

# 다층퍼셉트론의 성능향상을 위한 출력노드 수 증가

## Increasing Output Nodes for Performance Improvement of Multilayer Perceptrons

오상훈  
목원대학교

Oh Sang-Hoon  
Mokwon University

### 요약

다층퍼셉트론을 패턴인식 문제에 적용하는 경우 대부분 하나의 클래스 당 하나의 출력노드를 할당한다. 이 논문에서는 class 당 출력노드 수를 증가시키는 경우에 다층퍼셉트론의 성능이 향상되는 지 고립 단어 인식 문제의 시뮬레이션을 통하여 고찰하였다. 시뮬레이션 결과 출력노드 수가 하나인 경우보다 증가시킨 경우 더 나은 성능을 얻었다.

### Abstract

When we use multilayer perceptron model for pattern classification problems, we allocate one output node for each class. In this paper, we increase the number of output nodes for each class and investigate the performance of multilayer perceptrons through the simulation of isolated-word recognition problems.

## I. 서론

다층퍼셉트론(MLP: multilayer perceptron)은 층간 중 간층 노드 수가 있으면 임의의 함수를 근사화시킬 수 있다는 수학적 증명을 근거로 많은 응용 문제에 적용되어 왔으며, 특히 패턴인식 문제에 많이 사용되었다.[1] 패턴인식 문제에 사용되는 경우, MLP의 구조에서 출력층 노드는 패턴 class 수 만큼 주어지, 하나의 출력노드가 하나의 class를 대표하게 된다. 그리고, 학습이 완료된 MLP는 입력패턴에 의해 계산된 출력노드의 값이 입력패턴이 어느 class에 속할 확률을 나타낸다는 것도 수학적으로 증명되었다.[2]

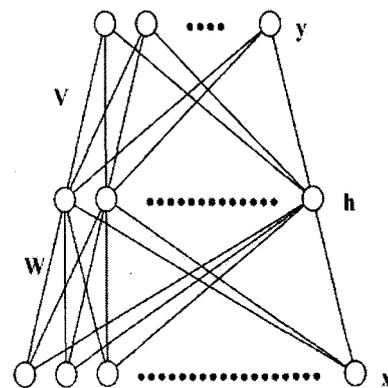
그렇지만, MLP의 학습속도가 느리거나 혹은 시험패턴에 대한 인식률이 충분치 않다는 단점을 해결하기 위하여 많은 개선책들이 제안되었다. 학습률을 학습 진행에 따라 변경하는 Bold Driver[3]나 Delta-Bar-Delta[4] 같은 방법이 있으며, MLP의 초기 가중치 값을 조정하는 방법도 있다.[5] 또한, CFM(Classification Figure of Merit)[6], CE(Cross Entropy)[7], nCE(n-th order extension of CE)[2] 같이 오차함수를 새롭게 제안한 방법이 있다. 그리고, Newton's method나 Conjugate Gradient Method처럼 2차 미분을 가중치의 변경에 사용하는 방법도 있다. 또한, MLP의 각 층을 분리해서 학습시키는 LBL (Layer-By-Layer)방법과 이를 위하여 중간층의 오차함수를 일반적인 MSE(Mean-Squared Error)가 아니라 중간층의 시그모이드 함수 기울기를 고려하여 변경시키는 방법도 제안되었다.[8]

이 논문에서는 MLP의 성능을 향상시키기 위하여 출력노드

수를 증가시켜 보았다. 즉, 패턴인식 문제에서 class당 하나의 출력노드가 할당되는데, 이 class당 출력노드 수를 증가시키면서 시험패턴에 대한 인식률이 향상되는 지를 점검하였다. 이는 여러 개의 인식기를 인위적으로 융합시키는 것이 아니라 학습에 의해 스스로 융합하는 효과가 있는 지를 점검해보고자 한 것이다.

먼저 Section II에서 MLP의 구조와 일반적인 MSE에 의한 학습방법을 설명한다. Section III에서 MLP의 성능향상을 위하여 제안된 nCE 오차함수를 이용한 MLP의 학습을 설명하고 또한 출력노드 수를 증가시키는 방법을 제안한다. Section IV에서 고립 단어 인식문제의 시뮬레이션으로 제안한 방법의 효용성을 확인하고 마지막으로 Section V에서 결론을 맺겠다.

## II. MLP의 구조와 학습방법



▶▶ 그림 1. 다층퍼셉트론 구조

다층퍼셉트론(MLP)이  $N$ 개의 입력 뉴런과  $H$ 개의 중간층 뉴런 및  $M$ 개의 출력 뉴런들로 구성되어 있다고 하자. 어떤 입력패턴  $\mathbf{x} = [x_1, x_2, \dots, x_N]$ 이 MLP에 입력되면,  $j$ 번째 중간층 뉴런의 값은

$$h_j = f(\hat{h}_j) = \tanh(\hat{h}_j/2), j = 1, 2, \dots, H \quad (1)$$

와 같이 주어진다. 여기서  $f(\cdot)$ 는 중간층 뉴런의 시그모이드 비선형 함수이며

$$\hat{h}_j = \sum_{i=0}^N w_{ji} x_i \quad (2)$$

는 중간층 뉴런에 입력되는 가중치 합이다.  $w_{ji}$ 는  $x_i$ 와  $h_j$ 를 연결하는 중간층 가중치이며  $x_0=1$ 로 주어지기에  $w_{j0}$ 는 바이어스라고 불린다. 같은 형태로  $k$ 번째 출력 뉴런에 입력되는 가중치 합은

$$\hat{y}_k = \sum_{j=0}^H v_{kj} h_j, k = 1, 2, \dots, M \quad (3)$$

이고,  $v_{kj}$ 는  $h_j$ 와  $y_k$ 를 연결하는 출력층 가중치이고,  $h_0=1$ 이며  $v_{k0}$ 는 바이어스이다.

$P$ 개의 학습패턴  $\mathbf{x}^{(p)} (p=1, 2, \dots, P)$ 와 이들의 출력층 목표벡터  $\mathbf{t}^{(p)} = [t_1^{(p)}, t_2^{(p)}, \dots, t_M^{(p)}]$ 가 주어지면, 가중치들은 출력층에서

$$E^{out} = \frac{1}{2} \sum_{p=1}^P \sum_{k=1}^M (t_k^{(p)} - y_k^{(p)})^2 \quad (4)$$

로 주어지는 MSE(mean squared error)를 최소화시키도록 변경된다. 즉, EBP(Error Back-Propagation) 알고리즘에 의해 MLP의 출력층 각 노드에 연결된 출력층 가중치는

$$\Delta v_{kj} = -\eta \frac{\partial E^{out}}{\partial v_{kj}} = \eta \delta_k^{(p)} h_j^{(p)} \quad (5)$$

에 따라 학습된다. 여기서,

$$\delta_k^{(p)} = -\frac{\partial E^{out}}{\partial y_k^{(p)}} = (t_k^{(p)} - y_k^{(p)}) f'(y_k^{(p)}) \quad (6)$$

이다. 또한, 중간층 노드에 연결된 중간층 가중치는

$$\Delta w_{ji} = -\eta \frac{\partial E^{out}}{\partial w_{ji}} = \eta f'(\hat{h}_j^{(p)}) x_i^{(p)} \sum_{k=1}^M v_{kj} \delta_k^{(p)} \quad (7)$$

에 따라 학습된다.

### III. MLP의 nCE 학습과 출력노드수 증가

MLP의 MSE 최소화를 위한 EBP 학습은 학습속도가 느린 단점이 있다는 것은 많이 지적되었다.[2] 특히, 초기 가중치를 설정함에 있어, 초기가중치의 크기가 크면 EBP 학습의 조차 포화 현상이 나타나 학습속도가 느려짐은 물론, 학습 성능도 나뭇이 수학적으로 분석되었다.[5] 여기서, 조차포화 현상이라는 것은 학습이 진행되지 않은 상황에서 특정 입력패턴에 대한 출력노드 값을 계산한 경우 그 값이 목표값과 멀리 떨어져  $(t_k^{(p)} - y_k^{(p)})$ 가 큼에도 불구하고 시그모이드 함수의 기울기  $f'(y_k^{(p)})$ 가 작은 값을 지니게 되어,  $\Delta v_{kj}$ 가 아주 작게 되어 학습이 잘 되지 않는 현상이다.

이러한 점에 착안하여 식 (6)에서 시그모이드 기울기에 해당하는 항을 제거하기 위하여 CE 오차함수가 제안되었으며, 이 보다 더 나은 성능을 지닌 nCE 오차함수 역시 제안되었다. 이 논문에서는 nCE 오차함수를 이용하여 MLP 학습에 대한 시뮬레이션을 한다. nCE 오차함수는

$$E_{nCE}^{out} = -\frac{1}{2} \sum_{p=1}^P \sum_{k=1}^M \int \frac{t_k^{(p)n+1} (t_k^{(p)} - y_k^{(p)})^n}{2^{n-2} (1 - y_k^{(p)})^2} dy_k^{(p)} \quad (8)$$

와 같이 주어지며, 이 경우 출력층의 error 신호는

$$\delta_k^{(p)} = -\frac{\partial E_{nCE}^{out}}{\partial y_k^{(p)}} = \frac{t_k^{(p)n+1} (t_k^{(p)} - y_k^{(p)})^n}{2^{n-1}} \quad (9)$$

로 변형된다.[2] 식 (9)를 보면 시그모이드의 기울기에 해당하는 항이 없어졌음을 알 수 있다. 이 error 신호를 목표값이 1인 경우 출력값의 변화에 대한 형태로 MSE를 이용한 conventional 방법, CE를 이용한 방법, nCE에서  $n=2$ 와 4인 경우에 대하여 비교하여 보면, MSE를 이용한 방법은 비록 목표값이 1임에도 불구하고 출력값이 -1에 근접한 경우 아주 작은 error 신호가 발생한다. CE를 이용한 방법의 경우 이러한 단점은 해결되었으나, 출력값이 1에 근접한 경우에도 error 신호가 상당히 과도하다. nCE를 이용한 방법은 CE의 이러한 단

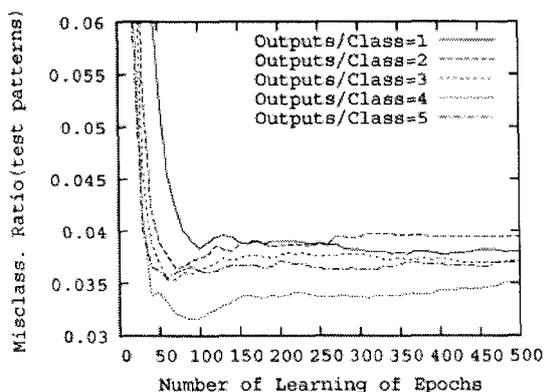
점도 해결되어 출력값이 1에 근접한 경우 error 신호가 아주 작아진다. nCE를 이용한 MLP 학습은 식(9)를 (5)와 (7)에 대입하여 출력층 가중치와 중간층 가중치를 변경시키는 것이다.

한편, 학습패턴의 수가 충분하다면 출력노드의 값은 어떤 입력패턴이 특정 class를 나타낼 확률이 수학적으로 유도되었다. 이를 근거로 MLP를 패턴인식 문제에 적용할 경우, 출력노드를 class당 하나씩 할당하여, 그 출력노드의 출력값 중 가장 큰 출력값이 나타내는 class를 입력패턴의 class라고 판단을 내린다.

하나의 패턴인식기로 충분한 성능을 얻지 못하는 경우, 여러 개의 패턴인식기를 구현한 후 그 출력을 서로 융합하여 인식 성능을 향상시키는 방법이 많이 사용된다. 이와 같은 인위적인 융합방법이 아니라, MLP에 단순히 class 당 출력노드 수를 증가시켜도 역시 여러 패턴인식기가 융합되어 인식 성능이 향상되는 효과를 기대할 수 있을 것이다. 따라서, 이 논문에서는 class당 출력노드 수를 증가시키는 방법을 통한 인식 성능 향상을 제안한다.

#### IV. 시뮬레이션

제안한 방법의 효용성을 확인하기 위하여 고립 단어 인식 문제를 MLP에 학습시켰다. 50단어를 9명의 화자가 2번씩 발성한 900개의 음성패턴을 1,024차원의 ZCPA를 추출한 후 MLP에 입력하여 학습시켰다.[9] MLP의 입력층 노드수는 1,024이며, 중간층 노드는 50개를 사용하였다. 또 다른 7명의 화자가 50단어를 3번씩 발성한 1,050음성패턴을 시험패턴으로 사용하여 MLP의 인식 성능을 조사하였다. 초기 가중치는  $[-1 \times 10^{-4}, 1 \times 10^4]$ 에서 균일분포를 지니도록 하였으며, 학습률은 0.05를 사용하였다. 그리고, class 당 출력노드 수는 1, 2, 3, 4, 5개가 되도록 하여 시험패턴에 대한 오인식률의 변화를 조사하였다.



▶▶ 그림 2. 출력노드 수 변경에 따른 인식 성능

그림 2에서 보는 바와 같이 class당 출력노드가 하나인 경우 시험패턴에 대한 오인식률은 340 epoch에서 3.78% 정도이다. 한편, 출력노드 수를 증가시켜 class당 출력노드 수가 2인 경우는 70번째 epoch에서 3.65%, 3인 경우 60번째 epoch에서 3.54%, 4인 경우 90번째 epoch에서 3.16%, 5인 경우 60번째 epoch에서 3.55%를 얻었다. 출력노드 수를 증가시키면 시험패턴에 대한 오인식률이 빨리 감소하며, 그 성능도 우수해짐을 볼 수 있다. 즉, 마치 여러 개의 패턴인식기가 융합되어 학습 성능이 향상되는 효과가 있다.

#### V. 결론

이 논문에서는 패턴인식 문제에 MLP를 적용시킬 경우에 학습 성능을 향상시키는 방법을 살펴보고, 그 중 nCE를 이용한 방법을 채택한 후, class 당 출력노드 수를 증가시켜 마치 여러 개의 패턴인식기가 융합되는 것과 같은 효과가 일어나도록 하는 방법을 제시하였다. 고립 단어 인식 문제의 시뮬레이션으로 제안한 방법이 인식 성능을 향상시키는 효과를 확인하였다.

#### 참고 문헌

- [1] Y. Ito, "Approximation of continuous functions on  $r^d$  by linear combinations of shifted rotations of a sigmoid function with and without scaling," *Neural Networks*, Vol.5, pp.105-115, 1992.
- [2] S.-H. Oh, "Improving the error back-propagation algorithm with a modified error function," *IEEE Trans. Neural Networks*, Vol.8, pp.799-803, 1997.
- [3] T. P. Vogl, J. K. Mangis, A. K. Rigler, W. T. Zink, and D. L. Alkon, "Accelerating the convergence of the back-propagation method," *Biol. Cybern.*, Vol.59, pp.257-263, 1988.
- [4] J. Y. F. Yam and W. S. Chow, "Extended least squares based algorithm for training feedforward networks," *IEEE Trans. Neural Networks*, Vol.8, pp.806-810, 1997.
- [5] Y. Lee, S.-H. Oh, and M. W. Kim, "An analysis of premature saturation in back-propagation learning," *Neural networks*, Vol.6, pp.719-728, 1993.
- [6] J. B. Hampshire II and A. H. Waibel, "A novel objective function for improved phoneme recognition using time-delay neural networks," *IEEE Trans. Neural Networks*, Vol.1, pp.216-228, June 1990.
- [7] A. van Ooyen and B. Nienhuis, "Improving the convergence of the back-propagation algorithm," *Neural Networks*, Vol.5, pp.465-471, 1992.
- [8] S.-H. Oh and S.-Y. Lee, "A new error function at hidden layers for fast training of multilayer perceptrons," *IEEE Trans. Neural Networks*, Vol.10, pp.960-964, 1999.
- [9] D.-S. Kim, S.-Y. Lee, and R.-M. Kil, "Auditory processing of speech signals for robust speech recognition in real-world noisy environments," *IEEE Trans. Speech and Audio Processing*, Vol.7, pp.55-69, 1999.