

# 신경망을 이용한 전문가 시스템의 구현

○권 회선\*, 김 배섭\*, 권 호열\*\*, 이 상희\*\*  
\*한림대학교 전자계산학과, \*\*강원대학교 전자공학과

## An Implementation of Connectionist Expert System

○H.S.Kwon\*, B.S.Kim\*, H.Y.Kwon\*\*, S.H.Lee\*\*  
\* Dept. of Computer Science, Hallym Univ.  
\*\* Dept. of Electronics Engineering, Kangwon National Univ.

### ABSTRACT

To resolve the knowledge acquisition bottleneck in the expert systems, the connectionist expert systems have been proposed, which facilitate learning capability of neural networks.

This paper is to modify Gallant's connectionist expert network so that it can be applied to more general problems :

- 1) The hidden nodes are added between the input nodes and an output node, so that the back propagation learning algorithm is used instead of perception based Pocket algorithm.
- 2) Inference engine is thus modified by modeling that a node may have uncertainties due to unknown inputs.

### 1. 서론

인공 신경망 모델(Artificial Neural Model or System)은 신경과학(neuroscience) 분야의 개념과 정보를 일부분 이용하여 기존의 디지털 컴퓨터의 직렬 처리에 의해서는 해결이 어려웠던 실제계의 문제에 적용하고자 하는 시도로서 개발되었다. 개발 초기에 제안되어 각광을 받았던 퍼셉트론 학습 방법에 대한 비판[2]과 함께 잠시 위축되었던 신경망 모델에 관한 연구는 그 후 은닉층을 가지는 망구조에 대한 학습 알고리즘인 오류 역전파 알고리즘[3]이 제안되어 그 구조의 간단함과 학습의 실험 성공에 힘입어 현재 널리 쓰이고 있다.

전문가 시스템에서는 적합한 지식 습득의 병목 현상을 해결하기 위한 목적으로 규칙 기반 지식 베이스를 신경망 모델을 이용하여 구현하려는 연구가 있었다 [1][4][5][6]. 이중 Gallant[1]에 의해 제안된 전문가 시스템은 학습에 의한 지식베이스 생성, 부분 정보로부터의 추론, 질의 생성, 추론 결과에 대한 설명기능을 갖추고 있으나 사용한 학습 알고리즘이 퍼셉트론에 기반을 두고 있기 때문에 비선형 판별 문제의 경우에 적용하는 데는 어려움이 있다.

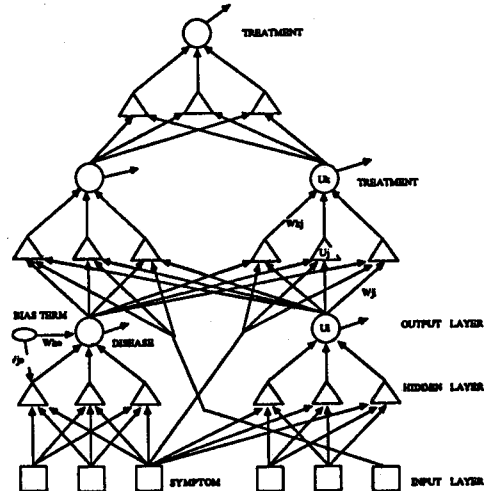
본 논문은 Gallant[1]이 제안한 connectionist 전문가 시스템의 구조를 일부 변경하여 일반화된 문제에 대한 적용 가능

성을 보여 주고자 한다. 차이점은 첫째, Gallant[1]의 경우는 단층 퍼셉트론에 기초를 두고 필요한 경우 임의의 중간노드를 사용한 구조이나 본 논문에서는 망내의 모든 출력노드가 은닉층 노드들을 가지는 다층 퍼셉트론 구조로 망을 구성하고 포켓 알고리즘 대신 오류 역전파 알고리즘으로 학습시켰다. 둘째, 그에 따라 추론엔진 부분이 새로운 구조에 적합하도록 변경되었다.

### 2. 지식 베이스 구현을 위한 신경망 모델

#### 1) 망 구조

노드들 사이의 의존 관계가 의존 행렬 형태로 주어지면 이를 사용해서 망구조가 결정된다. 이는 Gallant[1]의 경우와 비슷하나 본 논문에서는 은닉층을 갖는다는 점만이 다르다 [그림 1]. 제안된 망구조는 다음과 같은 특성을 갖는다.



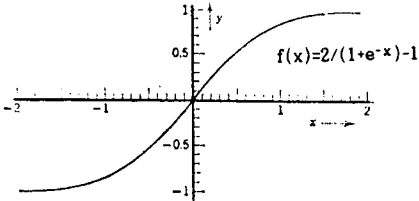
[그림 1] 제안된 망구조의 예

① 입력층의 노드들을 제외한 모든 노드들은 각각 의존 행렬에 의해 정해진 갯수의 입력노드를 가지며 은닉 노드의 갯수는 해결하는 문제에 맞게 정해지고 각각은 출력 노드 이자 다른 노드의 입력 노드로 동작할 수도 있다.

② 각각의 노드들은 (+1, -1, 0) 중에서 값을 갖는다. 이 중 '0' 인 값의 의미는 입력층의 노드인 경우에는 아직 값이 입력되지 않았음을 나타내고 그외의 노드들에 있어서는 아직 그 노드의 값이 결정되지 않았음을 나타낸다.

입력층 노드들은 외부로부터 입력된 값을 그대로 가지는 반면 그 이외의 노드의 값은 계산된 입력값과 임계값에 대응하는 활성화 함수값으로 결정된다.

본 논문에서 사용한 활성화 함수는 그림 2와 같다.



〈그림 2〉 활성화 함수.

③ 노드들 사이의 의존 관계의 강도를 표현하기 위한 가중치는 오류 역전파 학습 알고리즘에 의해 학습된다.

## 2) 오류 역전파 학습 알고리즘에 의한 가중치 학습

입력층의 노드들을 제외한 각각의 노드에 대하여 입력 학습 데이터와 그에 대응되는 기대값이 주어진다. 주어진 학습 데이터에 대하여 다음과 같은 오류 역전파 학습규칙[7]을 적용하여 가중치 및 임계값을 학습한다. 본 논문에서 사용되는 기호는 [7]을 따른다.

```
do{
  for all output nodes in network
  {
    for all hidden nodes
      that connected with output node
      {
        h      h
        NETpj = ∑ Wji Xpi + θj ; /*θ: 임계값*/

        h      h      h
        Ipj = f(NETpj) = 2.0 / (1.0 + exp(-NETpj)) - 1.0;

      }
      o
      NETpk = ∑ Wkj · Ipj + θk;

      o
      Opk = 2.0 / (1.0 + exp(-NETpk)) - 1.0;

      o
      σpk = (Ypk - Opk) f'(NETpk); /*Yk: 이상적인 출력값
                                       Ok: 실제 계산에 의한
                                       출력 값*/

      h      h      o      o
      σpj = f'(NETpj) · σpk · Wkj;

      o      h      h
      Wkj(t+1) = Wkj(t) + η · σpk · Ipj ; /* η: 학습 비율*/
      h      h      h
      Wji(t+1) = Wji(t) + η · σpj · Xpi;
    }
  }
} until (Ep < MAXERROR);
```

본 논문에서는 Gallant[1]에서 사용된 8개의 학습 데이터를 사용하였다. 가중치의 초기값은 [-1, 1]사이의 임의의 수로 하였고, 임계값의 초기값은 [0.0, 0.2]사이의 임의의 수로 하여 학습시켰으며 위 알고리즘에서 사용된 함수 및 변수 값은 다음과 같다.

- E<sub>p</sub> = 1/2 ∑<sub>k</sub> (Y<sub>pk</sub> - O<sub>pk</sub>)<sup>2</sup> : 평균제곱 오차
- f'(NET<sub>k</sub>) = (1/2) (f(NET<sub>k</sub>) + 1) (1 - f(NET<sub>k</sub>))
- MAXERROR = 0.01
- η = 0.5 : 학습 비율

학습된 신경망은 전문가 시스템의 지식 베이스로써 추론 엔진에 의한 논리적 추론을 위해 사용되어진다.

## 3. 추론 엔진의 구현

추론 엔진은 초기에 사용자로부터 추론에 필요한 단편적인 정보를 제공 받은후, 이 정보를 바탕으로 순방향 탐색을 이용 각각의 노드들에 대해 결론에 도달할 가능성을 추론한다. 결론에 도달하지 못한 노드들에 대해서는 결론에 도달할 가능성이 가장 큰 노드를 선택하여 역방향 탐색으로 질문을 선택한 후, 얻어진 정보를 이용해서 다시 추론을 반복하는 알고리즘에 의해 동작하며 필요에 따라 사용자에게 추론에 대한 설명을 제공할 수 있다.

### 1) 추론(순방향 탐색)

입력 노드를 제외한 노드들의 값을 결정하기 위해 순방향으로 노드들을 탐색하면서 결론의 도달 여부를 검사한다. 여기에서 문제는 입력값이 부분적으로 알려진 상태에서 값을 결정하고자 하는데 있다. 즉, 어떤 노드의 출력값은 값이 알려지지 않은 입력 노드의 영향으로 인해 불확실성을 가지게 되며, 불확실성의 정도는 다음과 같이 노드에 대해 계산된 입력값의 범위로 결정된다.

현재 알려진 입력 노드의 값으로부터 계산된 은닉층에 대한 입력값에, 알려지지 않은 입력 노드가 가지는 가중치의 합을 각각 가감하여 범위를 계산해 낸다. 계산된 최소값과 최대값 각각에 대해서 은닉층 노드들의 출력을 계산하고 여기에 관련된 출력노드 사이의 가중치가 곱해짐으로써 각각의 은닉 노드들이 출력 노드의 입력값에 미치는 영향력의 범위가 계산된다. 다음은 출력 노드가 가질 수 있는 입력값의 범위를 결정하기 위해 사용된 알고리즘이다

```
do{
  for all output nodes in network
  {
    for all hidden nodes that associated
      with each output node
    {
      h      h      h
      MINj = ∑ Wji · ai + ∑ |Wji|;
      known      unknown
    }
  }
}
```

$$MAX_j = \sum_{known}^h W_{ji} \cdot a_i - \sum_{unknown}^h |W_{ji}|$$

$$MIN_k = \sum \min [W_{kj} \cdot f(MIN_j + \theta_j), W_{kj} \cdot f(MAX_j + \theta_j)]$$

$$MAX_k = \sum \max [W_{kj} \cdot f(MIN_j + \theta_j), W_{kj} \cdot f(MAX_j + \theta_j)]$$

일단 출력노드에 대한 입력값의 범위가 구해지면, 이를 근거로해서 결론을 낼 수 있으며, 그 알고리즘은 다음과 같다.

```
do{
  for all output nodes in network
  {
    if (MINk > f-1(threshold)) then conclusion=1;
    else if (MAXk < f-1(-threshold)) conclusion=-1;
    else conclusion=0;
  }
}
```

본 논문에서 사용된 threshold 값은 0.7이고 f<sup>-1</sup>(x)는 지식 베이스 구성 과정에서 오류 역전과 알고리즘이 사용한 활성화 함수의 역함수이다.

### 2) 결론에 도달할 가능성 측정

추론 과정을 거친후 아직 결론에 이르지 못한 노드들이 있을 경우에는 사용자로부터 더 정보를 구해야한다. 이때 더 결론에 근접한 노드를 선택하여 적합한 정보를 사용자에게 요구하는 것이 의미가 있을 것이다. 결론에 근접한 노드를 선택하는데는 여러가지 방법이 있을 것이나 본 논문에서는 간단히 현재 알려진 입력 노드의 값만으로 계산된 출력값의 절대값을 사용하여 그값이 가장 큰 노드를 선택하기로 한다.

### 3) 질의 생성(역방향 탐색)

결론에 이를 가능성이 가장 큰 노드가 결정되면, 이 노드에 정보를 추가해 주기 위하여 적절한 질의를 골라야 한다. 망 내에서 의존 관계들의 영향력을 나타내는 요소는 가중치이므로 관련된 질의들 중에서 가장 영향력 있는 질의를 고르기 위해서는 가중치가 비교의 기준이 될 수 있을 것이다. 본 논문에서의 망구조는 은닉층을 포함하고 있으므로 다음과 같은 알고리즘이 적용된다.

② 2)에서 선택된 노드를 추구(pursue)노드로 한다.

① 추구 노드에 연결된 은닉 노드중 가중치의 절대값이 최대인 노드를 고른다. 즉, 그림 1에서  $U_k$ 가 추구노드라고 하면  $|W_{kj}| \geq |W_{kl}|, h \in \{ \text{노드 } U_k \text{에 연결된 은닉노드들의 인덱스} \}$ 을 만족하는 노드  $U_l$ 를 고른다.

② 노드  $U_j$ 와 연결 되어있고 값이 알려지지 않은 노드 중 가중치의 절대값이 최대인 노드를 고른다.

③ ②에서 선택된 노드가 입력층 노드이면 이 노드의 값을 사용자에게 물어보고, 아니면 이 노드를 다시 추구 노드로 한다. 예를들어 그림 1에서 노드  $U_1$ 가 선택 되었다면 이 노드를 추구 노드로 하여 ①-③과정을 반복 하게 된다.

### 4) 추론화 과정에서 내려진 결론에 대한 설명

사용자에 의한 설명 요구에 일일이 응답해 줄 수 있는 것은 전문가 시스템이 가진 장점중의 하나이다. 제안된 추론 엔진은 다음의 방법으로 결론에 대한 설명을 제공한다.

① 망구조에서 순수히 입력층 노드로써 동작하는 노드에 대해서는 그 입력층 노드가 영향을 주고 있는 출력층 노드를 설명으로 제공한다.

② 그 외의 노드들은 의존 행렬 상에서 자신과 연결 관계에 있으면서 결론을 내리게 되는데 영향을 주었던, 값이 결정된 노드에 대한 정보를 사용자에게 설명으로 제공하게 된다.

### 4. 실험 결과

Gallant[1]의 의존 행렬, 학습 데이터를 사용하여 제안된 방법으로 신경망을 이용한 전문가 시스템을 구성하였다. 구성된 망은 그림 1과 같이 은닉층을 포함하고 있다. 주어진 학습 데이터를 학습시킨 후 추론 엔진을 동작시킨 부분적인 결과를 그림 3에 보인다.

### 5. 결론

전문가 시스템의 지식 베이스를 이루는 노드들 사이의 의존 관계와 대응되는 법칙(rule)들에 해당하는 적절한 학습 데이터가 전문가로부터 주어진다면, 학습에 의해서 전문가 시스템의 지식베이스가 구축될 수 있다는 것을 보이고자 하였 본 논문에서는 Gallant[1]이 사용한 의존 관계 정보와 학습 데이터를 은닉층을 포함하는 망구조로 구성된 후 오류 역전과 알고리즘으로 학습시켰으며 구성된 망 구조에 적합하게 추론 엔진을 구현하였다. 제안된 망구조에서 Gallant[1]과의 차이점은 은닉층 노드의 사용으로 다중 퍼셉트론 구조를 가지게 되어 비선형 판별문제에 쉽게 적용될 수 있다는 점이며, 제안된 추론 엔진에서의 차이점은 순방향 추론과정을 출력 노드가 가지는 가능한 값의 범위가 입력 노드에서 은닉 노드들을 경유하여 전파되는 과정으로 구현하였다는 점이다.

본 논문에서는 비선형 판별 문제의 적용과 연속적인 값에 대한 처리가 가능한 오류역전과 학습 알고리즘에 의해 주어진 학습 데이터를 성공적으로 학습시킴으로써 보다 일반화된 시스템의 구현 가능성을 보였다.

1: Swollen feet  
2: Red ears  
3: Hair loss  
4: Dizziness  
5: Sensitive aretha  
6: Placibin allergy

Enter initial values for input variables.

Format: Variable number,value,...

PURSUIING:(26) Posiboost

PURSUIING:(21) Placibin

Is the patient allergic to Placibin?

--> y(es,n)o,u)known,?)explain.

n

PURSUIING:(21) Placibin

PURSUIING:(23) Supercilliosis

Does the patient have swollen feet?

--> y(es,n)o,u)known,?)explain.

n

CONCLUDE:(13) Supercilliosis IS FALSE

PURSUIING:(26) Posiboost

PURSUIING:(21) Placibin

PURSUIING:(14) Namastosis

Is the patient suffering from hair loss?

--> y(es,n)o,u)known,?)explain.

y

CONCLUDE:(22) Biramibio IS FALSE

PURSUIING:(21) Placibin

PURSUIING:(14) Namastosis

Is the aretha sensitive?

--> y(es,n)o,u)known,?)explain.

y

CONCLUDE:(14) Namastosis IS TRUE

CONCLUDE:(21) Placibin IS TRUE

CONCLUDE:(26) Posiboost IS TRUE

Would you like to try a new case?

--> y(es,n)o,?)explain

?

Enter 'e' for explaining of the

last question.  
Enter '#' for explanation of  
variable number(:ex. '5')

13

Supercilliosis IS FALSE  
due to the following rule:

IF Swollen feet IS TRUE.

IF Red ears IS FALSE.

Would you like to try a new case?

--> y(es,n)o,?)explain

n

[그림 3] 제안된 Connectionist 전문가 시스템을 동작  
시킨 예

## 6. 참고 문헌

- [1] S.I. Gallant, 'Connectionist expert system',  
Comm.ACM Vol.31, No.2, pp.152-169, Feb.1988
- [2] B.Widrow and M.E.Hoff, 'Adaptive switching  
circuits', IRE WESTON Conv. Record,  
part 4, pp.96-104, 1960  
Distributed Processing Cambridge,  
MA: MIT press, 1986
- [4] S.I.Gallant, 'Perceptron Based Learning  
Algorithms', IEEE Trans. on Neural Networks  
Vol.1, No.2, pp.179-191, 1990
- [5] R.C.Lacher, S.I.Hruska and D.C.Kuncicky,  
'Back-Propagation Learning in Expert Networks'  
, IEEE Trans. on Neural Net., Vol.3, No.1,  
pp.62-72, 1992
- [6] R.C.Lacher, Expert Networks:Paradigmatic  
Conflict, Technological Approchement
- [7] J.A.Freeman and D.M.Skapura, 'Neural Networks  
Algorithms, Applications and Programming  
Techniques', pp.89-124