 cost function에 의한 MEBP학습에서 w_{jk}의 조정은 기존 MEBP학습과 같고 v_{ij}의 조정은 G(yj)에 대해 chain rule을 적용한다. 그리고 전체 은닉 노드중 G(yj)의 값이 가장 작은 노드를 식(19)과 같이 최소화 하고 나머지 노드는 G(yj)를 최대화 한다.

$$C=\begin{cases}E^{MEBP}+G(y_j) & \text{if } j=\min[G(y_j)]\\ E^{MEBP}-G(y_j) & \text{otherwise}\end{cases}$$

$$\Delta v_{ij}=-\eta\frac{\delta C}{\delta v_{ij}}=-\eta\frac{\delta E^{MEBP}\pm\delta G(y_j)}{\delta v_{ij}}$$

$$=\Delta v_{ij}^{MEBP}\mp\eta\frac{2}{P}\Big((2-M^2(y_j))\sum_{p=1}^{P}y_{pj}y'_{pj}x_{pi}+(M^2(y_j)-S(y_j)-2)M(y_j)\sum_{p=1}^{P}y'_{pj}x_{pi}\Big) \tag{19}$$

그 결과, G(yj)의 값이 임계치(ε) 이하(-1~1의 시그모이드 함수: 0.000735)로 낮아지면 불필요한 은닉 노드가 된다. 이런 불필요한 은닉 노드는 식(20)처럼 은닉 노드의 평균을 출력층의 바이어스에 반영하고 해당 은닉 노드를 제거한다. 그리고 은닉 노드가 제거된 후, 추가적인 은닉 노드의 축소를 위해 같은 과정을 학습이 완료될 때까지 수행한다. 지금까지는 전제 학습 패턴에 대해 한번의 가중치만을 조정하는 batch 학습이다.

$$\begin{aligned}\alpha_k &= w_{jk}M(y_j) \quad \text{if } G(y_j)<\varepsilon(=0.000735)\\ w_{0k} &= w_{0k}+\alpha_k\end{aligned} \tag{20}$$

제안한 알고리즘의 기본 개념을 확장하여 standard학습에 적용한다. Standard학습은 하나의 학습 패턴에 대해 가중치를 조정한다. 따라서 전체 학습 패턴에 대한 은닉 노드의 평균과 분산은 현재 은닉 노드의 출력과 과거 은닉 노드의 출력을 포함한 식

(21)과 식(22)로 근사화한다. 근사화된 평균과 분산을 이용하여 standard학습을 위한 $G(y_j)$를 식(23)으로 정의하고, cost function에 대한 가중치v_{ij}의 조정은 식(24)을 이용한다.

$$M(y_j,t)=\frac{1}{P}\sum_{p=0}^{P-1}y_j(t-p) \quad -1\le M(y_j,t)\le 1 \tag{21}$$

$$S(y_j,t)=\frac{1}{P}\sum_{p=0}^{P-1}(y_j(t-p))^2-M^2(y_j,t) \quad 0\le S(y_j,t)\le 1 \tag{22}$$

$$G(y_j,t)=S(y_j,t)(2-M^2(y_j,t)) \tag{23}$$

$$\Delta v_{ij}=-\eta\frac{\delta C}{\delta v_{ij}}=-\eta\frac{\delta E^{MEBP}\pm\delta G(y_j,t)}{\delta v_{ij}} =\Delta v_{ij}^{MEBP}\mp\eta\frac{2}{P}y'_{pj}x_{pi}\big((2-M^2(y_j,t))y_{pj}+ (M^2(y_j,t)-S(y_j,t)-2)M(y_j,t)\big) \tag{24}$$

함수$G(y_j)$는 오차의 감소를 측정하는 것이 아니라 은닉 노드의 활성화 정도를 측정하는 것이다.그러므로 제안한 방법은 gradient descent방법에 의해 오차를 축소하는 EBP학습이나 MEBP학습등에 적용할 수 있다.

Ⅲ. 실험 및 결과

실험은 필기체 숫자인식 문제를 대상으로 실험하였다. 필기체 숫자는 CEDAR 데이터 베이스[17]의 숫자중 각 숫자에 대해 100개씩 모두 1,000개를 학습에 사용하였고, 시험 패턴은 각 숫자에 대해 50개씩 500개의 숫자를 이용하였다. 한 숫자의 영상은 12x12 픽셀(pixel)로 구성되며, 각 픽셀은 16 level의 gray image로 바꾼후 [-1~+1]로 정규화 한 다음, MLP에 입력시켰다. 그림 5는 숫자 3에 대한 학습 패턴을 나타낸다.

000001499a50	001222000000	000079abb830	000003998520
0000296202a0	00bfffd30000	000089b9bee0	000003acefd3
000000001920	001457df3000	000000002ea0	0000000002f9
00000002a400	0000002f9000	00000029fa00	0000000016e5
000016bec200	0000008f7000	00005dfd4000	00000036bf60
0019fe713900	000009ff3000	0001fffb5000	0000009cff20
00ae92001900	00000adffd40	00000018de20	0000000029d2
001100005700	000000017ff5	000000000cd0	2000000000e5
00000002a000	461000138ff5	0000000003f3	4875000000c6
4c82003b4671	affdddfffe50	000000015ec0	006bab5338e3
98beb8eeb500	05bdddb74000	3b4469dff910	000006bdec30
28bb97400000	000000000000	06bbba730000	000000012000

그림 5. 숫자 3에 대한 학습 패턴

은닉 노드를 축소하지 않는 MEBP학습은 입력 노드 144개, 출력 노드 10개, 은닉 노드 6~20개로 MLP를 구성하여 standard학습을 했다. 그리고 각 은닉 노드의 수에 대해 가중치의 초기화를 6회씩 다르게 하여 총 90회를 학습했다. 학습의 완료 조건은 학습 패턴을 100%인식하거나 학습 시간을 10000 ecpoch미만으로 했다. MEBP학습중 6~9개의 은닉 노드는 100%학습을 하지 못하는 경우도 발생했다. 표 1은 각 은닉 노드의 6회 실험에 대한 MEBP학습의 평균 결과이다.

그림 6은 은닉 노드 10개로 MEBP학습을 한후, 은닉층 다차원공간에서 원점에 대한 학습 패턴의 유클리드 거리를 히스토그램으로 나타냈다. 이 그림에서 원점과 vertex사이의 최대거리는 3.1623이고 대부분의 학습 패턴들이 vertex에 가까이 분포하고 있음을 보여준다. 따라서 은닉층은 학습 패턴을 은닉층 공간의 vertex쪽으로 이동시키는 역할임을 알 수 있다.

표 1. MEBP학습 결과

은닉 노드수	평균 인식률(%)		평균 학습 시간 (epoch)
	학습	시험	
6	99.63	78.10	9360.33
7	99.73	75.43	9908.16
8	99.93	78.26	6539.33
9	99.82	80.33	7454.50
10	100	81.87	2503.83
11	100	82.67	1603.17
12	100	84.97	864.17
13	100	85.37	799.83
14	100	85.66	942.33
15	100	86.13	635.67
16	100	87.50	563.17
17	100	87.00	547.33
18	100	88.10	499.67
19	100	87.93	499.67
20	100	87.93	508.67

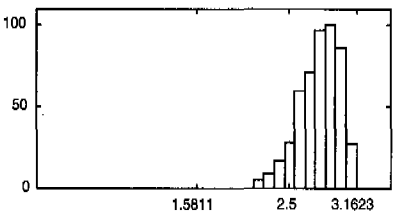


그림 6. 은닉층 출력의 유클리드 거리

제안한 방법과 비교를 위해 기존 은닉 노드 축소

방법으로 Mozer와 Smolensky의 방법과 Chauvin의 방법을 이용했다. Kruschke의 방법은 상수 r의 값에 따라 은닉 노드의 축소률이 유동적으로 변하기 때문에 실험에서 제외했다.

각 방법은 초기 은닉 노드를 11~20개로 구성한 후, 은닉 노드를 축소하였다. 학습 방법 및 조건은 모두 MEBP학습과 동일하게 했으며, 각 은닉 노드에 대해 6회씩 총 60회를 standard학습으로 실험했다. 은닉 노드의 축소는 학습 패턴의 인식률이 30~90%동안 각 epoch를 기준으로 수행했다.

각 방법의 불필요한 은닉 노드의 기준을 설명하면 다음과 같다.

- 제안한 방법 : 평균[$M(y_j)$=0.948+(0.052/2)=0.974]이 포화 영역에 속하며, 분산 [$S(y_j)$=(1-0.974)2=0.0007]이 작은 은닉 노드로써 $G(y_j)$=0.000735이하인 은닉 노드를 축소했다.
- Mozer 방법: ρ_j는 학습이 진행됨에 따라 증가 한다. 따라서 매 epoch마다 ρ_j를 최대값으로 정규화 한 다음 그 값이 0.1이하인 은닉 노드를 축소했다.
- Chauvin 방법: 전체 학습 패턴에 대한 은닉 노드의 평균 출력이 0<|0.052|인 노드를 축소했다.

각 방법에서 은닉 노드의 축소 기준($G(y_j)$, ρ_j, $e(y_j^2)$)에 대한 계산은 MLP의 은닉 노드 값을 출력할 때 함께 계산했다. 또한 불필요한 은닉 노드의 선택은 각 방법 모두 비교 연산을 수행하여, 각 epoch에 걸리는 학습 시간을 동일하게 했다.

표 2는 각 방법으로 초기 은닉 노드를 축소한 후, 최종 은닉 노드를 기준으로 60회의 실험중 최종 은닉 노드의 도수와 도수에 대한 평균 인식률과 평균 학습 시간을 나타낸 결과이다. 표 2의 실험 결과 처럼 제안한 방법은 다른 방법보다 더 많은 은닉 노드를 축소하고 있다. 그리고 MEBP학습시 6~9개의 은닉 노드로 학습할 수 없는 경우도 제안한 방법은 학습할 수 있었다.

표 2의 결과와 표 1의 MEBP학습 결과를 비교한 그래프는 그림 7,8이다. 그림 7은 최종 은닉 노드의 수를 기준으로 MEBP학습과 각 방법의 학습 시간을 비교했다. 그림 7에서 제안한 방법은 다른 방법보다학습 시간이 현저히 단축됨을 알 수 있다. 그림 8은 시험 패턴에 대한 평균 인식률이다. 여기서도 제안한 방법은 최종 은닉 노드의 수는 작지만 다른 방법보다 좋은 인식률을 보이고 있다. 그림 7,8에서 제안한 방법, Mozer방법, MEBP학습, Chauvin방법 순서로 좋은 결과를 보이고 있다. 제안한 방법은 적은 은닉 노드의 수로 학습 시간을 단축시키면서 높은 인식률을 얻을 수 있었다.

표 2. 은닉 노드 축소 결과

최종 은닉 노드수	방 법	도수	평균 인식률(%)		평균 학습 시간(epoch)
			학습	시험	
6	제안방법	8	99.97	79.67	5119.12
	Mozer	3	99.76	76.26	10000.00
	Chauvin				
7	제안방법	6	100	83.36	2695.50
	Mozer	5	99.96	78.84	6055.40
	Chauvin	1	86.70	70.40	10000.00
8	제안방법	7	99.98	84.57	2483.14
	Mozer	8	99.95	80.80	5045.62
	Chauvin	7	93.51	73.34	8047.57
9	제안방법	10	100	86.46	1099.70
	Mozer	8	100	83.62	1754.75
	Chauvin	9	99.27	79.64	5202.77
10	제안방법	8	100	86.45	1035.62
	Mozer	13	100	83.60	1452.61
	Chauvin	7	100	80.74	3089.14
11	제안방법	7	100	87.08	649.57
	Mozer	5	100	83.76	999.40
	Chauvin	12	100	82.50	1932.08
12	제안방법	9	100	87.68	601.11
	Mozer	5	100	86.28	921.00
	Chauvin	12	100	84.20	1134.00
13	제안방법	4	100	87.85	510.25
	Mozer	9	100	86.80	685.33
	Chauvin	5	100	84.56	1139.60
14	제안방법	1	100	88.60	497.00
	Mozer	2	100	87.10	581.00
	Chauvin	2	100	86.70	637.00
15	제안방법				
	Mozer	1	100	89.80	658.00
	Chauvin	5	100	86.52	638.00
16	제안방법				
	Mozer	1	100	87.80	756.00
	Chauvin				

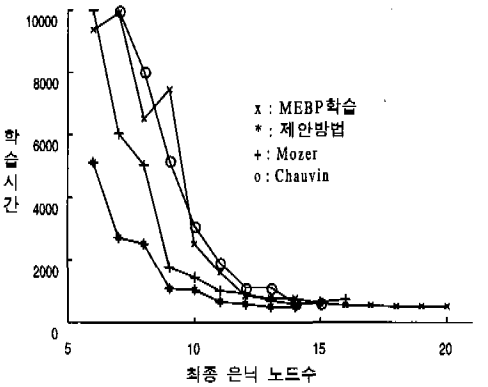


그림 7. 학습시간 비교

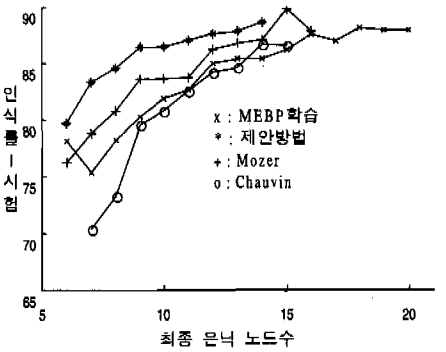


그림 8. 인식률 비교

기존 은닉 노드 축소 방법은 불필요한 은닉 노드의 제거시 그 노드가 학습한 정보량을 동시에 제거한다. 그러나 제안한 방법은 제거되는 은닉 노드가 가진 최소한의 학습 정보도 출력 노드에 전달하는 차이점이 있으며 학습 시간과 인식률에서 보다 좋은 결과를 보이고 있다.

그림 9는 각 방법에 대해 초기 은닉 노드의 수를 기준으로 은닉 노드를 축소한 후, 각 6회의 실험에 대한 평균적인 최종 은닉 노드의 수를 나타낸 그림이다. 이 그림에서도 제안한 방법이 다른 방법에 비해 높은 은닉 노드의 축소률을 나타냈다.

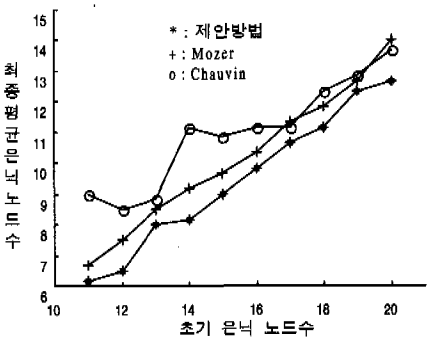


그림 9. 최종 은닉 노드수의 비교

전체 실험을 대상으로 총 은닉 노드의 축소률을 나타낸 결과는 표 3이다. 표 3의 'No convergence'는 제한된 학습 시간내에서 학습을 하지 못한 경우이다. 그리고 총 은닉 노드의 축소률의 계산시 'No convergence'은 제외했다. 표 3의 결과처럼 제안한 방법은 학습을 성공할 확률이 높고 다른 방법보다 축소률이 높으며 초기 은닉 노드의 37.2%까지 축소할 수 있었다.

표 3. 총 은닉 노드의 축소률

	제안 방법	Mozer	Chauvin	MEBP 학습
No convergence	3/60 (5%)	6/60 (10%)	7/60 (11.66%)	12/90 (13.33%)
총 은닉 노드의 축소률	346/930 (37.2%)	291/930 (31.29%)	239/930 (25.69%)	

결과적으로 제안한 방법은 MEBP학습보다 적은 은닉 노드로 학습을 할 수 있으며 학습 시간과 인식률에서 우수한 결과를 얻을 수 있었다. 그리고 다른 은닉 노드 축소 방법과 비교에서 학습 시간이 단축되고 은닉 노드의 축소률과 인식률이 높으므로 효과적인 은닉 노드 축소 방법임을 나타낸다.

Ⅳ. 결 론

본 논문은 학습하는 동안 은닉 노드의 출력에 대한 분산과 평균을 측정하는 새로운 cost function으로 불필요한 은닉 노드를 축소하는 방법을 제안하였다. 제안한 cost function은 필요한 은닉 노드의 분산을 크게 하고 평균을 활성화 함수의 중간값에 근사하도록 가중치를 조정한다. 반대로 불필요한 은닉 노드의 분산은 작고 평균은 활성화 함수의 포화 영역에 속하도록 가중치를 조정하여 그 출력을 상수화 하여 제거한다.

필기체 숫자인식 문제를 대상으로 한 실험에서, 제안한 방법은 학습 시간을 단축시키고 인식률을 높이면서 초기 은닉 노드의 수를 37.2%까지 축소했다. 또한 Mozer와 Smolensky의 방법과 Chauvin의 방법을 이용한 실험과의 비교에서도 학습 시간, 인식률, 은닉 노드의 축소률이 상대적으로 우수했다.

따라서 제안한 방법은 비결정적인 성질을 가진 MLP의 구조결정 문제를 해결할 수 있다. 그리고 학습 시간의 단축, 높은 인식률, 하드웨어 구현시 은닉 노드의 축소와 같은 장점이 있으며 학습 패턴에 대한 재학습이 필요 없다.

참 고 문 헌

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

[2] A. Van Ooyen and B. Nienhuis, "Improving the convergence of the back-propagation algo-

rithm," *Neural Networks,* vol. 78, pp. 465-471, 1992.

[3] J. R. Chen and P. Mars, "Stepsize variation methods for accelerating the backpropation algorithm," *Proc. IJCNN Jan. 15-19, 1990, Washington,* DC, USA, vol. I, pp. 601-604

[4] Youngjik Lee and Sang-Hoon Oh, "Improving the Error Back-Propagation Algorithm," *Proc. ICONIP'94*, Seoul, pp. 772-777, 1994.

[5] C. M. Bishop, *Neural Networks for Patern Recognition*, Clarendon Press, Oxford, 1997.

[6] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359-366, 1989.

[7] Dan Hammerstrom, "Working with Neural Networks," *IEEE Spectrum*, pp. 46-53, July, 1993.

[8] S. E. Fahlman and C. Lebiere, "The cascade correlation learning architecture," *Neural Information Processing System2*, D. S. Touretzsky, ed. Morgan Kaufman, pp. 524-532, 1990.

[9] Lee T.-C., A. M. Peterson, J. C. Tsai, "A multi-layer feed-forward neural network with dynamically adjustable structures." *IEEE International Conference on Sytems, Man, and Cybernetics*, pp. 367-369, 1990.

[10] R. Reed, "Pruning Algorithms - A Survey," *IEEE Trans. Neural Networks*, vol. 4, no. 5, pp. 740-747, 1993.

[11] E. D. Karnin, "A simple procedure for pruning back-propagation trained neural networks," *IEEE Trans. Neural Networks*, vol. 1, no. 2, pp. 239-242, 1990.

[12] Yann Le Cun, Jhon S. Denker and Sara A. Solla, "Optimal Brain Damage," *Proc. of NIPS'89,* pp. 598-605, 1989.

[13] M. Hagiwara, "Novel back propagation algorithm for reduction of hidden units and acceleration of convergence using artificial selection," *Proc. IJCNN'90,* vol. 1, pp. 625-630, June, 1990.

[14] J. K. Kruschke , "Improving generalizaton in back-propagation networks with distributed bottlenecks," *Proc. Int. Joint Conf. Neural Networks*, Washington DC, vol. I, 1989, pp. 443-447

[15] Y. Chauvin, "A back-propagation algorithm with optimal use of hidden units," *Advances in Neural Information Processing (1)*, D. S. Touretzky, Ed. (Denver 1988), 1989, pp. 519-526.

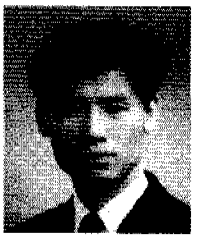
[16] M. C. Mozer and P. Smolensky, "Skeletonization: A technique for trmming the fat from a network via relevance assessment," *Advances in Neural Information Processing(I)*, D. S. Touretzky, Ed. (Denver 1988), 1989, pp. 107-115

[17] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern and Machine Intell.*, vol. 16, pp. 550-554, 1994.

[18] J. Villiers and E. Barnard, "Backpropagation Neural Nets with One and Two Hidden Layers," *IEEE Trans. Neural Netwoks,* vol. 4, no. 1, pp. 136-141, 1993.

[19] 오상훈, 이영직, 김명원, "역전파 학습시 초기 가중치가 학습의 조기 포화에 미치는 영향," *전자공학회논문지*, 제28권 4호, pp. 90-97, 1991.

곽 영 태(Young-Tae Kwak) 정회원



1993년 2월 : 충남대학교 컴퓨터공학과 학사
1995년 2월 : 충남대학교 컴퓨터공학과 석사
1996년 3월~현재 : 충남대학교 컴퓨터공학과 박사과정

<주관심 분야> 신경회로망, 패턴인식

이 영 직(Young-Gik Lee) 정회원

1979년 2월 : 서울대학교 전자공학과 학사
1981년 2월 : 한국과학기술원 산업전자공학과 석사
1989년 1월 : Polytechnic University 전기 및 전산과 박사
1981년~1984년 삼성전자 컴퓨터개발실
1989년 1월~현재: 한국전자통신연구원 멀티모달 I/F팀장, 책임연구원

<주관심 분야> 음성인식, 음성합성, 음성언어번역, 자동통역, 신경회로망, 패턴인식, 멀티모달 인터페이스

권 오 석(Oh-Seok Kwon) 정회원

1977년 2월 : 서울대학교 전자공학과 학사
1980년 2월 : 한국과학기술원 산업전자공학과 석사
1992년 2월~현재 : 한국과학기술원 전산학과 박사과정
1980년~현재: 충남대학교 컴퓨터공학과 교수
<주관심 분야> 신경회로망, 패턴인식, 퍼지이론 및 응용