

다층퍼셉트론의 계층적 구조를 통한 성능향상

Hierarchical Architecture of Multilayer Perceptrons for Performance Improvement

오상훈

목원대학교 정보통신공학과

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

요약

다층퍼셉트론이 충분한 중간층 노드 수를 지니면 임의의 함수를 근사시킬 수 있다는 이론적 연구결과에 기초하여, 다층퍼셉트론을 실제 문제에 응용하는 경우에 일반적으로 입력층, 중간층, 출력층으로 이루어진 3층 구조의 다층퍼셉트론을 사용한다. 그렇지만, 이러한 구조의 다층퍼셉트론은 입력벡터가 여러 가지 성질로 이루어진 복잡한 문제의 경우 좋은 일반화 성능을 보이지 않는다. 이 논문에서는 입력 벡터가 여러 가지 정보를 지닌 데이터들로 구성되어 있는 문제인 경우에 계층적 구조를 지닌 다층퍼셉트론의 구성으로 성능을 향상시키는 방법을 제안한다. 즉, 입력데이터를 서브-벡터로 구분한 후 서브-벡터별로 다층퍼셉트론을 적용시키며, 이 서브-벡터별로 적용된 하위층 다층퍼셉트론으로부터 인식 결과를 받아서 최종 결정을 하는 상위 다층퍼셉트론을 구현한다. 제안한 방법의 효용성은 단백질의 구조를 예측하는 문제를 통하여 확인한다.

■ 중심어 : | 다층퍼셉트론 | 계층적 구조 | 입력 벡터 |

Abstract

Based on the theoretical results that multi-layer feedforward neural networks with enough hidden nodes are universal approximators, we usually use three-layer MLP's(multi-layer perceptrons) consisted of input, hidden, and output layers for many application problems. However, this conventional three-layer architecture of MLP shows poor generalization performance in some applications, which are complex with various features in an input vector. For the performance improvement, this paper proposes a hierarchical architecture of MLP especially when each part of inputs has a special information. That is, one input vector is divided into sub-vectors and each sub-vector is presented to a separate MLP. These lower-level MLPs are connected to a higher-level MLP, which has a role to do a final decision. The proposed method is verified through the simulation of protein disorder prediction problem.

■ keyword : | Multilayer Perceptrons | Hierarchical Structure | Input Vector |

I. 서론

다층퍼셉트론이 선형적으로 분리 가능한 문제만을 학습할 수 있다는 단점이 지적된 이후 신경회로망의 연

구가 한동안 침체기를 겪었지만, 이의 한계를 극복한 다층퍼셉트론(MLP : multi-layer perceptron) 구조의 제안은 신경회로망 분야 연구를 활발하게 만들었다. 특히, 오류역전파(EBP: error back-propagation) 알고리

즘은 MLP의 학습 알고리즘으로 널리 사용되었다[1]. 여기에, Hornik 등은 MLP와 같은 전방향 신경회로망 구조에서 중간층 노드의 수가 충분하다면 임의의 함수를 근사시킬 수 있다는 수학적 증명을 발표하였다[2]. 또한, 학습에 의해 외부 환경에 강건한 신경회로망이 구축된다는 것도 시뮬레이션 및 수학적 분석을 통하여 입증되었다[3-7]. 이러한 수학적 배경을 기반으로 전방향 신경회로망은 패턴인식[8], 음성인식[9], 시계열예측[10], 데이터 마이닝[11], 바이오인포메틱스[12] 등 다양한 분야에 응용되고 있다.

Hornik 등이 증명한 이론적 토대[2]를 기반으로 MLP는 일반적으로 입력층, 중간층, 그리고 출력층으로 이루어진 3층 구조를 지니도록 하여 다양한 문제에 응용되었다. 이때, 학습방법의 개선을 통하여 성능을 향상시키려는 시도가 있었다[13-16].

한편, 일반적인 3층 구조 MLP와 다른 형태로 구조를 변경하여 성능을 개선하고자 하는 목적으로, 출력 노드의 숫자를 증가시키는 방법[17], 콘볼루션 구조 방법[18], TDNN(time-delayed neural network) 방법[14], 다층출력 퍼셉트론[23], 앙상블 방법[19] 등이 있다.

패턴인식 문제에서 각 출력노드는 하나의 클래스에 대응하도록 할당이 되는데, 출력노드 수를 증가시키는 방법[17]은 하나의 클래스에 대응하는 출력노드 수를 여러 개로 증가시켜 성능을 향상시키는 방법이다.

콘볼루션 구조 방법[18]은 입력층에 주어지는 벡터 혹은 영상을 여러 영역으로 나눈 후, 하나의 영역에 대하여 가중치 합을 계산하는 가중치 벡터들을 다른 영역에도 같이 적용하는 방법을 사용한다. 즉, 마치 영상을 여러 영역으로 나눈 후, 한 영역에 대하여 여러 가지 특징들을 추출하는 가중치 벡터들을 다른 영역에도 똑같이 사용한다는 개념을 도입한 것이다. 이러한 형태의 데이터 처리 과정이 여러 계층을 거치며 이루어지게 하는 방식이어서 일반적인 방법으로는 학습이 잘 이루어지지 않는다[22] TDNN[14]은 콘볼루션 방법을 음성인식에 적용하는 구조로 제안되었다.

다층출력 퍼셉트론[23]은 일반적인 구조의 MLP를 주어진 문제에 대하여 학습시킨 후 성능이 만족스럽지 못한 경우에 각 출력노드마다 상위의 출력노드를 추가

적으로 할당하는 방법이다. 추가된 출력노드는 하위층에 있는 모든 노드들과 가중치로 연결되며 일반적인 EBP에 따라 학습된다.

앙상블 방법[19]은 동일한 문제를 학습시키기 위하여 동일한 구조의 MLP를 여러 개 구현하여 학습패턴으로 학습시킨 후, 최종적으로 여러 개의 MLP로부터 얻은 출력을 활용하여 더 나은 결과를 얻는 방법이다. 이때, 학습패턴 전체를 여러 개의 sub-set으로 구분하여 여러 MLP들이 다른 학습패턴 sub-set으로 학습되도록 한다.

이 논문에서는 입력 벡터를 특정한 의미를 지닌 부분으로 분할하여, 각 분할된 영역을 다루는 MLP와 최종 결과를 출력해주는 MLP로 이루어지는 계층적 구조를 사용하여 성능을 향상시키는 방법을 제안한다. 즉, 입력층에 주어지는 벡터가 여러 가지의 성질 혹은 정보를 지닌 경우에 입력벡터 전체를 대상으로 일반적인 3층 구조의 MLP를 구성하기 보다는, 각 성질 혹은 정보에 해당하는 입력별로 3층 구조의 MLP를 각각 별도로 구성하도록 한다. 이렇게 구성된 각 입력 성질별 MLP에서 출력되는 값들을 상위층의 MLP에서 입력받아 최종 출력값을 나타내는 계층적 구조를 지니도록 한다.

이 논문의 구성은 다음과 같다. 먼저 II 장에서 MLP의 일반적인 구조와 학습방법에 대하여 설명한다. III 장에서 이 논문에서 제안하는 계층적 구조의 MLP를 설명하고, IV장에서 시뮬레이션으로 제시한 방법의 효용성을 확인한다. 마지막으로 V장에서 결론을 맺는다.

II. 일반적인 구조의 MLP와 학습방법

일반적으로 입력층, 중간층, 출력층으로 구성된 3층 구조의 MLP는 [그림 1]과 같다. 여기서, 입력층은 N 개의 입력 노드 x , 중간층은 H 개의 중간층 노드 h , 그리고 출력층은 M 개의 출력 노드 y 로 구성되어 있다고 하자. 이를 간략히 $N-H-M$ 구조 MLP라 한다. 어떤 N 차원의 입력패턴 $x = [x_1, x_2, \dots, x_N]$ 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

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

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

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

로 주어진다.

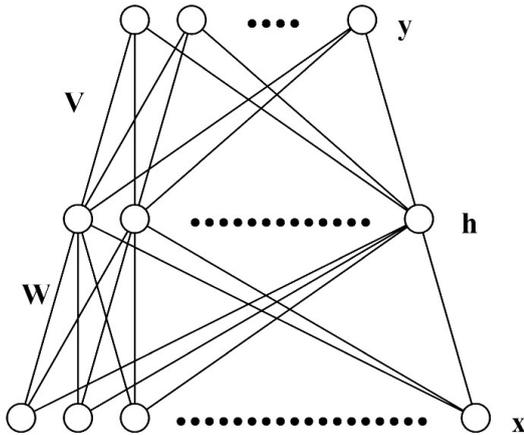


그림 1. 일반적인 MLP(Multi-layer Perceptron) 구조

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 일반적으로

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

로 주어지는 MSE(mean-squared error)를 최소화시키도록 MLP의 가중치들이 변경된다[1]. 패턴인식 문제에서 출력층 목표벡터의 각 요소 값들은

$$t_k^{(p)} = \begin{cases} +1, & \text{if } \mathbf{x}^{(p)} \in C_k \\ -1, & \text{otherwise} \end{cases} \quad (6)$$

와 같이 주어지며, 여기서, C_k 는 클래스 k 에 속하는 패턴들의 집합을 나타낸다. 그리고, 식 (6)과 같은 목표값 코딩에서, 목표값이 '+1'로 주어지는 출력노드를 목표노드라 한다. 예를 들면, 입력패턴이 클래스 k 에 속하면 k 번째 출력노드가 목표노드가 된다.

그렇지만, 식 (5)로 주어진 MSE를 최소화시키기 위한 EBP(Error Back-Propagation) 학습은 학습속도가 느리다는 단점이 많이 지적되었다[20]. 이의 해결책으로 CE(Cross-Entropy) 오차함수[15]가 제안되었으며, 이 보다 더 나은 성능을 지닌 nCE(n-th order extension of CE) 오차함수[13] 역시 제안되었다. nCE 오차함수는

$$E_{nCE}^{out} = - \sum_{p=1}^P \sum_{k=1}^M \int \frac{t_k^{(p)n+1} (t_k^{(p)} - y_k^{(p)})^n}{2^{n-2} (1 - y_k^{(p)})^2} dy_k^{(p)} \quad (7)$$

와 같이 주어지며, MLP의 출력층 가중치는

$$\Delta v_{kj} = -\eta \frac{\partial E_{nCE}^{out}}{\partial v_{kj}} = \eta \delta_k^{(p)} h_j^{(p)} \quad (8)$$

에 따라 학습된다. 여기서, η 는 학습률이며,

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

이다. 또한, 중간층 가중치는

$$\Delta w_{ji} = -\eta \frac{\partial E_{nCE}^{out}}{\partial w_{ji}} = \eta f'(h_j^{(p)}) x_i^{(p)} \sum_{k=1}^M v_{kj} \delta_k^{(p)} \quad (10)$$

에 따라 학습된다. 이렇게 nCE를 최소화 시키는 방식의 학습에 의해 얻어지는 성능이 MSE를 사용한 경우보다 좋은 이유는 출력노드의 부적절한 포화 현상을 방지할 수 있기 때문이다[13].

한편으로, 패턴수가 적은 “소수 클래스(클래스 1)” 데이터와 상대적으로 패턴수가 아주 많은 “다수 클래스(클래스 2)” 데이터가 주어질 때 두 클래스(two-class) 패턴인식 문제-소위, 데이터 불균형 문제-의 경우에는 위와 같은 학습방법으로 좋은 학습결과를 얻지 못한다. 이 경우에는 출력노드를 2개로 설정하여

$$E_{nm}^{out} = - \sum_{p=1}^P \left[\int \frac{t_1^{(p)^{n+1}} (t_1^{(p)} - y_1^{(p)})^n}{2^{n-2} (1 - y_1^{(p)})^2} dy_1^{(p)} \right. \\ \left. + \int \frac{t_2^{(p)^{m+1}} (t_2^{(p)} - y_2^{(p)})^m}{2^{m-2} (1 - y_2^{(p)})^2} dy_2^{(p)} \right] \quad (11)$$

와 같이 제안된 오차함수를 사용하면 된다[16]. 여기서, n 은 소수 클래스 목표노드에 적용되는 nCE오차함수의 차수이며, m 은 다수 클래스 목표노드에 적용되는 nCE 오차함수의 차수이다($n < m$). 식 (11)의 $y_k^{(p)}$ 에 대한 미분으로 출력층 노드의 $\delta_k^{(p)}$ 를 계산하면

$$\delta_k^{(p)} = \begin{cases} \frac{t_k^{(p)^{n+1}} (t_k^{(p)} - y_k^{(p)})^n}{2^{n-1}}, & \text{if } k = 1 \\ \frac{t_k^{(p)^{m+1}} (t_k^{(p)} - y_k^{(p)})^m}{2^{m-1}}, & \text{if } k = 2 \end{cases} \quad (12)$$

와 같이 된다. 그리고, 각 $\delta_k^{(p)}$ 에서 목표값이 '+1'인 경우와 '-1'인 경우의 수가 불균형인 것을 해소하기 위하여

$$\gamma = \frac{M_1}{M_2} \quad (13)$$

의 비율만큼 목표값이 '-1'인 $\delta_1^{(p)}$ 와 목표값이 '+1'인 $\delta_2^{(p)}$ 를 조정한다. 즉,

$$\delta_1^{(p)} \rightarrow \begin{cases} \delta_1^{(p)}, & \text{if } t_1^{(p)} = 1 \\ \gamma \delta_1^{(p)}, & \text{if } t_1^{(p)} = -1 \end{cases} \quad (14)$$

와

$$\delta_2^{(p)} \rightarrow \begin{cases} \gamma \delta_2^{(p)}, & \text{if } t_2^{(p)} = 1 \\ \delta_2^{(p)}, & \text{if } t_2^{(p)} = -1 \end{cases} \quad (15)$$

같이 변경된다. 여기서, M_1 은 소수 클래스의 학습패턴 수이며, M_2 는 다수 클래스의 학습패턴 수이다. 따라서, 소위 데이터 불균형 문제에서는 식 (14)와 (15)로 얻어진 것을 식 (8)과 (10)에 대입하여 가중치들을 학습시킨다[16].

III. 계층적 구조를 지닌 MLP

[그림 2]는 [그림 1]과 같이 구성된 입력층, 중간층, 출력층으로 구성된 3층($N-H-M$) 구조 MLP의 개략도이다. 이러한 구조의 MLP는 이론적으로 임의의 함수

를 근사시킬 수 있다는 것이 증명되었다[2]. 이를 토대로, 입력으로 주어지는 벡터를 대상으로 [그림 1]과 같은 형태의 MLP를 구성하는 접근 방법이 사용되었다. 그렇지만, 실제 문제의 응용에 있어서 원하는 만큼의 학습 성능을 얻지 못하는 경우가 있다. 특히, 입력 벡터가 하나의 성질을 지니는 것이 아니라 여러 가지 성질을 지니는 복잡한 문제의 경우에는 [그림 2]와 같은 형태의 MLP는 학습 후 얻게 되는 성능이 만족스럽지 못하다.

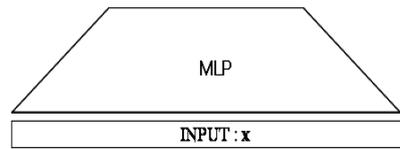


그림 2. 일반적 구조 MLP의 개략도

따라서, 여기서는 입력벡터가 여러 가지(R 가지) 성질을 지닌 경우에 [그림 3]과 같은 형태를 지니는 계층적 구조의 MLP를 제안한다. 이 계층적 구조의 MLP는 먼저 N 차원의 입력벡터 \mathbf{x} 를 R 가지 성질에 따라 R 개의 sub-vector로 분리한다. 이 sub-vector들을 각각 $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_R$ 이라 하면,

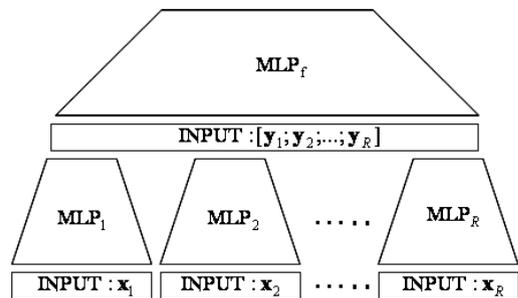


그림 3. 제안하는 계층적 구조 MLP의 개략도

$$\mathbf{x} = [\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_R] \quad (16)$$

와 같이 표현된다. 이때, $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_R$ 는 각각 N_1, N_2, \dots, N_R 차원의 벡터들이며

$$\sum_{v=1}^R N_v = N \quad (17)$$

이다. 그 다음에 x_1, x_2, \dots, x_R 를 입력층으로 사용하는 3층 구조 MLP들을 구성하고 이들을 각각 $MLP_1, MLP_2, \dots, MLP_R$ (하위층 MLP들)이라 칭한다. 즉, 하위층 MLP들은 전체의 입력 벡터 중 특정 성질에 해당하는 부분만을 입력으로 사용하여 학습이 진행된다. 이 하위층 MLP들의 출력값들을 각각 y_1, y_2, \dots, y_R 이라 하면, 이들은 그 다음 단계에 있는 3층 구조 MLP_f (상위층 MLP)의 입력으로 사용되어, 최종적으로 MLP_f 의 출력값이 목표값이 나오도록 학습이 이루어진다.

M class 패턴인식 문제의 경우 [그림 2]에 해당하는 $N-H-M$ 구조 MLP의 학습방법은 II장에서 설명한 바와 같다. 이제, [그림 3]에 해당하는 계층적 MLP에서 M class 패턴인식 문제를 학습시키는 방법을 설명하겠다. $MLP_1, MLP_2, \dots, MLP_R$ 의 구조를 각각 $N_1-H_1-M, N_2-H_2-M, \dots, N_R-H_R-M$ 이라 하자. 그러면, 하위층 MLP들의 출력 y_1, y_2, \dots, y_R 은 각각 M 차원의 벡터이므로 MLP_f 에 입력되는 노드 수는 $M \times R$ 이다. 따라서, MLP_f 의 구조는 $M \times R - H_f - M$ 이라 한다.

이렇게 구성된 계층적 MLP들의 가중치들이 먼저 임의의 값으로 초기화된 이후에[20], 하위 층의 MLP들이 II장에서 설명된 것처럼 목표값 $t^{(p)} = [t_1^{(p)}, t_2^{(p)}, \dots, t_M^{(p)}]$ 를 사용하여 각각 학습이 된다. 그 다음에 하위층 MLP들로부터 출력된 벡터들 y_1, y_2, \dots, y_R 을 $M \times R$ 차원 벡터로 구성하여 MLP_f 에 입력시킨 후, MLP_f 역시 $t^{(p)} = [t_1^{(p)}, t_2^{(p)}, \dots, t_M^{(p)}]$ 이 출력되도록 II장에서 설명된 것과 동일한 방법으로 학습이 된다. 이를 매 학습패턴마다 반복하여 전체 학습패턴에 대한 학습이 한번 이루어진 것을 epoch 단위로 표현한다. 여러 번의 epoch을 거치면서 가중치들을 지속적으로 변경시키며 필요에 따라 성능을 측정한다. 한편, 경우에 따라서는 $MLP_1, MLP_2, \dots, MLP_R$ 중에서 성능이 나쁜 하위층 MLP는 제거하여 전체적인 구조를 간략화 할 수도 있다.

IV. 시뮬레이션

1. 시뮬레이션 대상 문제

제안한 방법의 효용성을 확인하기 위하여 복잡하면서도 입력 벡터가 다양한 특징들로 이루어진 문제가 필요하다. 이를 위하여, 고등과학원의 주기형 박사로부터 96개의 단백질에서 데이터를 추출한 “단백질 disorder 예측 문제”를 제공받았다. 이 문제에서는 단백질에서 추출된 특성벡터를 근거로 order (class 0) 혹은 disorder (class 1)에 해당하는 것을 예측하는 two-class 문제이다. 단백질에서 특징 추출 시 sliding window를 적용하였으며, window size는 27이다.

단백질에서 추출된 특징 벡터는 21×27 차 아미노산 배열 프로파일(amino acid sequence profile), 3×27 차 2차 구조 프로파일(secondary structure profile), 3×27 차 용해도 프로파일(solvent accessibility profile), 27차 소수성 프로파일(hp: hydrophobicity profile), 평균 hp, 전체 hp로 이루어진 758차원 벡터이다. 이 벡터가 19816개로 이루어진 데이터베이스에서, disorder에 해당하는 것은 1189개로 6%에 해당한다.

2. 학습 및 성능비교 방법

일반적인 구조를 지닌 MLP와 이 논문에서 제안하는 계층적 구조의 MLP를 단백질 disorder 예측 문제의 시뮬레이션으로 비교하였다. 먼저 [그림 1]과 같은 일반적인 3층 구조의 MLP는 입력노드 758개, 중간층 노드 20개, 출력노드 2개(758-20-2 구조)로 구성하였다. 초기 가중치는 $[-1 \times 10^{-4}, 1 \times 10^{-4}]$ 에서 균일분포가 되도록 임의로 설정하였다. 그리고, II장에서 설명한 데이터 불균형 문제에 적합한 학습 방법을 적용하여 수식 (11)에서 $n=2, m=8$ 의 파라미터 값으로 학습시켰다. 학습률은 $0.001 \times [(n+1) + (m+1)]/2$ 을 사용하였다.

단백질 disorder 예측 문제의 학습을 위한 데이터베이스가 758차원을 지닌 19816개의 벡터로 이루어져 있는데, 시험패턴과 학습패턴의 구분이 없으므로 일반화 성능을 측정하기 위하여 “1-out of-5 교차확인(cross validation) 방법”을 사용한다. 여기서, “1-out of-5 교차확인 방법”이란 주어진 데이터를 임의의 5개 집합으

로 구분한 후, 한 집합에 속하는 데이터는 시험패턴으로 사용하고 나머지 집합의 모든 데이터는 학습패턴으로 사용하는 방법을 각각 다른 시험패턴 집합을 대상으로 5번에 걸쳐 실시한 후, 얻어진 데이터의 평균으로 일반화 성능(혹은 시험패턴 성능)을 측정하는 방법이다.

그 다음으로 이 논문에서 제안한 [그림 3]과 같은 계층적 구조를 지닌 MLP에 동일한 문제를 학습시켰다. 이 경우 758차원 입력 벡터를 567차원 아미노산배열 프로파일, 81차원 2차 구조 프로파일, 81차원 용해도 프로파일, 27+2차원 hp 프로파일 부분으로 구분하였다 ($R=4$). 여기에 MLP_1 은 567-5-2 구조, MLP_2 와 MLP_3 는 각각 81-5-2 구조, 그리고 MLP_4 는 29-5-2 구조가 되도록 하여 하위층 MLP들 (MLP_1, \dots, MLP_4)의 중간층 노드 수를 합하면 20이 되도록 하였다. 그리고, 상위층 MLP(MLP_f)는 하위층에서 출력된 결과를 입력으로 받아서 8-5-2구조로 정하였다.

일반적인 구조 MLP와 이 논문에서 제안한 계층적 구조 MLP의 성능을 비교할 때, 단백질 disorder 예측 문제는 데이터 불균형이 심한 문제이므로 일반적인 인식률로는 그 성능을 제대로 반영할 수 없다. 따라서, 데이터 불균형 문제에서 많이 사용되는 G-Mean (Geometric Mean)을 성능비교 기준으로 정하였다[16]. 여기서, G-Mean은 다수 클래스 인식률과 소수 클래스 인식률을 구한 다음 이들의 기하평균을 구한 것을 뜻한다. 또한, 데이터 불균형 문제에서 일반적으로 소수 클래스의 인식률이 저하되어 나타나므로 소수 클래스의 인식률도 같이 비교하여 보았다.

3. 시뮬레이션 데이터 분석

“단백질 disorder 예측 문제”를 “1-out of-5 교차확인 방법”으로 일반적인 구조 MLP와 제안한 계층적 구조 MLP에 학습시켜 비교하였다. 여기서, MLP는 초기 가중치에 따라 성능이 영향을 받으므로, 9가지의 다른 초기 가중치를 가지는 경우에 대하여 학습시켰다. 즉, 5가지 시험패턴 가지 수와 9가지 초기가중치에 대한 시뮬레이션이므로 총 45가지의 경우에 대하여 학습 시뮬레이션을 하였다. 이때, 5000 epoch 동안 학습시키면서

매 10 epoch마다 시험패턴을 대상으로 성능을 측정한다. 시험패턴 인식률은 시험패턴을 MLP에 입력 후 출력노드 값을 구하여, 큰 출력 값을 지닌 출력노드가 목표노드에 해당하면 인식된 것으로 판정하여 측정한다. 이렇게 학습과정 중에 매 10 epoch마다 얻어진 시험패턴에 대한 성능 데이터를 45가지 경우에 대하여 평균을 구한 후, 가장 좋은 결과를 보이는 epoch 위치의 데이터를 [표 1]에 기록하였다.

표 1. 단백질 disorder 예측 문제의 학습 결과[단위:%]

	Conv. MLP	MLP_1	MLP_2	MLP_3	MLP_4	MLP_f
소수 클래스 인식률	76.53	72.34	68.27	75.29	72.78	78.47
G-Mean	76.33	75.72	70.92	74.89	68.57	78.53

먼저 [표 1]에서 Conv. MLP는 일반적인 구조 MLP를 뜻하는데, 소수클래스 인식률은 76.53%, G-Mean은 76.33%를 얻었다. 이는 단백질 disorder 예측 문제가 상당히 복잡하고 어려운 문제이기 때문이다.

그 다음으로 제안한 계층적 구조 MLP를 이용한 학습 결과는 하위층 MLP들($MLP_1, MLP_2, MLP_3, MLP_4$) 각각에 대한 것과 상위층 MLP(MLP_f)에 대하여 측정하였다. 하위층 MLP_1 은 소수클래스 인식률과 G-Mean이 각각 72.34%와 75.72%를 얻었다. Conv. MLP보다 저하된 결과를 보이는 데, Conv. MLP는 758차원 입력 벡터 전체를 학습한 것인 반면에 MLP_1 은 567차원 아미노산배열 프로파일만을 학습한 것이기 때문이다. 81차원 2차 구조 프로파일을 학습한 MLP_2 는 소수클래스 인식률 68.27%, G-Mean 70.92%를 얻어 상당히 저하된 성능을 얻었다. 81차원 용해도 프로파일을 학습한 MLP_3 는 소수클래스 인식률 75.29%, G-Mean 74.89%를 얻었으며, 29차원 hp 프로파일을 학습한 MLP_4 는 소수클래스 인식률 72.78%, G-Mean 68.57%를 얻었다. 이 결과를 보면 758차원 입력 벡터 중 어느 프로파일을 학습하였느냐에 따라 성능이 상당히 다르게 됨을 볼 수 있다. 즉, MLP_1 과 MLP_3 가 MLP_2 와

MLP_4 보다 좋은 성능을 보이는데, 이는 각 프로파일 이 다른 성질을 지녔음을 간접적으로 나타낸다.

하위층 MLP들의 출력값들을 받아서 학습한 MLP_f 는 소수클래스 인식률이 78.47%, G-Mean이 78.53%를 얻어 Conv. MLP 보다 향상된 결과를 보였다. 즉, 758차 원의 입력 전체를 하나의 MLP에 학습시킨 경우보다 제한한 형태로 각 입력의 성질별로 별도의 하위층 MLP들을 구축하여 학습시킨 후 그 출력을 상위층 MLP에서 학습시키는 것이 성능을 향상시켰음을 나타낸다. 물론, Conv. MLP보다 MLP_1 , MLP_2 , MLP_3 , MLP_4 가 성능이 저하되었지만 이는 부분적인 프로파일 만을 학습에 사용하였기 때문이다. 그렇지만, 각 프로파일에 맞추어 학습된 하위층 MLP들의 결과를 활용하여 상위층 MLP가 Conv. MLP 보다 더 좋은 성능을 보이도록 하는 효과가 있음을 확인하였다.

MLP_f 와 하위층 MLP들 간의 관계를 더 확인해보기 위하여, 하위층 MLP들 중에서 좋은 성능을 보인 MLP_1 과 MLP_3 만을 상위층 MLP에 연결시키는 구조로 학습한 결과 소수클래스 인식률은 77.82%, G-Mean은 78.10%를 얻었다. 이 결과는 [표 1]에 나와 있는 MLP_f 보다는 다소 낮지만 Conv. MLP보다는 좋다. 즉, 하위층의 MLP_1 과 MLP_3 는 상위층에 대한 기여도가 높은 반면에, MLP_2 와 MLP_4 는 상위층에 대한 기여도가 낮다고 볼 수 있다.

이제 각 구조별 복잡도를 비교해보자. MLP의 복잡도는 그 MLP를 구성하고 있는 바이어스를 포함한 가중치의 수로 비교할 수 있다. 758-20-2 구조의 Conv. MLP는 $759 \times 20 + 21 \times 2 = 15222$ 개의 가중치를 지녔다. 제한한 계층적 구조 MLP는 MLP_1 이 2852개, MLP_2 와 MLP_3 가 각각 422개, MLP_4 가 162개의 가중치를 지녔으며, 상위층의 MLP_f 가 57개의 가중치를 지니 총합 3915개의 가중치를 지녔다. 이는 Conv. MLP가 지닌 가중치의 25.7%에 해당한다. MLP_1 과 MLP_3 만을 MLP_f 에 연결시킨 계층적 구조에서는 가중치의 개수가 3311개로 더 감소한다. 즉, 제한한 방법은 MLP의 복잡도는 줄이면서 더 좋은 성능을 얻게 해준다.

주어진 입력벡터를 sub-vector로 분리하는 것은 적어도 문제의 복잡도가 상승하지는 않을 것이다. 따라서, 계층적 구조가 일반적인 구조보다 가중치의 숫자가 적으므로, 학습에 필요한 시간은 줄어들게 될 것이다.

한편, 입력데이터에 feature extraction이나 feature selection을 적용하여 가공한 후 MLP에 적용하여 성능을 향상시키는 방법도 있다[21]. 이 분야는 또 다른 연구영역이며, 이 논문은 입력데이터를 특별한 가공 없이 MLP에 적용시키는 방법을 제안한 것이다.

IV. 결론

이 논문에서는 입력벡터가 여러 가지 성질을 지닌 복잡한 문제의 경우, 일반적인 구조의 MLP를 구성하여 학습을 시키는 것 보다 계층적 MLP를 사용하면 성능이 향상되는 것을 제안하였다. 제안한 계층적 구조 MLP는 각 성질별로 입력을 받아서 학습을 하는 하위층 MLP들과 하위층 MLP들의 출력을 받아서 입력으로 활용하는 상위층 MLP로 구성된다.

“단백질 disorder 예측 문제”를 대상으로 일반적인 구조 MLP와 제안한 계층적 구조 MLP의 시뮬레이션으로 성능을 비교한 결과, 제안한 방법이 더 좋은 성능을 보였다. 또한, 제안한 계층적 구조가 일반적인 구조 MLP 보다 복잡도가 많이 감소함을 보였다. 특히, 하위층 MLP들이 각 입력 성분의 특징에 따라 다른 성능을 보이며, 상위층 MLP의 성능에 대한 기여도가 다르게 나타남을 확인하였다.

참고 문헌

- [1] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, The MIT Press, 1986.
- [2] K. Hornik, M. Stincombe, and H. White, "Multilayer feedforward networks are universal

- approximators," *Neural Networks*, Vol.2, pp.359-366, 1989.
- [3] M. Stevenson, R. Winter, and B. Widrow, "Sensitivity of feedforward neural networks to weight errors," *IEEE Trans. Neural Networks*, Vol.1, pp.71-90, 1990.
- [4] Y. Xie and M. A. Jabri, "Analysis of the effects of quantization in multilayer neural networks using a statistical model," *IEEE Trans. Neural Networks*, Vol.3, pp.334-338, 1992.
- [5] J. Y. Choi and C.-H. Choi, "Sensitivity analysis of multilayer perceptron with differentiable activation functions," *IEEE Trans. Neural Networks*, Vol.3, pp.101-107, 1992.
- [6] Y. Lee and S.-H. Oh, "Input noise immunity of multilayer perceptrons," *ETRI Journal*, Vol.16, pp.35-43, 1994.
- [7] S.-H. Oh and Y. Lee, "Sensitivity analysis of single hidden-layer neural networks with threshold functions," *IEEE Trans. Neural Networks*, Vol.6, pp.1005-1007, 1995.
- [8] R. P. Lippmann, "Pattern classification using neural networks," *IEEE Communication Magazine*, pp.47-64, 1989.
- [9] 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.
- [10] A. S. Weigend and N. A. Gershenfeld, *Time Series Prediction: Forecasting the future and understanding the past*, Addison-Wesley Publishing Co., 1994.
- [11] Y.-M. Huang, C.-M. Hung, and H. C. Jiau, "Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem," *Nonlinear Analysis*, Vol.7, pp.720-747, 2006.
- [12] Z. R. Yang and R. Thomson, "Bio-basis function neural network for prediction of protease cleavage sites in proteins," *IEEE Trans. Neural Networks*, Vol.16, pp.263-274, 2005.
- [13] S.-H. Oh, "Improving the error back-propagation algorithm with a modified error function," *IEEE Trans. Neural Networks*, Vol.8, pp.799-803, 1997.
- [14] 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.
- [15] A. van Ooyen and B. Nienhuis, "Improving the convergence of the back-propagation algorithm," *Neural Networks*, Vol.5, pp.465-471, 1992.
- [16] 오상훈, "다층퍼셉트론에 의한 불균형 데이터의 학습방법", 한국콘텐츠학회 논문지, 제9권, 제7호, pp.141-148, 2009.
- [17] 오상훈, "다층퍼셉트론의 출력노드 수 증가에 의한 성능향상", 한국콘텐츠학회 논문지, 제9권, 제1호, pp.123-130, 2009.
- [18] D. Simard, P. Y. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks," *Proc. Int. Conf. Document Analysis and Recognition(ICDAR)*, Washington DC, USA, pp.958-962, 2003.
- [19] Z.-H. Zhou and X.-Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Trans. Knowledge and Data Eng.*, Vol.18, No.1, pp.63-77, 2006.
- [20] 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.

[21] Sang-Hoon Oh "On the Design of Multilayer perceptrons for Pattern Classifications," Proc. Int. Conf. on Convergence Content 2009, Hanoi, Vietnam, pp.59-62, Dec. 17-19 2009.

[22] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, Vol.313, pp.504-507, 2006.

[23] F. J. Owens, G. H. Zheng, and D. A. Irvine, "A multi-output-layer perceptron," Neural Computing & Applications, Vol.4, pp.10-20, 1996.

<관심분야> : 지능정보처리 알고리즘 개발 및 IT에의 응용, 독립성분분석, NMF, 패턴인식, 음성신호 처리, 바이오 인포매틱스

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 ~ 2009년 8월 : 조지아공대 College of Computing, Div. Computational Science and Eng. 방문교수