

## 신경망 처리 시스템을 위한 학습 알고리즘의 구현

전산정보처리과 황 인 선  
전 임 강 사

### I. 서 론

산업사회에서 정보화 사회로 전환되어 가면서 누구나 빠른 시간내에 다량의 정보를 용이하게 얻고자 하며, 고속처리에 의한 대규모 상호 연결을 통하여 문제를 해결하고자 하는 새로운 계산모델 신경망(Neural networks)이 이용되어지고 있다.<sup>1)</sup>

신경망은 생물학적 신경세포의 구조와 기능을 매우 간단하게 논리적으로 모델화하여 기존의 연산방식보다 처리량의 한계를 극복하여 주어진 문제에 대한 정확한 해결방법을 모르는 경우에도 자체적으로 학습능력을 사용하여 문제의 특성에 적당한 해결을 하여 준다. 그리고 신경망은 입력공간과 출력공간 사이의 사상(mapping) 특성에 따라 연상기억, 필터, 변환, 분류, 인식 및 최적화 기법 등의 기능체제를 구성하여 컴퓨터를 사용하는 시각처리(visual process), 음성인식(voice recognition), 망 구성(networking topology), 인공지능, 지식표현(knowledge representation), 로봇제어(robotics control) 등의 분야에서 잡음(white noise)이나 변형된 정보의 처리에 매우 효과적으로 사용된다.<sup>2)</sup>

신경망에서 서로 연관된 입력과 출력이 올바르게 사상되도록 하는 것을 학습과정이라고 하며 중간층(middle layer)을 포함하고 있는 구성내에서 비 선형적인 문제를 학습시키기 위한 방법으로서 물리적인 장치를 사용하여 구현된 하드웨어적 방법보다도 활성화 함수의 선형 해석을 용이하게 하기 위한 소프트웨어적인 기법을 통하여 기존의 컴퓨터 시스템에서도 보조 장치의 도움없이 적응력을 향상시키고 문제 해결을 원활하게 처리하도록 하고 있다.<sup>3)</sup>

그러나 신경망 체계의 능력은 오히려 많은 문제점을 가지고 있다. 왜냐하면 이를 사용하여 답을 구하는 데는 용이하나 문제에 대한 해결 과정을 분석하여 사전 지식을 표현하는데 어려움이 많기 때문이다.<sup>4)</sup> 따라서 신경망 자체에 대한 정형화된 알고리즘이 필요하며 이와 같은 해석이 신경망 처리 시스템내에서 이루어져야 한다.<sup>5)6)</sup>

본 논문에서는 이를 위하여 신경망의 작동원리를 학습기법에 적응하여 신경망내에서 정보 전달의 처리 절차와 정형화된 알고리즘을 제안하였다.

## II. 신경망 처리 시스템

### 1. 신경망의 구조

본 논문에서 사용한 신경망은 그림 1 과 같이 신경망의 기본단위인 뉴론(neuron) 과 그 사이의 시냅스 연결(synapse link)이다. 따라서 신경망을 구성할 때 뉴론의 입력에 따른 출력 특성을 결정하는 데이터로 주어진 입력과 출력의 쌍에 존재하는 사상(mapping)을 자체의 적응도(adaptability)에 의하여 생성하며 각층(layer)에서의 입력으로 앞의 모든 층의 출력과 처음의 입력을 동시에 받아들여 출력층이 모든 연결 각도를 갖도록 하는 완전한 연결 강도(connection weight)를 반복법에 의해 조정해 나가도록 한다.<sup>6)</sup> 또한 축적되어진 경험을 바탕으로 새로운 사함에 대해 효과적이고 효율적으로 반응할 수 있도록 스스로 변화해 갈 수 있는 능력을 갖춘 학습에 의하여 매개변수를 추정하는 최적화 기법이 이용되고 있다.<sup>6)</sup>

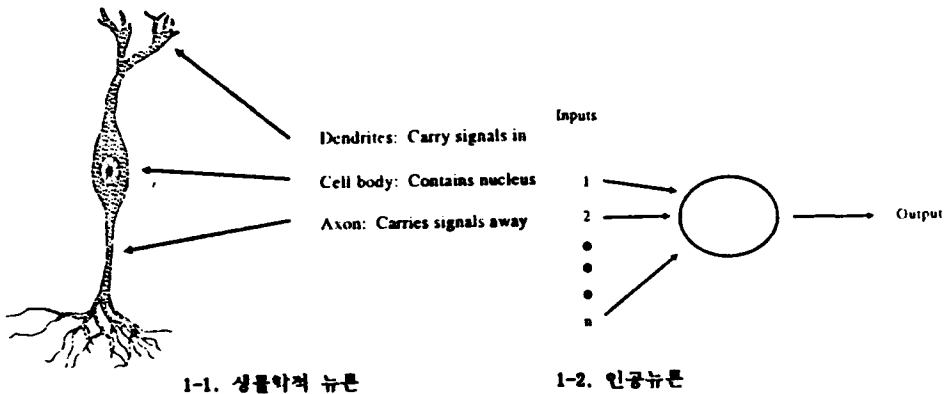


그림 1. 신경망의 구조

### 2. 신경망에서의 학습

신경망은 정상동작 상태 (working phase)와 학습상태 (learning phase)의 두가지 기능을 갖고며 경험이 반복됨에 따라서 어떤 일정한 자극에 대한 반응 양식을 변화, 조정, 발전시키는 능력을 학습이라고 한다.<sup>7)</sup> 생물학적 작용에 대하여 논리적인 표현방법은 지식기반을 토대로한 휴리스틱 모델내에서 입력조건에 대한 상호처리요소(interconnection processing element : 이하 PE) 즉 인공신경 (artificial neuron)과 선형 배열 그룹에 대한 층(layer)으로 표현한다. 여기서 그림 2는 학습을 위한 PE의 결합상태를 나타낸다.<sup>13)</sup>

PE는 그림 2와 같이 정보전송로인 연결경로에 의해 서로 병렬로 결합되어 있고 PE는 뉴론의 세포체를 표현하며 연결경로는 뉴론의 시냅스를 나타낸다.

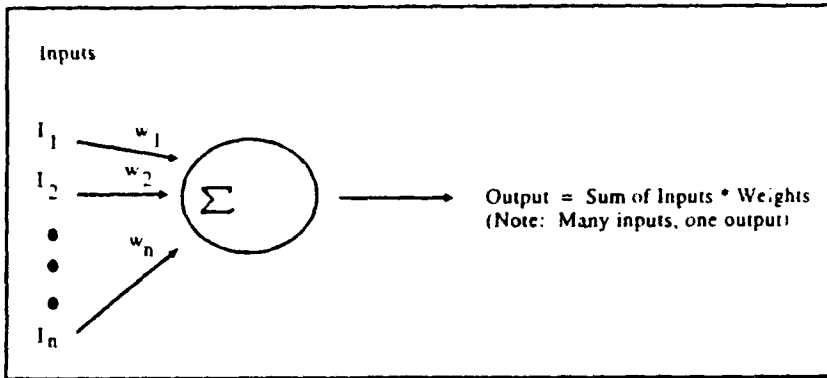


그림 2. 인공신경망의 구성

또한 수 많은 뉴론으로부터 여러개의 입력신호를 받아 처리되지만 결과를 생성할 때는 단 하나의 출력신호만을 발생시키고 이것이 여러개로 분리되어 그림 3에서 보여지는 바와 같이 제각기 다른 뉴론의 입력신호를 분산시키는 것이다.”

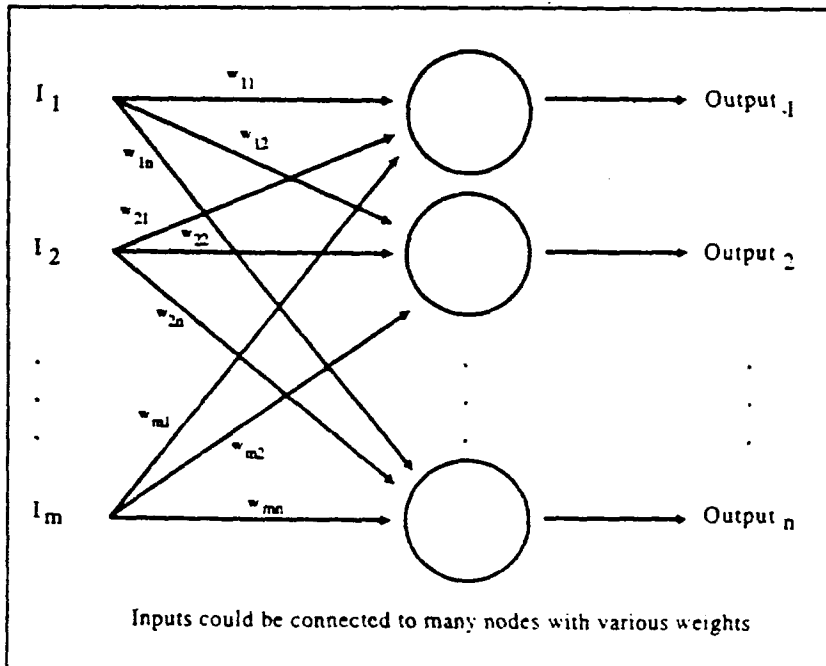


그림 3. 타 뉴론과의 결합 (층의 생성)

위와 같이 분산된 뉴론은 실제 출력과 원하는 출력 사이의 차이를 최소화시키는 방법으로 최적화 기법이 사용되고 있으며 이것은 지도 학습 (supervised learning) 알고리즘으로 이상

적인 출력과 실제 출력 사이의 평균 제곱 오차에 해당하는 비용함수의 값을 최소화 하기 위하여 경사정보 (gradient information) 방법에 의해 신경망의 전체 오차를 최소화하도록 연결강도를 구하기 위해  $n$ 개의 입력에 각각 연결경로에 해당하는 가중치  $W$ 를 주고 이의 합에서 다음과 같이 반복식을 적용하는 것이다.

$$W_{a-1} = W_{a+1} (\partial E / \partial W) \dots\dots\dots (1)$$

$a$  : 학습률

또한 출력의 결과를 입력쪽으로 back propagation 해주는 연결 경로의 유무에 따라 고정적인 학습이 수행되므로 학습률  $a$ 의 변화에 따라 뉴런간의 연결경로  $W$ 는  $(-\infty, \infty)$  사이의 실수이다.

### 3. 신경망 처리 시스템의 구성

신경망의 문제해결방식은 간단한 식 1을 반복적으로 적용하기만 하면 처리되는 경사추적 기법이기 때문에 정형화된 시스템을 구성하여 단계적 (step by step)으로 처리절차를 추적하는 모델이 필요하다. 따라서 본 연구에서는 정형화된 알고리즘의 해를 얻을 수 있도록 하기 위하여 그림 4 처럼 상태를 구성하여 신경망의 한 노드(뉴런)는 그 입력으로 들어오는 값에 적절한 가중치 (weight)를 곱하여 모두 더한 후 자신의 임계값 (critical value) 보다 크면 +1, 작으면 -1를 출력하는 임계함수 (critical function)를 구한다.

$$y = f(\sum W_i X_i - \theta) \dots\dots\dots (2)$$

$$\text{otherwise } f(X) = +1, \text{ if } x > 0$$

$$f(X) = -1, \text{ if } x \leq 0$$

식 2에서와 같은 상태표현에서 구성된 상태도 (state diagram)은 그림 4와 같다.

여기서 각 단말은 뉴론을 표현하며 뉴론과 시냅스의 구성을 인공적으로 재현하여 접속하고 있다. 각각의 뉴론은 +1과 -1의 두가지 상태를 가지며 현재 뉴론의 수를 4개로 하여 4개의 뉴론( $N_1, N_2, N_3, N_4$ )의 상태를 ( $X_1, X_2, X_3, X_4$ )로 한다.

## III. 정형화 알고리즘

신경망의 학습을 위하여 본 논문에서 제안한 정형화 알고리즘은 다음과 같이 나타낸다.

step 1. interconnection : ← initialization

step 1. input : output : ← assignment

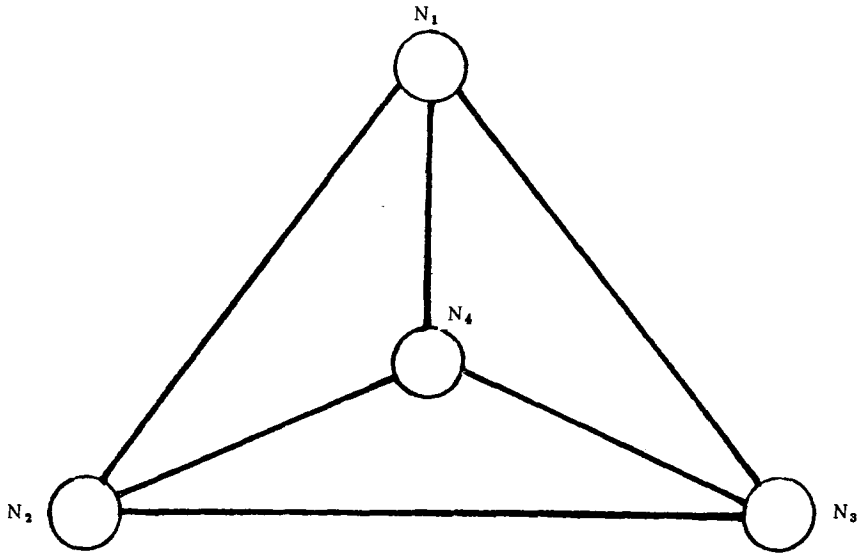


그림 4. 신경망 시스템의 상태도

- step 3. output layer : ← computation
- step 4.  $output = F(\sum W_{ij} \cdot X_i)$  : ← computation
- step 5.  $W_j$  : ← compare
- step 6. layer : ← add

처음에는 모든 연결강도의 값을 초기화 하여 입력과 출력층에 이상 출력값을 설정하여 출력층의 뉴런이 실제 출력값(actual output value)를 계산한다. 또한 실제 출력값과 원하는 출력값을 비교하여 학습을 끝내고 그렇지 않으면 원하는 출력값에 대한 실제 출력값을 변연경로의 값을 비교하여 정해진 일정 횟수의 반복 처리후 원하는 결과가 나오지 않을 때 층을 하나 증가시킨 후 문제 해결과정을 추적하여 모든 층에서의 출력과 원하는 입력을 새로운 입력으로 하여 2 단계에서부터 반복하여 확정된 결과를 신경망 처리 시스템에서 처리한다.

#### IV. 알고리즘의 구현

본 논문에서 이용된 알고리즘은 IBM호환의 PC/AT이며 C-언어로 그림 5의 전 과정을 구현하였다.

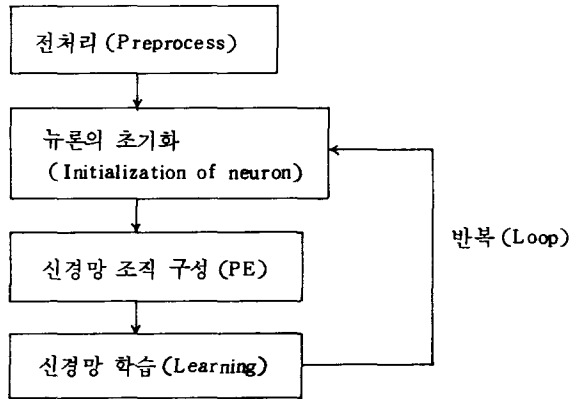


그림 5. 전체 학습 알고리즘 순서

3장에서 설명된 step1의 과정을 다음 알고리즘으로 표기하며 모든 연결경로에 대하여 초기화를 수행하며 전처리 작업으로서 수행시간은  $O(n \log n)$  복잡도를 갖는다.

```

void h_init ( )
{
  int i ;
  int j ;
  for ( i = 0 ; i <= imax ; i ++ )
  {
    for ( j = 0 ; j <= jmax ; j ++ )
    {
      b[ba(i , j , 0)] = cha 0 ;
      b[ba(i , j , 1)] = cha 0 ;
      b[ba(i , j , 2)] = cha 0 ;
    }
  }
}
  
```

주어진 문제의 해를 구하기 위하여 초기화된 뉴론을 복합 구성하여 입력의 값에 대한 정형화된 출력값을 얻을 수 있도록 하며, 또한 문제의 풀이 과정을 추적 분석해 나갈 수 있다. 위와같이 실제 학습을 하는데 필요로 하는 시간은  $O(n^2)$ 이 필요하며, 일반적 많이 사용되는 Back propagation 알고리즘에 있어서도 유효 시간이  $O(n^2)$ 가 비교할 때 본 알고리즘에서 구현된 상태의 복잡도가 같음을 볼 수 있고, 이 알고리즘은 문제의 분석과정에서는 유용성 있게 활용할 수 있다.

```

void h_learn(sw)
{
    if (sw=1) {
        cprintf (" find neuron");
    }
    for (i=0; i <= imax; i++)
    {
        for (j=0; j <= jmax; j++)
        {
            if (j=0) {
                x1=b[ba(i,j,0)]; x2=b[ba(i,j,1)]; x3=cha0;
            } else {
                x1=b[ba(i,j,0)]; x2=b[ba(i,j,1)]; x3=b[ba(i,j-1,2)];
            }
            n[na(i,j,0)]=cxor3(x1, x2, x3);
            if (i=0) {
                x1=b[ba(i,j,0)]; x2=b[ba(i,j,2)]; x3=cha0;
            } else {
                x1=b[ba(i,j,0)]; x2=b[ba(i,j,2)]; x3=b[ba(i-1,j,1)];
            }
            n[na(i,j,1)]=cxor3(x1, x2, x3);
        }
    }
}

```

## V. 결 론

본 논문에서는 기존의 신경망 기법에서 이용하고 있는 back propagation에 있어서 문제의 해를 보완하기 위하여, 주어진 문제의 추적을 분석할 수 있는 정형화된 알고리즘을 제안하였다. 또한 이를 실제 프로그래밍하여 신경망 처리 시스템내에서 유용성을 입증하였다. 이 알고리즘은 주어진 문제의 해를 추적할 경우 수행시간에 있어서 전처리시간은  $O(n \log n)$ , 저장공간  $O(n)$ , 질의 응답시간  $O(n^2)$ 의 복잡도(time complexity)를 나타내었다.

본 연구에서 발생된 문제점으로서는 한정된 뉴론을 표본화 하여 이용되어 짐으로써 고속성과 추적기능에 대하여 확실한 해를 얻을 수 있으나 생물학적 신경조직의 뉴론과 같이 수많은 뉴론이 결합하여 사용되어질 경우에는 저장공간이 확대되어 소형 컴퓨터에서는 처리의 해를 구하는데 한계성에 이르리라 사료된다.

추후 연구 방향으로서는 보다 더 수행시간을 압축하여 생물학적 뉴론의 반응시간과 유사

하도록 소형 시스템에 적합한 고속 알고리즘이 개발되어야 하며, 많은 입력 자료를 가지고 선형 활성화 함수의 구간을 적절하게 신경망의 입력 형태로 변환시키는 부분에 대한 연구가 진행되어야 할 것이다.

### 참 고 문 헌

1. Nelson, Marilun McCord : *A practical guide to Neural Nets*, Addison wealey, USA 344(1991)
2. V. Vermi : *Neural Network - Artifical Neural networks*-computer society press, USA, 286 (1988)
3. M. AA. Arbib : *Lecture in Neural Networks and AI*, Information processing, USA(1988.6)
4. KiTE computer Society : *Computer Technology-Neuro computing*, 6(1989.6)
5. J. F. Shepanski : *Fast Learning in Artifical Neural systems*, IEEE IJCNN proceedings, 1, 465 ~ 472(1988).
6. 김병주, 김광백, 차의용 : “선형 활성화 함수를 사용한 Back propagation 알고리즘”, 한국정보과학 회지 18, 131 ~ 134(1991)
7. R. Hecht Nielsen : *Theory of BACK-propagation Network*, IEEE IJCNN proceeding, 1, 593 ~ 605(1988)



**A Study on the implementation of learning  
algorithm for neural net processing system**

**Hwang, In Sun**  
*Dept. Computer & Information Processing*  
*Kwangju Health Junior College*

*>Abstract<*

Many neural net system is to solve optimum solution, use effective Back propagation algorithm.

But solution of problem needs how to find it, when it come to taking a course such a question and explanation to go on trace, analysis of formatted learning algorithm.

This paper substantiated of value for them by implement formatted learning algorithm. The result of time complexity is preprocessor time  $O(n \log n)$ , space done  $O(n)$  and query-answer time  $O(n^2)$ .