

다층퍼셉트론의 출력 노드 수 증가에 의한 성능 향상

Performance Improvement of Multilayer Perceptrons with Increased Output Nodes

오상훈

목원대학교 정보통신공학과

Sang-Hoon Oh(shoh@mokwon.ac.kr)

요약

일반적으로 다층퍼셉트론을 패턴인식 문제에 적용할 경우 클래스 당 하나의 출력 노드를 배정하고, 이 출력 노드의 인덱스가 입력 패턴의 클래스를 뜻하도록 한다. 이 논문에서는 이와 달리 다층퍼셉트론의 성능 향상을 위하여 클래스 당 출력노드 수를 증가시키는 방법을 제안한다. 두 개의 클래스 문제를 대상으로 클래스 발생확률이 동일하고 각 클래스 내에서 출력노드가 균일분포를 지닌다는 가정 하에, 이 방법의 효용성을 확률론적인 유도를 통하여 증명하였다. 그리고, 50개의 고립단어 인식의 시뮬레이션으로 출력노드를 증가 시킬 경우 성능이 향상됨을 확인하였다.

■ 중심어 : | 다층퍼셉트론 | 성능향상 | 출력 노드 수 |

Abstract

When we apply MLPs(multilayer perceptrons) to pattern classification problems, we generally allocate one output node for each class and the index of output node denotes a class. On the contrary, in this paper, we propose to increase the number of output nodes per each class for performance improvement of MLPs. For theoretical backgrounds, we derive the misclassification probability in two class problems with additional outputs under the assumption that the two classes have equal probability and outputs are uniformly distributed in each class. Also, simulations of 50 isolated-word recognition show the effectiveness of our method.

■ keyword : | Multilayer Perceptrons | Performance Improvement | Output Nodes |

I. 서론

다층퍼셉트론(MLP: multilayer perceptron)은 충분한 중간층 노드 수가 있으면 임의의 함수를 근사화 할 수 있다는 수학적 증명을 근거로 많은 응용 문제에 적용되어 왔으며, 특히 패턴인식 문제에 많이 사용되었다[1]. 패턴인식 문제에 사용되는 경우, MLP의 구조에서 출력층 노드는 패턴의 클래스(class) 수만큼 주어지며, 하나의

출력노드가 하나의 클래스를 대표하게 된다. 그리고, 학습이 완료된 MLP에서 입력패턴에 의해 계산된 출력노드의 값은 입력패턴이 어느 클래스에 속할 확률을 나타낸다는 것도 수학적으로 증명되었다[2]. 또한, 입력된 패턴에 잡음이 섞이는 경우 출력에 나타나는 MLP의 잡음 강건성이 정성적으로 분석되고, 정량적으로는 MLP의 입력/중간층의 노드 수 및 가중치의 함수로 유도되었다[3].

그렇지만, MLP의 학습은 MSE(mean-squared error)를 최소화 시키는 EBP(error back-propagation) 알고리즘[4]을 따르는데, 이 EBP 알고리즘의 학습속도가 느리거나 혹은 시험패턴에 대한 인식이 충분치 않다는 단점을 해결하기 위하여 많은 개선책들이 제안되었다.

학습 성능을 향상시키기 위하여 오차함수를 제안한 방법으로 CE(Cross Entropy)[5]과 nCE(n-th order extension of CE)[6][7] 오차함수가 있는데, 이 방법들은 출력층 가중치의 변경 식에서 시그모이드 기울기 항을 제거하는 방식이다. 또한, CFM(Classification Figure of Merit)[8]함수는 클래스가 속하는 것을 나타내는 출력노드와 다른 출력 노드들 간의 출력값 차이가 최대가 되게 하기 위하여 제안된 방법이다.

한편, 클래스 경계면에 속한 패턴들을 찾아내어 학습에 사용하는 방법[9], 선택적 주의 집중 역할을 수행하는 입력층을 사용하는 방법[10], 출력층을 여러 개로 늘리는 방법[11] 등도 제안되었다.

또한, 학습속도를 빠르게 하기 위한 방법들도 제안되었는데, 학습률을 변동시키는 방법으로는 Bold Driver[12]나 Delta-Bar-Delta[13] 등이 있다. 또한, MLP의 각 층을 분리해서 학습시키는 LBL(Layer-By-Layer)방법[14][15]과 이를 위하여 중간층의 오차함수를 일반적인 MSE가 아니라 시그모이드 함수 기울기를 고려하여 변경시키는 방법도 제안되었다[16].

이 논문에서는 이러한 접근들과 달리 MLP의 성능을 향상시키기 위하여 출력노드 수를 증가시키는 새로운 방법을 제안한다. 일반적으로, 패턴인식 문제에서 클래스 당 하나의 출력노드가 할당되고, 임의의 입력패턴이 주어지면 MLP의 출력 값을 계산한 후 최대의 출력 값을 지닌 출력노드의 인덱스가 입력 패턴이 속한 클래스를 나타내는 Max. 범칙에 따라 인식결과를 정한다. 이러한 인식기의 성능을 향상시키기 위하여 여러 개의 인식기를 구현하고, 그 각각의 인식기 출력으로부터 최종 인식 결과를 판단하는 방법도 제안되었다[17]. 이 경우는 최종 판단을 내리기 위한 별도의 네트워크가 구축되고 학습되어야 한다. 이와 달리, 여기서 제안하는 방법은 클래스 당 출력노드 수를 증가시키고 학습에 의해 여러

출력들이 스스로 융합하는 효과가 있는 것을 보이고자 한다.

먼저 Section II에서 MLP의 구조와 학습방법을 설명한다. Section III에서 출력노드 수를 증가시키는 방법을 제안하며 이의 효용성을 확률 유도를 통하여 증명하고, Section V에서 고립단어 인식문제의 시뮬레이션으로 제안한 방법의 효용성을 확인한다. 마지막으로 Section VI에서 결론을 맺는다.

II. 다층퍼셉트론의 구조와 학습방법

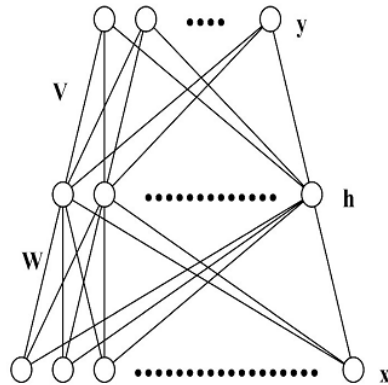


그림 1. 다층퍼셉트론 구조

다층퍼셉트론(MLP)이 [그림 1]과 같이 N 개의 입력노드와 H 개의 중간층 노드 및 M 개의 출력 노드들로 구성되어 있다고 하자. 어떤 N 차원의 입력패턴 $\mathbf{x} = [x_1, x_2, \dots, x_N]$ 이 MLP에 입력되면, j 번째 중간층 노드 h_j 의 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 시그모이드 비선형 함수이며

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

는 중간층 노드에 입력되는 가중치 합이다. w_{ji} 는 x_i 와 h_j 를 연결하는 중간층 가중치이며 $x_0 = 1$ 로 주어지고 w_{j0} 는 바이어스이다. 같은 형태로 k 번째 출력 노드에

입력되는 가중치 합은

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

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

$$y_k = f(\widehat{y}_k) = \tanh(\widehat{y}_k/2), k = 1, 2, \dots, M \quad (4)$$

로 주어진다. 이렇게 식 (1)에서 (4)까지의 과정에 따라 주어진 입력으로부터 출력 노드 값을 구하는 것을 전방향 계산이라 한다.

이러한 구조의 MLP에 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

$$E^{out} = \frac{1}{2} \sum_{p=1}^P \sum_{k=1}^M (t_k^{(p)} - y_k^{(p)})^2 \quad (5)$$

로 주어지는 MSE를 최소화시키도록 MLP의 가중치들이 변경된다. 즉, EBP 알고리즘[4]에 의해 MLP의 출력층 각 노드에 연결된 출력층 가중치는

$$\Delta v_{kj} = -\eta \frac{\partial E^{out}}{\partial v_{kj}} = \eta \delta_k^{(p)} h_j^{(p)} \quad (6)$$

에 따라 학습된다. 여기서,

$$\delta_k^{(p)} = -\frac{\partial E^{out}}{\partial y_k^{(p)}} = (t_k^{(p)} - y_k^{(p)}) f'(\widehat{y}_k^{(p)}) \quad (7)$$

이다. 또한, 중간층 노드에 연결된 중간층 가중치는

$$\Delta w_{ji} = -\eta \frac{\partial E^{out}}{\partial w_{ji}} = \eta f'(\widehat{h}_j^{(p)}) x_i^{(p)} \sum_{k=1}^M v_{kj} \delta_k^{(p)} \quad (8)$$

에 따라 학습된다.

위와 같은 EBP 학습은 학습속도가 느린 단점이 있다는 것이 많이 지적되었다[3]. 특히, 학습이 진행되지 않은 상황에서 특정 입력패턴에 대한 출력노드 값 $y_k^{(p)}$ 을 계산한 경우 그 값이 목표 값 $t_k^{(p)}$ 과 멀리 떨어져 있어 $(t_k^{(p)} - y_k^{(p)})$ 가 큼에도 불구하고 시그모이드 함수의 기울기 $f'(\widehat{y}_k^{(p)})$ 가 작은 값을 지니게 되어 식 (7)의 값이 작아지고, 결국 식 (6)에서 Δv_{kj} 가 아주 작게 되어 학습이 잘 되지 않는다.

이러한 점에 착안하여 식 (7)에서 시그모이드 기울기에 해당하는 항 $f'(\widehat{y}_k^{(p)})$ 을 제거하기 위하여 CE(Cross-Entropy) 오차함수[5]가 제안되었으며, 이보다 더 나은 성능을 지닌 nCE(n-th order extension of CE) 오차함수[6] 역시 제안되었다. 이 논문에서는 nCE 오차함수($n=4$)를 이용하여 MLP의 학습에 대한 시뮬레이션을 한다. nCE 오차함수[6]는

$$E_{nCE}^{out} = -\frac{1}{2} \sum_{p=1}^P \sum_{k=1}^M \int \frac{t_k^{(p)^{n+1}} (t_k^{(p)} - y_k^{(p)})^n}{2^{n-2} (1 - y_k^{(p)})^2} dy_k^{(p)} \quad (9)$$

와 같이 주어지며, 이 경우 출력층의 오차 신호는

$$\delta_k^{(p)} = -\frac{\partial E_{nCE}^{out}}{\partial y_k^{(p)}} = \frac{t_k^{(p)^{n+1}} (t_k^{(p)} - y_k^{(p)})^n}{2^{n-1}} \quad (10)$$

로 변형된다[6]. 식 (10)을 (7)과 비교해보면 시그모이드의 기울기에 해당하는 항 $f'(\widehat{y}_k^{(p)})$ 이 없어졌음을 알 수 있다. nCE를 이용한 MLP 학습은 식(10)을 (7)에 대입하여 출력층 가중치와 중간층 가중치를 변경시키는 것이다. 그리고, 이와 같이 출력층과 중간층의 가중치들을 변경을 하는 과정을 역방향 계산이라 한다.

III. 출력 노드 수 증가에 의한 성능향상

학습패턴의 수가 충분하다면 MLP의 출력노드 값은 어떤 입력패턴이 특정 클래스에 속할 확률을 나타내며 MLP는 Bayes 인식기임이 수학적으로 유도되었다[2]. 이를 근거로 MLP를 패턴인식 문제에 적용할 경우, 출력노드를 클래스 당 하나씩 할당하며 인식 결과는 Max. 법칙에 따르게 된다.

그렇지만, 이렇게 하나의 클래스 당 하나의 출력노드가 할당된 MLP가 충분한 성능을 얻지 못하는 경우 성능을 향상시키는 방법으로 클래스 당 출력노드 수를 증가시키는 방법을 제안한다. 이 방법은 입력층과 중간층은 공유하면서 각 클래스를 담당하는 출력노드 숫자는

여러 개가 되도록 하여 인식능력이 향상되도록 하는 방법이다.

제안한 방법의 효용성을 수학적으로 증명하기 위하여, 두 개의 클래스 c_1 과 c_2 가 같은 확률을 지니는 인식문제를 고려하자. 이 경우 출력노드의 목표값은 입력이 클래스 1에 속하면 -1로 주어지고 클래스 2에 속하면 +1로 주어졌다고 하자. 또한, 각 클래스에 속하는 입력패턴에 대하여 MLP의 출력노드는 균일분포를 가진다고 가정하자.

먼저, 출력 노드는 하나만 있는 경우에 오인식 확률을 유도하자. 이 경우 MLP의 출력 값이 음이면 c_1 , 양이면 c_2 로 판단을 내리게 되며, 각 클래스에서 출력 노드는

$$p(x|c_1) = \begin{cases} \frac{1}{2(x_{c_2} - x_{c_1})}, & x_{c_1} - 2x_{c_2} < x < x_{c_2} \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

과

$$p(x|c_2) = \begin{cases} \frac{1}{2(x_{c_2} - x_{c_1})}, & x_{c_1} < x < x_{c_2} - 2x_{c_1} \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

와 같이 균일 분포(uniform distribution)을 가진다고 가정한다. 여기서, x_{c_1} 은 음이고 x_{c_2} 는 양이라고 가정한다. 그리고, $p(\cdot)$ 는 확률밀도함수(pdf.: probability density function)이고, $P(\cdot)$ 는 확률질량함수(pmf: probability mass function)이다. 유도를 간편하게 하기 위하여 $x_{c_2} = -x_{c_1}$ 이라고 두자. Bayes 법칙은 특정 입력 x 가 주어진 경우 $p(x|c_1)P(c_1)$ 과 $p(x|c_2)P(c_2)$ 를 계산한 후 그 결과가 큰 것을 기반으로 하여 인식 결

과를 판단한다. 이에 따르면 오인식 확률은 입력이 클래스 2에 속하는 데 출력 x 가 음일 확률과 클래스 1에 속하는 데 출력 x 가 양일 확률의 합으로 계산되며

$$P_{E_1} = \int_{x_{c_1}}^0 p(x|c_2)P(c_2)dx + \int_0^{x_{c_2}} p(x|c_1)P(c_1)dx = \frac{1}{4} \quad (13)$$

와 같이 유도된다.

한편, 출력 노드를 두 개로 증가시켜서 각각을 x 와 y 로 표기하고, 같은 아이디어를 적용하여 확률밀도 함수를 정하면 식 (14)와 (15)로 주어진다. 여기에서도 x 와 y 의 값을 보고서 둘 다 음이면 c_1 , 둘 다 양이면 c_2 , 하나는 양이고 하나는 음이면 절대 값이 큰 출력의 부호를 보고서 판단을 내린다. 이 경우, c_2 에 속한 패턴이 오인식될 확률은 출력이 둘 다 음인 확률과 하나가 음이고 하나가 양인 경우 음인 출력의 절대 값이 양인 출력의 절대값 보다 더 큰 확률의 합에 해당하므로

$$P_{E_2}^{(c_2)} = \Pr\{x < 0, y < 0|c_2\}P(c_2) + \Pr\{|x| > |y|, x < 0, y > 0|c_2\}P(c_2) + \Pr\{|x| < |y|, x > 0, y < 0|c_2\}P(c_2) = \frac{1}{16} \quad (16)$$

이고, 같은 방식으로 c_1 에 속하는 패턴이 오인식될 확률은

$$P_{E_2}^{(c_1)} = \Pr\{x > 0, y > 0|c_1\}P(c_1) + \Pr\{|x| < |y|, x < 0, y > 0|c_1\}P(c_1) + \Pr\{|x| > |y|, x > 0, y < 0|c_1\}P(c_1) = \frac{1}{16} \quad (17)$$

로 구해진다. 그러면, 출력노드가 2개인 경우 오인식 확률은

$$p(x, y|c_1) = \begin{cases} \frac{1}{4(x_{c_2} - x_{c_1})(y_{c_2} - y_{c_1})}, & x_{c_1} - 2x_{c_2} < x < x_{c_2} \text{ and } y_{c_1} - 2y_{c_2} < y < y_{c_2} \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

$$p(x, y|c_2) = \begin{cases} \frac{1}{4(x_{c_2} - x_{c_1})(y_{c_2} - y_{c_1})}, & x_{c_1} < x < x_{c_2} - 2x_{c_1} \text{ and } y_{c_1} < y < y_{c_2} - 2y_{c_1} \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

$$P_{E_2} = P_{E_2}^{(c_1)} P(c_1) + P_{E_2}^{(c_2)} P(c_2) = \frac{1}{8} \quad (18)$$

이 된다. 이 결과를 출력노드가 하나인 경우 유도된 식 (13)과 비교하면

$$P_{E_1} > P_{E_2} \quad (19)$$

임을 알 수 있다. 즉, 출력노드 수를 증가시키면 오인식률이 감소함을 알 수 있다.

한편, 식 (13)의 경우는 오인식 확률이 $p(x|c_1)P(c_1)$ 와 $p(x|c_2)P(c_2)$ 가 겹치는 면적에 해당하며, 식 (18)의 경우는 $p(x, y|c_1)P(c_1)$ 와 $p(x, y|c_2)P(c_2)$ 가 겹치는 면적에 해당한다. 차원이 높아질 수록 이 겹치는 면적이 줄어들 것이다.

여기서 유도한 것은 클래스가 두 개인 경우에 출력노드의 값이 음인지 양인지에 의해 판단을 내리는 경우이다. 클래스 수가 3개 이상인 경우는 클래스 별로 별도의 출력 노드를 할당하고 Max. 법칙에 의해 판단을 내리는데, 이 경우에도 클래스 당 출력 노드 수가 증가되면 성능이 향상되는 것은 마찬가지로 적용될 것이다.

V. 시뮬레이션

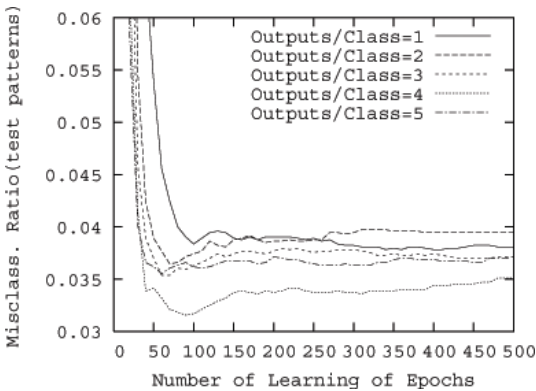


그림 2. 고립단어 인식 문제의 출력노드 수 변경에 따른 인식 성능 변화

제안한 방법의 효용성을 확인하기 위하여 고립단어 인식문제를 MLP에 학습시켰다. 50단어를 9명의 화자가 2번씩 발성한 900개의 음성패턴에서 1,024차원의

ZCPA(zero-crossing peak amplitude) 특징[18]을 추출한 후 MLP에 입력하여 500 epoch 동안 학습시켰다. MLP의 입력층 노드 수는 1,024이며, 중간층 노드는 50개를 사용하였다. 또 다른 7명의 화자가 50단어를 3번씩 발성한 1,050음성패턴을 시험패턴으로 사용하였다. 초기 가중치들은 $[-1 \times 10^{-4}, 1 \times 10^{-4}]$ 에서 균일분포를 지니도록 하였으며, 학습률은 0.05를 사용하였다. 그리고, 클래스 당 출력노드 수는 1, 2, 3, 4, 5개가 되도록 하여 MLP를 학습시키면서, 시험패턴에 대한 오인식률의 변화를 Max. 법칙에 따라 조사하였다. 각각의 경우에 초기 가중치를 다르게 하여 9번의 시뮬레이션을 한 후 그 평균치를 그림으로 그렸다.

[그림 2]에서 보는 바와 같이 기존 방법인 클래스 당 출력노드가 하나인 경우는 시험패턴에 대한 오인식률은 340 epoch에서 3.78%를 얻었다. 한편, 이 논문에서 제안한 방법에 따라 출력노드 수를 증가시켜 얻은 성능으로는 클래스 당 출력노드 수가 2인 경우는 70번째 epoch에서 3.65%, 3인 경우 60번째 epoch에서 3.54%, 4인 경우 90번째 epoch에서 3.16%로 감소하였다. 그렇지만, 클래스당 출력노드가 5인 경우 60번째 epoch에서 3.55%를 얻었다. 출력노드 수를 증가시키면 시험패턴에 대한 오인식률이 [그림 2]에서 보는 바와 같이 클래스당 출력노드가 하나인 경우보다 빨리 감소하며, 그 성능도 우수해짐을 볼 수 있다. 즉, 마치 여러 개의 패턴인식기가 융합되어 학습 성능이 향상되는 효과가 있다. 그렇지만, 출력노드 수를 증가시킨다고 성능이 지속적으로 좋아지는 것은 아니었다. 그 이유는 이론적 유도에서 세운 가정-출력노드가 클래스 내에서 균일분포를 지닌다는 가정 성립되지 않기 때문으로 추정된다.

여기에서 제안한 방법을 CEDAR 데이터 베이스[19]로 구축된 필기체 숫자인식 문제에 적용하여 보았다. CEDAR 데이터 베이스는 18,468개의 학습패턴과 2,213개의 시험패턴으로 구성되어 있으며 각 필기체 숫자는 16레벨의 그레이 값을 가지는 12x12 영상으로 이루어져 있다. 이를 입력이 114개이고 중간층 노드가 30개인 MLP에 클래스 당 출력노드 수를 1부터 5까지 변화시켜가며 시뮬레이션 하였다. MLP의 초기 가중치들은 $[-1 \times 10^{-4}, 1 \times 10^{-4}]$ 에서 균일분포를 가지도록 하였

으며, 학습률은 0.005를 사용하였다. 고립단어 인식문제와 마찬가지로 9번의 시물레이션 후 시험패턴에 대한 오인식률의 평균치를 점검하여 성능이 제일 좋은 시점의 오인식률과 epoch을 조사한 결과, 클래스 당 출력노드가 하나인 경우는 230 epoch에서 4.44%였다. 그렇지만 클래스당 출력노드가 2개인 경우는 80 epoch에서 4.71%, 3 개인 경우는 100 epoch에서 4.92%, 4개인 경우는 50 epoch에서 5.13%, 5개인 경우는 50 epoch에서 4.96%의 데이터를 얻었다. 즉, 기존 방법인 클래스 당 출력노드가 하나인 경우보다 클래스 당 출력노드 수를 증가시킨 경우에 좋은 성능을 얻지 못하였다. III장에서 수학적으로 유도한 모델은 각 클래스가 발생할 확률이 동일하며 출력노드는 각 클래스 내에서 균일분포를 가진다고 가정하였다. 그렇지만, CEDAR 데이터 베이스에 구축된 필기체 숫자인식 문제의 경우, 각 클래스는 발생 확률이 다르기 때문에 이러한 가정에 부합하지 않아서 성능이 향상되지 않은 것으로 여겨진다. 이러한 약점을 보완하기 위해서는 클래스 당 출력노드 수를 각 클래스의 발생확률에 따라 다르게 조정해줄 필요가 있다.

여러 개의 MLP를 학습시킨 후 그 결과를 융합하여 최종 판단을 내리는 인식기와 여기서 제안한 방법의 계산량을 비교해보겠다. 입력이 N 차원이고 중간층 노드가 H 개이며 클래스 수는 M 이라고 하자. 그러면 이 문제를 학습하기 위하여 $N-H-M$ 로 구성된 MLP(입력 N , 중간층 H , 출력 M)는 한 번의 전방향 계산에 식 (2)와 (3)을 수행하기 위하여 $(N+1)H+(H+1)M$ 만큼의 곱하기 연산이 필요하다. 이러한 MLP가 C 개 구성되어 있고 그 출력을 연결하여 최종 판단을 하는 한 층의 $MC-M$ 네트워크가 있다고 하면, 한 번의 전방향 계산에는 전체적으로 $[(N+1)H+(H+1)M]C+(MC+1)M$ 번의 곱하기 연산이 필요하다. 한편, 여기서 제안한 방법은 r 클래스 당 출력노드 수를 C 라고 하면, $N-H-MC$ 구조의 MLP가 구성되므로 한 번의 전방향 연산에서 $(N+1)H+(H+1)MC$ 만큼의 곱하기 연산이 필요하다. 즉, 이 두 방법의 연산량 차이를 구하면, 제안한 방법이 한 번의 전방향 계산에서 $(N+1)H(C-1) + (MC+1)M$ 만큼의 곱하기 연산을 덜 하게 된다.

이제 역방향 계산량을 비교해보겠다. $N-H-M$ 로 구성된 MLP는 $n=4$ 인 경우 (10)의 계산에 $2M$ 만큼의 곱하기 연산이 필요하며, 이를 사용한 (6)의 계산에 $2M(H+1)$ 만큼의 곱하기가 수행된다. 또한, (8)의 계산에 $H(M+N+3)$ 만큼의 곱하기 연산이 있어야 한다. 이러한 MLP가 C 개 구성된 경우 $[2M(H+2)+H(M+N+3)]C$ 번의 곱하기 연산이 한 번의 역방향 계산에 필요하다. 거기에다가 최종 판단을 위하여 구성된 $MC-M$ 네트워크의 역방향 계산에 $2M(MC+2)$ 만큼의 곱하기가 더 필요하다. 한편, 여기서 제안한 $N-H-MC$ 구조의 MLP는 한 번의 역방향 계산에 $2M(H+2)+H(M+N+3)$ 만큼의 곱하기 연산만 있으면 된다. 즉, 역방향 계산에서 제안한 방법이 C 개의 MLP를 구성한 경우 보다 곱하기 연산에서 $[2M(H+2)+H(M+N+3)](C-1)+2M(MC+2)$ 만큼 연산량이 줄어든다.

VI. 결론

이 논문에서는 패턴인식 문제에 MLP를 적용시킬 경우에 성능을 향상시키는 방법으로 클래스 당 출력 노드 수를 증가시키는 것을 제안하였다. 이 방법의 효율성을, 클래스 발생 확률이 동일하며 출력노드는 각 클래스 내에서 균일분포를 가진다는 가정 하에, 확률론적으로 분석하여 증명하였다. 고립단어 음성인식 문제를 시물레이션하여 클래스 당 출력 노드 수가 증가될수록 시험패턴에 대한 오인식률이 감소함을 확인하였다. 여기서 제안한 방법은 하나의 패턴인식기로 충분한 성능을 얻지 못하는 경우, 여러 개의 패턴 인식기를 구현한 후 그 출력을 서로 융합하여 인식성능을 향상시키는 인위적인 융합방법이 아니라, MLP에 단순히 클래스 당 출력노드 수를 증가시켜도 역시 여러 패턴인식기가 융합되어 인식성능이 향상되는 효과를 보임을 확인한 것이다.

그렇지만, 고립단어 인식에서 출력노드가 증가할수록 인식성능이 지속적으로 향상되지 않는 것이 나타났으며, 필기체 숫자인식문제의 경우처럼 클래스의 발생 확률이 다를 경우에는 성능개선을 볼 수 없었다. 이러

한 부분의 약점을 보완하는 연구가 더 필요하다.

참 고 문 헌

- [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] H. White, "Learning in artificial neural networks: a statistical perspective," *Neural Computation*, Vol.1, pp.425-464, 1989.
- [3] 오상훈, "다층퍼셉트론의 잡음강건성 분석 및 향상 방법," 한국콘텐츠학회 논문지, 제9권, 제1호.
- [4] 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.
- [5] A. van Ooyen and B. Nienhuis, "Improving the convergence of the back-propagation algorithm," *Neural Networks*, Vol.5, pp.465-471, 1992.
- [6] S.-H. Oh, "Improving the error back-propagation algorithm with a modified error function," *IEEE Trans. Neural Networks*, Vol.8, pp.799-803, 1997.
- [7] S.-H. Oh and S.-Y. Lee, "An adaptive learning rate with limited error signals for training of multilayer perceptrons," *ETRI Journal*, Vol.22, No.3, pp.10-18, 2000(9).
- [8] 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, 1990(6).
- [9] B. B. Chaudhuri and U. Bhattacharya, "Efficient training and improved performance of multilayer perceptron in pattern classification," *Neurocomputing*, Vol.34, pp.11-27, 2000.
- [10] K.-Y. Park and S.-Y. Lee, "Out-of vocabulary rejection based on selective attention model," *Neural Processing Letters*, Vol.12, pp.41-48, 2000.
- [11] F. J. Owens, G. H. Zheng, and D. A. Irvine, "A multi-output-layer perceptron," *Neural Computation & Applications*, Vol.4, pp.10-20, 1996.
- [12] 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.
- [13] 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.
- [14] R. Paris, E. D. Di Claudio, G. Orlandi, "A generalized learning paradigm exploiting the structure of feedforward neural networks," *IEEE Trans. Neural Networks*, Vol.7, pp.1450-1459, 1996.
- [15] C. Yu, M. T. Manry, J. Li, and P. L. Narasimha, "An efficient hidden layer training method for multilayer perceptron," *Neurocomputing*, Vol.70, pp.525-535, 2006.
- [16] 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.
- [17] J.-H. Jeong, H. Kim, D.-S. Kim, and S.-Y. Lee, "Speaker adaptation based on judge neural networks for real world implementations of voice-command systems," *Information Science*, Vol.123, pp.13-24, 2000.
- [18] D.-S. Kim, S.-Y. Lee, and R.-M. Kil, "Auditory processing of speech signals for robust speech recognition in real-worldnoisy environments," *IEEE Trans. Speech and Audio Processing*,

Vol.7, pp.55-69, 1999.

- [19] J. J. Hull, "A database for handwritten text recognition research," IEEE Trans. Pat. Ana. Mach. Int., Vol.16, pp.550-554, 1994.

저 자 소 개

오 상 훈(Sang-Hoon Oh)

종신회원



- 1986년 2월 : 부산대학교 전자공학과(공학사)
 - 1988년 2월 : 부산대학교 대학원 전자공학과(공학석사)
 - 1999년 8월 : 한국과학기술원 전기 및 전자공학과(공학박사)
 - 1988년 1월 ~ 1989년 12월 : LG 반도체(주) 사원
 - 1990년 1월 ~ 1998년 6월 : 한국전자통신연구원 기초기술연구부 및 이동통신기술연구소 선임연구원
 - 1999년 8월 ~ 2000년 3월 : 한국과학기술원 뇌과학 연구센터 연구원
 - 2000년 4월 ~ 2000년 10월 : 일본 RIKEN, Brain Science Institute, Research Scientist
 - 2000년 10월 ~ 2001년 10월 : (주)엑스텔테크놀로지 연구소장
 - 2001년 11월 ~ 2002년 2월 : 한국과학기술원 초빙교수
 - 2002년 3월 ~ 현재 : 목원대학교 정보통신공학과 부교수
 - 2008년 8월 ~ 현재 : 조지아공대 College of Computing, Div. Computational Science and Eng. 방문교수
- <관심분야> : 지능정보처리 알고리즘 개발 및 IT에의 응용, 독립성분분석, NMF, 패턴인식, 음성신호 처리