

다층신경망의 학습능력 향상을 위한 학습과정 및 구조설계

이대식* · 이종태**

A multi-layered neural network learning procedure and
generating architecture method for improving neural
network learning capability

Daesik Lee* · Jongtae Rhee**

■ Abstract ■

The well-known back-propagation algorithm for multi-layered neural network has successfully been applied to pattern classification problems with remarkable flexibility. Recently, the multi-layered neural network is used as a powerful data mining tool. Nevertheless, in many cases with complex boundary of classification, the successful learning is not guaranteed and the problems of long learning time and local minimum attraction restrict the field application. In this paper, an Improved learning procedure of multi-layered neural network is proposed. The procedure is based on the generalized delta rule but it is particular in the point that the architecture of network is not fixed but enlarged during learning. That is, the number of hidden nodes or hidden layers are increased to help finding the classification boundary and such procedure is controlled by entropy evaluation. The learning speed and the pattern classification performance are analyzed and compared with the back-propagation algorithm.

Keyword : 신경망, 역전파 신경망, 데이터 마이닝

논문접수일 : 2001년 4월 18일 논문게재확정일 : 2001년 5월 17일

* LG-EDS e-Solution사업부 CRM팀

** 동국대학교 산업공학과 교수

1. 서론

최근 데이터마이닝(data mining), 경영예측, CRM(customer relationship management) 등 지식산업 분야에서 학습 및 추론, 데이터 분류, 및 군집화 등 지능정보시스템의 방법론 개발에 관한 연구가 활발히 행해지고 있다. 이러한 지능정보시스템의 구축방법론으로는 전통적인 통계적 기법들, 연역적 학습으로 의사결정 나무(decision tree)를 구성하여 분류하는 방법, 그리고 신경망을 사용하는 방법이 많이 사용되고 있다. 가장 많이 사용되는 신경망 모델로는 오류 역전파(back-propagation) 알고리즘을 사용하는 다층 신경망을 들 수 있다. 오류 역전파 알고리즘은 에너지라고 불리는 에러함수의 최소치를 구하기 위해 신경망의 층(Layer)을 따라 노드의 연결강도의 변경치를 구하기 위해 이루어지는 반복적인 미분과정을 체계적이고 효율적으로 정립한 것이라고 할 수 있다.

그러나 오류 역전파 알고리즘은 학습과정에서 유의할 만한 비용함수의 감소없이 오랜 시간을 보내게 된다던가 지역 최소점(local minimum)에 빠져 학습에 실패하기도 하는데, 학습대상 데이터의 분류경계가 복잡해짐에 따라 그렇게 될 위험성은 더 커진다. 또한 학습 이전에 신경망의 구조를 결정하는 데에 일반적인 규칙이 정해져 있지 않아 실제 적용하는 데에 많은 어려움이 있다. 본 연구의 목적은 학습대상 데이터의 분류경계를 효율적으로 발견하기 위해 학습을 진행하는 과정에서 신경망의 구조를 스스로 설계해 나갈 수 있는 알고리즘을 개발하는 것이다. 제시된 방법론에서는 최소규모의 신경망을 가지고 학습을 시작하여 은닉노드 또는 은닉층을 증가하여 가는 방식을 취한다. 특히 은닉노드를 추가할 것인가 또는 은닉층을 추가할 것인가 하는 사항과 추가되는 은닉노드의 주요 학습대상 데이터를 결정하는 사항에 있어 엔트로피 척도를 기준으로 이용한다. 추가된 은닉노드에 해서는 주요 학습대상데이터만을 대상으로 학습하는 과정을 거치는데 이는 입력공간에서 에러

를 유발시키는 데이터가 포함된 부분의 분류경계를 효과적으로 발견하기 위한 것이다. 이러한 학습과정 및 네트워크 확정과정은 모든 데이터의 완전학습 또는 목적수준의 학습이 이루어질 때까지 반복되어진다

2. 패턴분류와 신경망구조 설계방법의 고찰

2.1 패턴분류

일반적으로 패턴분류과정은 특징추출단계와 학습단계로 나눌 수 있으며, 본 논문은 이중에서도 학습단계에 초점을 맞춘 것이다. 특징 추출에는 학습할 패턴을 학습에 용이한 형태로 변환하는 것을 말하며, 이에선 전처리(preprocessing), 분할(segmentation), 정규화(normalization) 등의 방법이 사용되고 있다[2, 3]. 여기서 전처리는 잡음(noise pattern)이나, 관측시의 왜곡과 같은 불량데이터를 배제하여 시스템에서 필요로 하는 자료만을 추출하는 과정을 포함한다.

학습단계는 특징추출 단계를 거친 데이터들을 분류할 수 있는 함수를 발견하는 일로 볼 수 있다. 이러한 분류함수를 결정하는 것은 통계학분야의 판별분석(discriminant analysis)과 Logistic regression 등의 방법, 기계학습(machine learning)분야의 CHAID, C4.5, CART 등의 방법, 그리고 다층신경망을 포함한 다양한 신경망 응용방법들이 사용된다. 이 중에서도 신경망을 사용한 방법은 예측된 결과의 설명력이 부족하다는 단점이 있으나 성능이 우수하여 다양한 응용연구가 수행되고 있다[4, 5, 6].

2.2 신경망 구조 설계에 대한 선행 연구

패턴 분류를 위한 최적의 신경망 구조를 설계하기 위한 연구는 최근 활발히 진행되고 있다. 이러한 연구는 크게 세 가지 부류로 나누어 볼 수 있

는데 큰 구조의 신경망으로부터 시작하여 작은 구조의 신경망으로 축소해 나가는 방법과 하나의 노드로부터 시작하여 문제를 해결할 때까지 노드를 증가시켜 큰 구조의 신경망으로 확대해 나가는 방법, 그리고 최근 연구에 많이 사용되고 있는 유전자 알고리즘을 이용하여 신경망의 구조를 설계하는 방법이다[9, 13]. 이들 각각에 대해 살펴보면 다음과 같다.

2.2.1 전지법(Network Pruning Method)

Sietsma and Dow[19]가 제안하였으며, 학습이 끝난 이후 중요하지 않은 연결강도를 갖는 노드들을 제거해 버린 이후에 다시 학습을 수행하여 나가는 방법이다. 중요성이 낮은 연결강도를 갖는 노드들을 제거한 이후에 추가학습을 수행하는데 추가학습은 보다 효율적으로 이루어지게 된다.

2.2.2 연결강도 소멸법(Weight Decay Method)

Hinton[12], Scalettar and Zee[18]가 제안한 방법으로 학습을 수행하면서 연결강도를 일정비율 감소시켜 나가는 방법이다. 이는 연결강도의 합을 목적식(cost function)에 포함함으로써 이루어지는데, 중요한 역할을 하는 연결강도의 학습을 방해할 수 있으며, 연결강도들의 값이 지나치게 작아질 수 있다는 단점이 있다.

2.2.3 출력노드 출발법(Upstart Algorithm)

Frean[8]이 제시한 알고리즘으로 출력 노드로부터 시작하여 잘못된 패턴들을 분류하여 나누어 가는 방식이다. 가장 먼저 생긴 노드(출력노드)를 가능한 한 입력데이터를 정확히 분류할 수 있도록 학습시킨다. 그 뒤에 출력이 틀린 모든 학습데이터를 두개의 새로운 노드를 만들어서 학습시킨다. 새로 추가된 노드는 적어도 하나 이상의 자료를 분류할 것이고 이렇게 함으로서 모든 패턴을 구분할 수 있게된다. 출력노드 출발법은 완전한 다층신경망 구조를 갖지 않고 나무(tree)형태의 신경망 구

조로 만들어진다.

2.2.4 연관분류법(Cascade Correlation Algorithm)

Fahlman and Lebiere[7]가 제시한 알고리즘으로 Frean[8]이 제시한 출력노드 출발법과 유사하지만 학습된 연결강도를 고정시킨 후에 계속 위층으로 노드를 늘려 나가는 것이 다른 점이다.

2.2.5 순서적 은닉노드 학습법(Sequential Learning Method)

Marchand et al.[14]이 제시한 방법으로 하나의 은닉층을 가지는 신경망의 구축방법이다. 은닉 노드는 하나 또는 그이상의 패턴이 분류 될 때까지 하나씩 증가되며 분류된 패턴은 다음의 은닉 노드의 학습에서 제외된다. 즉, 기존 은닉노드의 연결강도는 고정시키고 학습이 되지 못한 데이터만을 새로 만들어진 은닉 노드의 연결강도를 이용하여 학습시키는 것이다.

2.2.6 분할법(Tiling Algorithm)

Mézard and Nadal[15]가 제시한 방법으로 은닉층을 늘려가면서 다층신경망의 구조를 만들어간다. 학습방법으로 포켓(pocket) 알고리즘을 사용한다. 각각의 증속된 층은 이전의 층에 비해 더 적은 수의 노드를 갖게 되며 최종적으로 하나의 출력노드를 가질 때 학습은 종료된다.

2.2.7 유전자 알고리즘을 사용한 방법

Harp et. al.[11], Miller et. al.[16]등이 제시하였으며, 최근 많이 응용되고 있는 유전자 알고리즘을 이용하여 신경망 구조를 설계하는 방법이다. 신경망구조를 유전자정보로 변환하고 유전자정보에 대응되는 신경망구조의 성능을 평가한 후에 우수한 결과를 보이는 유전자정보들을 이용해 다시 새로운 유전자 집합을 만드는 방법이다. 이 방법은 큰 규모의 신경망 구조설계가 어렵고, 많은 계산을 필요로 한다는 단점이 있다.

3. 제시된 다층신경망의 학습과 구조설계

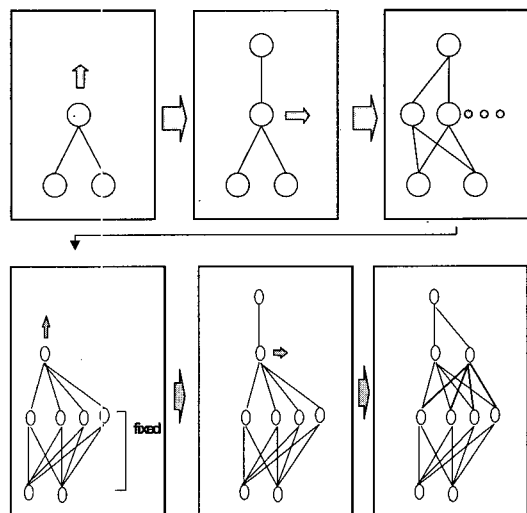
3.1 역전파 알고리즘의 개선방안

일반적으로 다층신경망의 학습알고리즘은 입력-출력 쌍으로 이루어진 학습집합(training set)으로부터 주어진 데이터들을 분류할 수 있도록 전방향출력(feed-forward)네트워크의 연결강도를 결정해 주기 위한 방법을 제공하여야 한다.

역전파 알고리즘에서는 학습대상 데이터 전체에 대한 발생에러를 종합적으로 최소화시키는 방향으로 연결강도를 조정하기 때문에 학습진행에 지장을 주는 입력공간부분의 분류경계를 효율적으로 찾기 어려운 경우가 발생한다. 즉, 분류경계를 신속히 찾기 위해서는 전체 데이터의 에러를 모두 고려하기보다는 문제가 되고 있는 부분입력공간의 오류발생 데이터에 초점을 맞추는 것이 효과적이며 이를 위해 일시적으로 전체 에러의 크기가 다소 증가되는 방향으로 연결강도를 변경하는 것이 바람직하나 전통적인 역전파 알고리즘에서는 이를 수용하기 어렵다. 뿐만 아니라 학습이 진행되고 있는 신경망의 구조가 대상 데이터의 분류에 적합한 것이 아닐 경우 전체 데이터의 발생에러를 최소화시키는 데에는 상당한 시간이 걸리거나 학습에 실패하게 되는 문제점이 있다. 일반적으로 역전파 알고리즘에서는 이를 해결하기 위해서 테스트데이터가 보여주는 학습성공율을 바탕으로 신경망의 구조를 변경해 보는 간접적인 방법을 사용한다. 본 연구에서는 이러한 문제를 해결하기 위해 학습신경망의 구조를 최소의 규모에서 시작하여 학습도중 확장해 가는 개념을 사용한다. 신경망의 구조를 확장하는 것은 기본적으로 은닉노드의 추가를 의미하는데 본 연구에서는 은닉노드를 단순히 추가하는 것이 아니라 은닉노드를 추가함으로써 에러를 발생시키는 부분입력공간의 데이터 분류경계를 신속히 찾아갈 수 있는 방법을 제시하고자 한다. 은닉노드의 추가시점은 학습의 속도가 현저히 느

러지는 시점으로, 이 때는 노드들의 연결강도의 값의 변화가 매우 작거나 진동현상을 보이게 된다.

신경망의 구조확장은 위에서 설명한 바와 같은 은닉노드의 추가 이외에 새로운 은닉층을 추가하는 것을 생각할 수 있다. 학습과정에서 은닉노드들을 계속 추가해나가는 것보다 은닉층을 한 층 더 증가시키는 것이 보다 효과적이 될 수 있는데 이는 기존의 은닉노드들이 대상 데이터의 분류경계상 주요 부분을 찾고 있다면 분류경계의 미세 부분을 찾기 위해 은닉노드를 계속 증가시키는 것보다 기존의 은닉노드들에 의해 변환된 데이터 공간에서 학습을 재개하는 것이 효과적이 될 수 있기 때문이다. 따라서 본 연구에서는 신경망 구조의 확장을 위해 은닉노드의 추가를 기본적으로 고려하지만 은닉노드의 수가 많고 새로운 은닉노드 추가에 의한 어려감소의 크기가 미미할 것으로 판단될 경우 은닉층의 증가를 선택하는 방안을 택한다. 이러한 과정은 학습의 속도를 개선할 뿐 아니라 학습대상 데이터의 분류를 위한 최적 신경망구조를 발견한다는 것을 뜻한다. <그림 1>은 본 연구에서 제시하고 있는 신경망의 확장에 의한 구조설계과정의 개념을 나타낸 것이다. <그림 1>에서는 출력층에 출력 노드가 하나인 경우를 예시하고 있는데



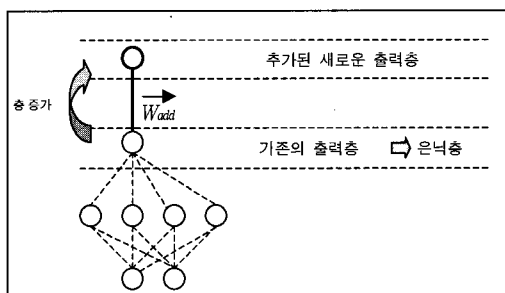
<그림 1> 신경망의 구조설계 과정

그림에서 볼 있듯이 은닉노드의 증가는 출력층 바로 아래에 해당하는 최상의 은닉층에서 일어나며 은닉층을 추가하게 되면 새로운 은닉층이 최상은닉층이 되고 이전의 출력노드는 새로운 최상은닉층의 첫 번째 은닉노드가 되며 새로운 출력노드가 만들어진다. 그리고 은닉층 증가 이후의 은닉노드의 추가는 새로운 최상 은닉층에서 일어나게 된다.

3.2 신경망의 확장에 따른 학습과정

앞 절에서 설명한 바와 같이 신경망의 확장은 은닉노드의 추가 또는 은닉층의 추가에 의해 이루어진다. 단, 신경망의 초기구조가 입력층과 출력층만으로 이루어져 있는 경우, 신경망의 확장은 은닉층의 증가로 시작된다. 즉, 은닉층이 없는 상황에서는 은닉노드를 추가할 수 없으므로 신경망의 구조확장은 은닉층의 확장으로 시작될 수 밖에 없다.

신경망의 확장과정에서 만약 새로운 최상은닉층이 생기게 되면 앞 절에서 설명한 바와 같이 기존의 출력노드는 새로운 최상은닉층의 첫 은닉노드가 되므로 새로운 출력노드를 만들어야 한다. 아래 <그림 2> 새로운 층이 추가되는 상황을 나타낸 것이다.

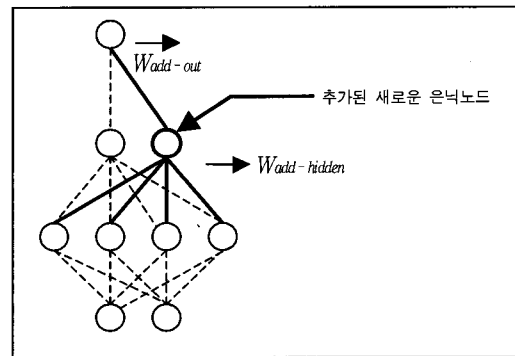


<그림 2> 새로운 층이 추가된 신경망 구조

<그림 2>와 같이 새로운 출력노드가 만들어진다면 새로운 출력노드와 기존의 출력노드였던 최상 은닉층의 은닉노드사이의 연결강도를 구해야 한다. 이를 위해 이 연결강도를 0이 아닌 값으로 초

기화한 다음 학습과정을 거치는 데 이 때에는 신경망의 다른 연결강도들은 모두 고정시키고 이 연결강도만을 변경하면서 학습을 진행한다. 이 학습과정이 수렴하면 일반적인 학습과정으로 돌아간다. 즉, 어떤 연결강도도 고정시키지 않은 상태에서 학습을 하는 것이다.

은닉층을 추가하지 않고 은닉노드만을 최상 은닉층에 추가하는 상황은 아래 <그림 3>에 나타나 있다. 이 경우에는 추가된 은닉노드와 출력노드사이, 그리고 추가된 은닉노드와 최상은닉층의 아래층의 노드들사이의 연결강도를 초기화해 주어야 한다. 이에 관해서는 다음 절에서 자세히 설명하도록 한다.



<그림 3> 새로운 은닉노드가 추가된 신경망 구조

3.3 엔트로피를 이용한 신경망의 확장과 추가된 은닉노드의 학습

엔트로피는 학습대상 데이터의 무질서 정도, 즉 정보량을 의미하는 것으로 본 연구에서는 데이터의 분류 효과를 가늠하는 척도로 사용한다. 엔트로피의 평가결과는 두 가지 목적에 이용된다. 먼저, 신경망의 확장 시 은닉노드를 증가할 것인가 또는 은닉층을 증가할 것인가를 결정하기 위한 정보로 이용한다. 은닉노드의 추가는 최상 은닉층에서 일어나는데 만약 은닉노드의 추가 이후 학습이 진행되어 제차 신경망을 확장하고자 할 경우 최근의 은닉노드의 증가가 엔트로피의 감소에 얼마나 기

여하였는가를 평가할 수 있다. 만약 그 기여정도가 미미하고 이미 다수의 은닉노드가 최상 은닉층에 있다면 추가적으로 은닉노드를 증가하는 것보다 새로운 최상 은닉층을 추가하는 것이 바람직한 것으로 판단한다.

엔트로피 평가치를 이용하는 두 번째 목적은 은닉노드를 추가할 때 추가되는 은닉노드의 주요 학습대상 데이터를 결정하기 위한 것이다. 은닉노드의 추가는 앞에서 설명한 바와 같이 입력공간에서 에러를 발생시키는 데이터의 분류공간을 신속히 찾기 위해 행해지는데, 추가되는 은닉노드는 이러한 목적을 효과적으로 달성해야 한다. 이를 설명하기 위해 출력목표치가 0 또는 1인 경우에 대한 엔트로피를 고려하기로 한다. 이 경우 학습데이터들은 신경망의 출력값의 범위와 목표유형에 따라 <표 1>과 같이 네 가지의 부분으로 나뉘어 진다. <표 1>의 신경망 출력은 은닉노드가 추가되기 전의 신경망에 의해 구해진 것이다. 여기서 신경망의 출력치는 0과 1 사이의 연속적인 값이지만 출력값이 0.5미만인 경우는 목표유형 $T=0$, 그리고 0.5이상인 경우는 목표유형 $T=1$ 에 해당하는 데이터를 정확히 분류한 것으로 간주한다.

<표 1> 목표출력치와 신경망 출력치에 따른 패턴의 개수와 엔트로피

목표유형 신경망출력	$T=1$	$T=0$	계	$entropy$
$0 \leq O_j < 0.5$	N_1^-	N_0^-	N^-	$entropy_{c-}$
$0.5 < O_j \leq 1$	N_1^+	N_0^+	N^+	$entropy_{c+}$
계			N	$entropy_{sys}$

<표 1>에서 N_1^- , N_0^- , N_1^+ , N_0^+ 는 각 부분에 속하는 학습데이터의 개수이며 N_1^- 와 N_0^+ 는 오분류가 일어난 데이터의 개수이다. $entropy_{c-}$ 는 신경망의 출력치가 0.5미만인 데이터들의 엔트로피이며 $entropy_{c+}$ 는 0.5이상인 경우의 데이터들의 엔트로피로서 다음과 같이 정의된다.

$$\begin{cases} entropy_{c+} = - \sum_i N_i^+ / N^+ \log_2 N_i^+ / N^+ \\ entropy_{c-} = - \sum_i N_i^- / N^- \log_2 N_i^- / N^- \end{cases} \quad (1)$$

시스템 엔트로피는 학습데이터들 중 오분류된 데이터의 수가 증가함에 따라 증가하므로 이를 기준으로 학습이 올바르게 수행되고 있는가를 평가할 수 있다. 시스템 엔트로피의 정의는 식 (2)와 같다.

$$entropy_{sys} = - \sum_i N_i^+ / N \cdot \log_2 N_i^+ / N - \sum_i N_i^- / N \cdot \log_2 N_i^- / N \quad (2)$$

식 (1)의 $entropy_{c-}$ 와 $entropy_{c+}$ 는 오분류된 데이터의 개수 N_1^- 와 N_0^+ 가 클수록 큰 값을 갖는데 두 엔트로피 $entropy_{c-}$ 와 $entropy_{c+}$ 를 비교하여 전자가 크다면 N_1^- 을, 후자가 크다면 N_0^+ 을 감소시키는 것을 은닉노드의 주요 목표로 선택한다. 만약 두 개의 엔트로피가 같다면 랜덤하게 선택한다. 추가되는 은닉노드의 주요 목표가 N_1^- 을 감소시키는 것이라면 N^- 에 해당하는 데이터를 주요 학습대상 데이터로 삼으며 N_0^+ 을 감소시키는 것이라면 N^+ 에 해당하는 데이터를 주요 학습대상 데이터로 삼는다. 이는 N_1^- 을 감소시키기 위해서는 기존의 신경망이 같은 영역으로 분류하고 있는 N_1^- 와 N_0^- 에 해당하는 데이터를 정확히 분류해야 하며 N_0^+ 를 감소시키기 위해서는 기존의 신경망이 같은 영역으로 분류하고 있는 N_1^+ 와 N_0^+ 에 해당하는 데이터를 정확히 분류해야 한다는 직관이 따른 것이다.

주요학습대상 데이터가 선택되면 이들만을 대상으로 학습하되 신경망의 모든 연결강도를 변경해 가는 것이 아니라 추가된 은닉노드의 연결강도만을 변경해나간다. 즉, <그림 3>의 \vec{W}_{add_hidden} 이외의 연결강도를 모두 고정시킨 상태에서 이들만을 변경한다. 이러한 주요학습대상 데이터의 학습 과정은 연결강도의 변화량이 거의 없어지는 수렴

상태에 다다를 때까지 계속한다.

연결강도가 수렴상태가 도달하게 되면 일반적인 학습과정으로 다시 돌아간다. 즉, 고정되는 연결강도가 없는 조건에서 주요학습대상 데이터가 아닌 전체 학습데이터를 대상으로 학습한다. 이와 같이 확장된 신경망으로 학습을 진행하다가 다시 학습 과정이 현저히 느려지면 지금까지 설명한 바와 같이 또다시 은닉노드 또는 은닉층의 추가를 결정하게 되는 것이다. 아래에는 이러한 학습과정을 단계 별로 요약하였다.

단계 0. 초기 신경망 구조를 결정한다. 이는 입력 층과 출력층만을 가지는 구조일 수도 있고, 은닉층을 하나 가지되 그 은닉층에 포함된 노드가 하나인 구조일 수도 있다.

단계 1. 학습데이터에 대한 시스템 엔트로피를 식 (6)에 의해 구하고 학습을 시작한다.

단계 2. 학습이 수렴상태가 될 때까지 진행한다. 만약 이 때의 비용함수값이 만족수준 이하가 되었다면 학습을 종료한다. 그렇지 않다면 단계 3으로 간다.

단계 3. 학습 데이터에 대한 시스템 엔트로피를 다시 구한 다음 현재의 신경망 학습직전에 구해 두었던 시스템 엔트로피와 비교한다. 만약 이 값이 현저히 감소되었거나 최상은닉층의 은닉노드의 수가 최소수준 이하로 적다면 은닉노드를 추가하는 단계 5로 간다. 그렇지 않다면 은닉층을 추가하는 단계 4로 간다.

단계 4. 새로운 최상은닉층을 만들고 기존의 출력노드를 추가된 은닉층의 첫 은닉노드가 되게 한다. 또한 새로운 출력노드를 만들고 새로운 출력노드와 기존의 출력노드였던 최상은닉층의 은닉노드 사이의 연결강도를 초기화한 다음 이 연결강도값을 조정한다. 즉, 이 연결강도 이외의 연결강도들을 고정시킨 상태에서 학습을 시작하여 수렴상태가 될 때까지 진행한다. 수렴상

태가 되면 단계 1로 돌아 간다.

단계 5. 최상은닉층에 하나의 은닉노드를 추가하고 추가된 은닉노드와 출력노드사이, 그리고 최상은닉층의 아래층 노드들과 추가된 은닉노드사이의 연결강도들을 초기화한 다음 단계 6으로 간다.

단계 6. 주요학습대상 데이터를 선택하고 이들을 대상으로 학습을 하되 추가된 은닉노드의 연결강도들 이외의 연결강도들은 모두 고정시킨 상태에서 학습을 진행한다. 학습이 수렴상태에 다다르면 단계 1로 돌아간다.

위에서 설명한 단계 3에서 은닉층의 은닉노드 수의 최소수준은 입력노드의 수로 정하였다. 이는 은닉층의 노드수가 입력노드의 수보다 적은 경우 입력공간의 차원이 줄어들어 분류정보의 손실위험성이 있을 것으로 생각되었기 때문이다. 가상 데이터를 이용한 실험에서 은닉노드의 수는 입력노드의 수의 2배 이상이 되는 경우가 많았는데 이러한 경우 입력공간의 차원이 증가되어 데이터분류가 잘 일어나는 것으로 판단이 되었다.

3.4 연결강도 학습과정의 수식화

<그림 4>는 출력노드가 하나인 다층 신경망의 일반적 구조로서 L 층은 신경망의 최상층으로서 출력층을 나타낸다. <그림 4>에서 bias는 항상 -1의 값을 출력하는 고정노드이다. 본 연구에서는 편의상 출력노드가 하나인 경우에 대해서만 수식을 전개하고자 한다.

본 연구에서 사용되는 기호는 다음과 같다.

- T^μ = 학습데이터 μ 의 목표출력값
- I_k^μ = 학습데이터 μ 의 k 번째 값.
- O_{ii}^μ = l 층, i 번째 은닉노드의 출력치.
- O_L^μ = L 층의 출력노드의 출력치.
- H_{ii}^μ = l 층, i 노드의 입력치.

$$H_{ii}^{\mu} = \sum_{j=0}^{n_{(l-1)}} W_{ij} \cdot O_{(l-1)j}^{\mu}$$

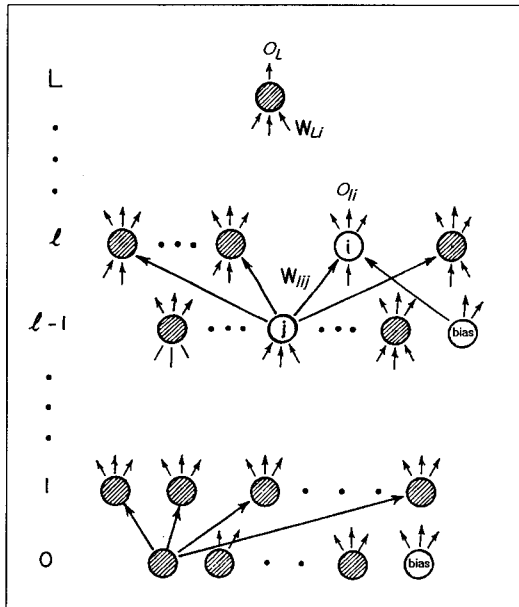
H_L^{μ} = L층의 출력노드의 입력치.

$$H_L^{\mu} = \sum_{j=1}^{n_L} W_{Lj} \times O_{(L-1)j}^{\mu} - \theta_L$$

W_{ij} = l층의 노드 i와 (l-1)층의 노드 j와의 연결강도.

W_{Lj} = L층의 출력노드와 (L-1)층의 노드 j와의 연결강도.

단, W_{i0} 와 W_{L0} 는 입력이 항상 -1인 bias의 연결강도



〈그림 4〉 출력노드가 하나인 다층 신경망

각 노드에서의 출력함수 $g(\cdot)$ 는 비선형 함수 (sigmoid function)이며 식 (3)과 같이 표현된다.

$$g(H_{ii}) = 1 / \{1 + \exp(-H_{ii}/T_p)\} \quad (3)$$

식 (1)에서 T_p 값은 sigmoid 함수인 $g(\cdot)$ 의 기울기를 결정하는 상수이며, 은닉노드의 학습이 진행되면서 이 값을 감소시켜 나간다. T_p 값을 낮추어 감에 따라 출력함수의 기울기가 가파르게 되

어 결과적으로 계단함수(threshold function)의 형태를 취하게 되는데 이는 은닉노드의 학습을 수렴하게 하는 역할을 한다.

O_{ii}^{μ} 은 식 (1)의 출력함수 $g(\cdot)$ 로부터 식 (4)과 같이 결정된다.

$$O_{ii}^{\mu} = g\left(\sum_{j=0}^{n_{(l-1)}} W_{ij} \cdot O_{(l-1)j}^{\mu}\right) \quad (4)$$

식 (2)로부터 출력된 값들은 다음 (l+1)층으로 전파되며 최종적으로 출력층 L에서는 최종 출력 O_L^{μ} 가 식 (5)와 같이 산출된다.

$$O_L^{\mu} = g(H_L^{\mu}) = g\left(\sum_{j=0}^{n_L} W_{Lj} \times O_{(L-1)j}^{\mu}\right) \quad (5)$$

하나의 패턴 μ 에서 발생하는 오차(Error)는 목표 출력치와 신경망의 실제 출력치와의 차이이다. N개의 학습패턴에 대한 비용함수(cost function)는 평균자승오차(mean square error; MSE)를 사용한다. 즉,

$$\begin{aligned} E(W) &= \frac{1}{2N} \sum_{\mu=1}^N (T^{\mu} - O_L^{\mu})^2 \\ &= \sum_{\mu=1}^N (T^{\mu} - g\left(\sum_{j=0}^{n_{(L-1)}} W_{Lj} \times O_{(L-1)j}^{\mu}\right))^2 \end{aligned} \quad (6)$$

위의 비용함수를 최소화하기 위해 최급강하규칙(最急降下法; gradient decent rule)을 사용하여 비용함수가 가장 급격히 감소하는 방향으로 연결강도의 수정량이 결정된다. 즉, 출력노드의 경우,

$$\Delta W_{Lj} = \frac{\eta}{N} \sum_{\mu} [T^{\mu} - O_L^{\mu}] \cdot g'(H_L^{\mu}) \cdot O_{(L-1)j}^{\mu} \quad (7)$$

$$W_{Lj}(t+1) = W_{Lj}(t) + \Delta W_{Lj} + \alpha [W_{Lj}(t) - W_{Lj}(t-1)] \quad (8)$$

그리고 은닉노드의 경우,

$$\Delta W_{ij} = \frac{\eta}{N} \sum_{\mu=1}^N (T_{ii}^{\mu} - O_{ii}^{\mu}) \cdot g'(H_{ii}^{\mu}) \cdot O_{(l-1)j}^{\mu} \quad (9)$$

$$W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij} + \alpha [W_{ij}(t) - W_{ij}(t-1)] \quad (10)$$

가 된다.

위의 식에서 N은 학습대상 데이터의 개수인데 일반 학습과정에서는 전체 학습 데이터의 개수이며 은닉노드의 추가 시 이루어지는 주요학습대상 데이터에 대한 학습과정에서는 주요학습대상 데이터의 개수를 의미한다. 또한 위의 식에서 η 는 학습계수이며, α 는 모멘텀계수로서 신경망 학습과정의 안정성과 신속성에 기여한다. 위의 식에서 보여주듯이 연결강도의 수정은 학습대상 패턴이 모두 제시된 이후에 각각의 패턴이 발생시킨 오차를 합하여 연결강도를 수정하는 일괄 수정법(batch update method)을 사용하고 있는데 이는 하나의 패턴이 제시될 때마다 연결강도를 수정하는 패턴 수정법(pattern update method)에 비해 학습시에 발생하는 진동(oscillation)을 줄이고 학습시간을 다소 단축시키는 이점이 있다.

은닉노드의 학습시의 경우에는 인위적으로 목표 출력치를 $T_{ii}^{\#}$ 를 출력노드의 학습에 도움이 되는 방향으로 조정해 준다.

식 (9)와 (10)에서 $T_{ii}^{\#}$ 는 식 (12)과 같다.

$$T_{ii}^{\#} = \begin{cases} T_{ii}^{\#} & , W_{Lj} \geq 0 \\ NOT(T_{ii}^{\#}) & , W_{Lj} < 0 \end{cases} \quad (11)$$

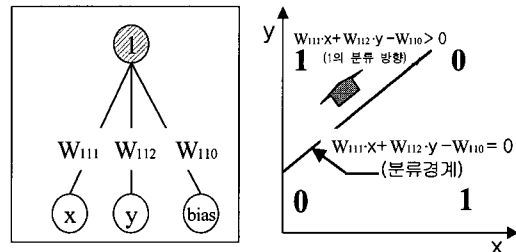
여기서 $T_{ii}^{\#}$ 는 은닉노드의 학습방향을 설정해 주는 것이다. 즉 출력노드와 학습되는 은닉노드의 연결가중치가 음으로 초기화되었다면, 은닉노드의 학습은 원래의 목표 패턴과 반대 방향으로 설정하는 것이다. 은닉노드의 학습 방향을 식 (11)와 같이 설정하여 줄이므로서 출력노드의 학습은 더욱 빠르게 수행 될 수 있다. 이는 학습 시에 연결강도의 부호가 바뀌는 데에 많은 시간이 걸리기 때문이며, 이와 같이 은닉노드의 학습을 수행함으로써 출력노드의 학습 시에 W_{Lj} 연결강도의 부호가 바뀌지 않으며, 학습속도도 높일 수가 있다.

3.7 예제(XOR 문제)

본 절에서는 예제로서 많이 사용되고 있는 XOR 문제를 가지고 제시된 학습과정을 설명한다. XOR 문제는 두 개의 입력변수 x, y 와 하나의 출력치 T 를 갖는 문제로 다음과 같은 4개의 학습데이터 $P = \{x, y, T\}$ 로 이루어져 있다.

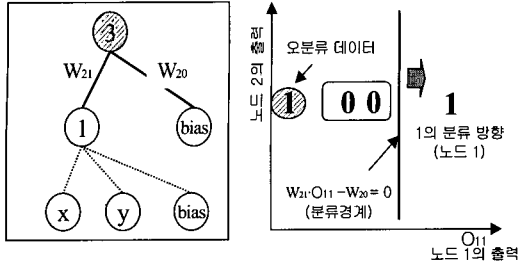
$$\begin{aligned} P^1 &= \{0, 0, 0\} \\ P^2 &= \{0, 1, 1\} \\ P^3 &= \{1, 0, 1\} \\ P^4 &= \{1, 1, 0\} \end{aligned}$$

위의 네 가지 학습데이터의 학습을 위해 입력층과 출력노드만으로 이루어진 신경망 구조선택하였다. <그림 5>는 이 신경망의 구조와 학습이 이루어진 이후 입력공간에서의 입력데이터분포 및 출력노드에 의해 생성된 데이터 분류경계선을 나타내고 있다. 그림에서 분류경계선위의 화살표는 '1' 유형으로 분류한 영역의 방향을 나타낸다.



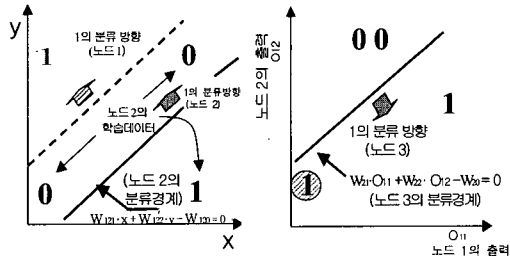
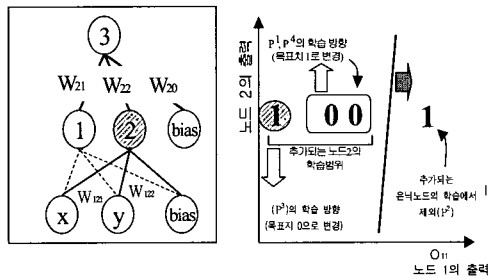
<그림 5> 첫 번째 노드의 학습

<그림 5>의 우측그림서는 데이터의 선형분리가 불가능하여 P^3 가 오분류된 것을 나타낸다. 이를 해결하기 위해 신경망을 확장하여야 하는데 신경망의 초기구조에 은닉층이 없으므로 은닉층을 추가하고 새로운 출력노드를 만들어야 한다. <그림 6>은 확장된 신경망 구조와 이를 이용한 학습결과를 나타내고 있다. <그림 6>의 오른쪽 그림은 학습데이터의 입력값들에 대한 은닉층의 은닉노드 출력값의 분포와 출력노드에 의한 분류경계선을 나타내고 있다.



<그림 6> 출력 노드의 학습

<그림 6>에서 보듯이 P^3 가 여전히 오분류되어 있으며 이를 해결하기 위해 신경망을 확장하여야 하는데 현재 은닉노드의 수가 하나 뿐이므로 은닉층의 증가는 고려하지 않고 은닉노드를 추가한다.



<그림 7> 은닉노드가 추가된 신경망의 구조와 신경망의 학습종료

<그림 7>은 은닉노드가 추가된 신경망의 구조와 이를 이용한 학습결과를 보여주고 있다. <그림 7>의 두 번째 그림은 출력층에서 추가되는 은닉노드의 학습대상과 학습방향을 설정하는 것을 나타내고 있다. 은닉노드가 추가되기 전의 엔트로피가 $entropy_{c-} \rightarrow entropy_{c+}$ 이므로 추가되는 은닉노드의 주요학습대상 데이터는 P^2 가 제외된 P^1, P^3, P^4 이며, 추가된 은닉노드의 연결가중치는 이 데이터들만의 학습과정을 거쳐 수렴된 후 일반

적인 다층 신경망의 학습을 수행하였다. 만약 $W_{22} < 0$ 값으로 초기화되었다면, 앞절의 수식 (11)에 따라, 추가된 노드3에서 P^3 의 학습방향은 1에서 0으로 변경되어 학습하게 된다. <그림 7>의 세 번째 그림은 P^1, P^3, P^4 만을 대상으로 한 학습결과 추가된 은닉노드가 입력공간에서 생성한 분류경계선을 보여주고 있다. 또한 <그림 7>의 네 번째 그림은 최종적으로 학습을 완료한 후 출력노드가 두 은닉노드의 출력범위 상에서 생성한 분류경계선을 보여주고 있다. 모든 데이터가 성공적으로 분류된 것을 알 수 있다. 결과적으로 XOR문제를 해결한 신경망 구조는 하나의 은닉층을 가지며 그 은닉층은 두 개의 은닉노드로 구성된 것을 알 수 있다.

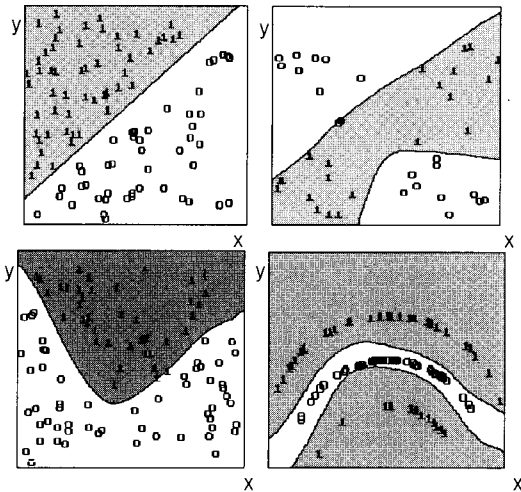
4. 모의실험 결과

본 연구에서는 제시된 알고리즘의 학습능력 평가를 위해 모의실험을 수행하였다. 모의실험을 위해 여섯 가지의 형태의 학습예제집합(training example set)을 인위적으로 발생시켜 제시된 알고리즘으로 학습시켰다. 학습예제집합의 데이터는 두 개의 입력치 (x, y)와 하나의 출력치를 갖고 있으며 출력치는 0 또는 1의 값을 갖도록 하였다. 또한 학습데이터의 입력값은 [0, 1]범위의 값으로 정규화(normalize)하여 신경망에 입력되었으며, 학습예제들중 마지막 5, 6번 예제는 제시된 알고리즘과 역전파 알고리즘에 대해 입력공간에서 생성된 분류경계와 평균자승오차(MSE)를 비교하였다.

4.1 제시된 알고리즘의 학습수행 결과

제시된 학습알고리즘의 수행도 평가를 위해 네 가지의 학습 집합을 각각 선형(학습예제 1), XOR 구조(학습예제 2), 포물선(학습예제 3), 원호(학습예제 4)의 형태로 다양하게 발생시켰다. 각 예제 데이터의 입력공간상의 분포와 신경망의 학습결과 는 <그림 8>에 나타나 있다. <그림 8>은 각각의

학습예제에 대해 제시된 학습 알고리즘으로 신경망을 학습시킨 이후 생성된 입력공간에서의 분류 경계를 구하여 나타낸 것이다.



<그림 8> 예제와 생성된 분류경계(좌측상단 그림부터 예제 1, 예제 2, 예제 3, 예제 4)

<그림 8>에서 '1'로 표시된 데이터는 목표출력치가 1이며 '0'로 표시된 데이터는 목표출력치가 0임을 뜻한다.

제시된 알고리즘으로 다층신경망을 학습시킨 결과 주어진 패턴에 대해 학습이 적절히 수행됨을 알 수 있었다. 학습 종료 이후의 결과는 <표 2>와 같다.

<표 2> 예제 1~예제 4의 학습 결과

결과 학습 집합	데이터 수	시간 (epoch)	오차 (MSE)	엔트 로피	생성 층수	생성 노드 수
예제 1	100	952	0.0067	0	1	1
예제 2	46	3389	0.0190	0	2	5
예제 3	100	5503	0.0023	0	3	9
예제 4	100	8750	0.0023	0	4	13

<표 3>을 보면 학습종료 이후의 엔트로피는 모두 "0"으로 주어진 학습데이터를 모두 오분류 없이 분류하였다. 신경망의 출력치와 목표출력치와

의 평균자승오차(MSE)오차는 네 예제 모두 매우 작은 수준으로 얻어져 학습 데이터들이 잘 분류됨을 알 수 있다. 학습이 종료된 이후의 신경망의 구조는 아래의 <표 3>과 같다. <표 3>을 보면 학습 집합의 분류경계가 복잡해질수록 신경망이 규모가 커짐을 알 수 있다.

<표 3> 예제의 생성된 신경망 구조

학습 집합	구조 층 수	은닉 및 출력층의 노드 수			
		1층	2층	3층	4층
예제 1	1	1	.	.	.
예제 2	2	4	1	.	.
예제 3	3	6	2	1	.
예제 4	4	6	4	2	1

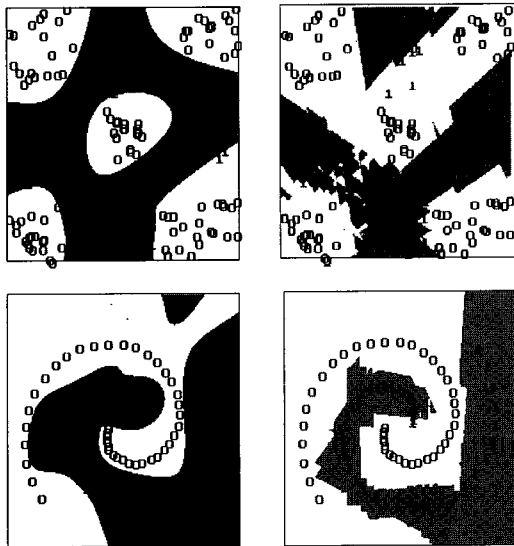
4.2 역전파 알고리즘과의 비교 및 고찰

예제 5번과 6번은 현재 다층신경망의 학습에서 가장 일반적으로 사용되는 역전파 알고리즘과 제시된 알고리즘을 비교하기 위한 것이다. 예제 5는 150개의 데이터를 9개의 영역으로 랜덤하게 분포시킨 문제이며, 예제 6은 분류하기가 가장 난해하다고 알려진 회전데이터 문제(spiral data problem)이다.

역전파 알고리즘은 두 예제 모두 학습 수행회수를 15,000 epoch로 하였다. 특히, 역전파 알고리즘은 최적 신경망 구조를 스스로 결정하지 못하기 때문에 본 실험에서는 예제를 잘 해결할 수 있는 구조를 시행착오적으로 구하여 학습결과를 도출하였다.

두 가지 예제의 수행결과는 <그림 9>, <표 4>, <표 5>에 나타나 있다. <그림 9>에서 볼 수 있듯이 예제 5와 예제 6의 수행 결과 제시된 알고리즘이 역전파 알고리즘에 비해 보다 부드러운 분류 경계를 생성하고 있음을 알 수 있다. 생성된 분류 경계는 신경망의 일반화 능력(generalization capability)을 평가할 수 있는 근거가 된다. 제시된 알고리즘의 수행결과 얻어진 바와 같은 부드러운 경계

영역은 트리 학습 알고리즘(CART, C4.5, CHAID)으로는 얻지 못하는 것으로, 제시된 신경망의 분류 능력은 상당히 우수하다고 보여진다. 역전파 알고리즘으로 생성된 경계영역이 매우 거친 것은 적정하지 않은 수의 노드로서 학습이 수행되면 필요한 연결강도들이 잡음(noise)을 전파하게 되어, 분류경계의 생성에 있어 좋은 결과를 주지 못한 결과로 보여진다.



〈그림 9〉 학습데이터와 구해진 분류경계 (좌: 제시된 방법, 우: 역전파 방법)

〈표 4〉를 살펴보면 제시된 알고리즘은 역전파 알고리즘에 비해 적은 수의 epoch를 거치고도 학습 종료후의 오차가 각각 0.0038과 0.0324로서 역전파 알고리즘의 0.0324와 0.031에 비해 작음을 알 수 있다. 또한 사용된 노드의 숫자도 각각 16개와 14개로 역전파 알고리즘의 26개와 36개보다 오히려 훨씬 적은 수의 노드를 사용하여 문제를 해결할 수 있었다. 역전파 알고리즘의 경우 제시된 알고리즘에서 얻어진 신경망 구조와 같은 규모의 신경망을 가지고 학습한 결과, 두 예제 경우 모두 오차가 0.05를 넘는 좋지 않은 결과를 나타내었다. 이는 역전파 알고리즘의 경우 은닉노드의 학습효과가

높지 못하여 과도한 수의 은닉노드를 필요로 할 수 있는데 반해 제시된 알고리즘은 은닉노드가 주요학습대상 데이터를 효과적으로 분류함으로써 분류경계를 잘 찾아감을 의미한다.

〈표 4〉 예제 5, 6의 학습결과 비교

결과		학습예제 수	학습시간 (epoch)	오차 (MSE)	층의 수	노드의 수
5	제시된 모형	150	13291	0.0038	3	16
	역전파 모형	150	15000	0.0324	3	26
6	제시된 모형	80	14851	0.0028	3	14
	역전파 모형	80	15000	0.031	3	36

학습이 종료된 이후의 신경망의 구조는 아래의 〈표 5〉와 같다. 〈표 5〉에서의 역전파 모형의 구조는 학습예제를 풀 때까지 반복 학습하여 결정하였다.

〈표 5〉 예제 5, 6의 신경망구조 비교

결과		층의 수	은닉 및 출력층의 노드 수		
학습집합			1층	2층	3층
5	제시된 모형	3	11	4	1
	역전파 모형	3	15	10	1
6	제시된 모형	3	11	2	1
	역전파 모형	3	20	15	1

5. 결론 및 추후 연구과제

본 연구에서는 기존의 다층신경망 학습규칙이 연속적 입력값을 갖는 데이터를 분류하는데 있어 발생하는 단점을 보완하기 위하여 새로운 학습 알고리즘을 제시하였다. 제시된 알고리즘은 학습 과정에서 스스로 은닉층수와 은닉노드의 수를 증가시켜 나감으로써 신경망의 구조를 설계해 나가며, 적절히 확장된 은닉층과 은닉노드수로서 학습 데이터를 분류해 낼 수 있었다. 신경망의 구조를 설계하기 위한 척도로서는 엔트로피를 사용하였으며, 은닉노드의 학습효율을 높이기 위하여 학습

시에 주요학습대상 데이터를 지정하는 방안을 제시하였다. 이렇게 주요학습대상 데이터를 지정하여 줌으로써 학습의 진행을 방해하는 데이터를 추가되는 은닉노드가 학습하여 빠른 학습을 할 수 있도록 하였다.

신경망의 구조설계의 기법은 크게, 큰 신경망 구조에서 시작하여 줄여나가는 방법과 최소한의 신경망 구조에서 증가시켜 나가는 방법으로 구분된다. 신경망의 연결강도를 줄여나가면서 구조를 결정하는 방법은 초기의 신경망 구조를 결정하는데 어려움이 있으며, 너무 큰 구조의 신경망으로 시작할 경우 학습의 속도가 느려지는 단점을 가지고 있다. 따라서 본 연구의 신경망구조설계 방법은 최소한의 신경망 구조에서 최적의 신경망 구조로 발전시켜나가는 방법을 사용하였다. 신경망의 구조설계 알고리즘은 Marchand et al.[14]이 제시한 순서적 은닉노드 학습법과 유사한 방법으로 학습된다. 하지만 신경망의 구조설계 알고리즘에 층의 증가를 위해 엔트로피라는 새로운 척도를 추가하여 보다 복잡한 문제해결을 위한 신경망의 구조를 설계할 수 있도록 하였다.

본 연구에서는 제시된 알고리즘의 평가를 위하여 여섯 가지의 예제에 대한 시뮬레이션을 수행하였다. 실험결과 역전파 알고리즘보다 정교한 분류경계와 적은 오차로 학습을 완료하였다. 제시된 과정에 의해 적절히 설계된 신경망의 구조는, 역전파 알고리즘의 최적수행을 위해 시행착오적으로 얻어진 신경망구조보다 더 적은 수의 노드로 구성되어졌다. 다만 제시된 알고리즘이 역전파알고리즘에 비해 실험결과 적은 수의 epoch 내에 작은 오차로 학습을 완료하였으나 소요시간을 척도로한 역전파 알고리즘과의 비교분석은 완전히 이루어졌다고 볼 수는 없다. 이는 역전파 알고리즘과 제시된 알고리즘은 학습과정이 상이하며 역전파알고리즘의 학습 속도 개선을 위한 연구도 나뉘도록 진행되고 있기 때문이다.

제시된 알고리즘의 단점은 일반적인 역전파 알고리즘에 비해 단계가 복잡하며 신경망을 확장시

키는 과정이 언제 끝나게 될 지 모른다는 것이다. 또한 신경망이 확장될 때 일부 연결강도만을 변경시키는 과정이 있음으로 해서 시간이 낭비될 소지가 있는데 이는 결과적으로 추가된 은닉노드의 학습방향을 최적으로 조정하기 위한 것으로 시간차원의 낭비가 되는지 판단하기 위해서는 보다 면밀한 분석이 필요하다. 추가되는 은닉 노드가 주요학습대상 데이터를 먼저 학습하는 것은 마치 데이터의 분류를 위해 트리 학습알고리즘을 적용시킨 효과를 가져다 주게 되는데 일반적인 트리학습 알고리즘에 비해 신경망은 훨씬 부드러운 분류경계를 생성한다. 다만 일반적으로 알려진 바와 같이 신경망은 생성 분류경계에 대한 설명력이 결여된다는 단점이 있다.

제시된 알고리즘이 대용량의 실제 데이터베이스를 이용한 데이터마이닝에 사용되기에는 추가적인 연구가 필요하다. 먼저, 현실의 데이터는 결측치와 이상치가 존재하며, 많은 수의 데이터에 대해 완전한 분류를 실행하였을 경우 과학습(overfitting)이 발생하기 때문이다. 이러한 점을 보완하기 위해서는 데이터를 적절한 크기로 샘플링하여 규모를 줄이고, 학습성공률에 대한 목표수준을 과하게 잡지 않은 상태에서 제시된 학습알고리즘에 의해 적절한 신경망의 구조를 설계한 후, 전통적인 신경망의 학습방법을 적용시키는 방법이 있을 것이다. 또한 신경망의 문제점인 설명력의 결여를 극복하기 위한 방안으로, 제시된 알고리즘이 트리의 학습방법의 개념과 다소 유사하다는 점에서 트리 학습방법에서와 같은 설명과정을 개발하는 것도 의미있는 연구가 될 것이다. 기타의 추후 연구과제로는 신경망 구조를 확장하는 것과 과도한 신경망 구조를 축소하는 것을 포함한 종합적 신경망구조 설계 및 학습알고리즘을 개발하는 것이 있다.

참 고 문 헌

- [1] 강병서, 다변량 통계분석, 한현사, 1993.
- [2] 김대수, 신경망 이론과 응용(I), 하이테크경

- 보, 1992.
- [3] 오영환, "패턴 인식의 개관," 정보과학회지, 제 11권, 제5호, 1993.
- [4] 이대식, 이종태, "개선된 Hybrid-Neural Network를 이용한 패턴분류에 관한 연구," '95추계학술대회 발표논문집, 대한산업공학회, 1995, pp.406-410.
- [5] Cios, K.J. and L. Ning., "A machine learning method for generation of a neural network architecture : a continuous id3 algorithm," *IEEE Transactions on Neural Network*, Vol 3., 1992, pp.280-291.
- [6] Daesik, L., K. Taehoon. and R. Jongtae., "Improved learning technique of multi-layered neural network by automatic architecture modification," *In Proceeding of the 20th International Conference on Computers and Industrial Engineering*. Vol.2., 1996, pp. 773-776.
- [7] Fahlman, S.E. and C. Labiere., "The cascade-correlation learning architecture," In Touretzky, D.(ed.), *Advances in Neural Information Processing Systems 2*. Morgan Kaufmann, 1990, pp.524-532.
- [8] Frean, M., "The upstart algorithm : a method for constructing and training feed-forward neural networks," *Neural Computation 2*, 1990, pp.198-209.
- [9] Gallant, S.I., *Neural Network Learning and Expert Systems*, A Bradford Book, 1992.
- [10] Gallant, S.I., "Perceptron-Based Learning Algorithms," *IEEE Transactions on Neural Network*, Vol.1., 1990, pp.179-191.
- [11] Harp, S.A., A.T. Skapura. and A. Guha., "Designing application-specific neural networks using the genetic algorithm," *Advances in Neural Information Processing Systems 2*, 1988, pp.177-185.
- [12] Hinton, G.E., "Learning distributed representations of concepts," *In Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, 1986, pp.1-12.
- [13] John, H. and K. Anders., *Introduction to The Theory of Neural Computation*, Addison-Wesley, 1990.
- [14] Marchand, M., M. Golea. and P. Ruján., "A convergence theorem for sequential learning in two-layer perceptrons," *Europhysics Letters 11*, 1990, pp.487-492.
- [15] Mézard, M. and J.P. Nadal., "Learning in feedforward layered networks : the tiling algorithm," *Journal of Physics*, 1986, pp.2191-2204.
- [16] Miller, G.F., P.M. Nadal. and S.U. Nadal., "Designing neural network using genetic algorithm," *In Proceeding of the Third International Conference on Genetic Algorithm*, 1987, pp.379-384.
- [17] Minsky, M.L. and S.A. Papert., "Perceptrons," MIT Press, 1969.
- [18] Scalettar, R. and A. Zee., "Emergence of grandmother memory in feed forward networks : learning with noise and networks : learning with noise and forgetfulness," *In Connectionist Models and their Implications : Readings from Cognitive Science*, 1988, pp.309- 332
- [19] Sietsma, J. and R. J.F. Nadal., "Neural network pruning why and how," *IEEE International Conference on Neural Network*, Vol.1., 1988, pp.325-333.