

신경회로망을 이용한 전문가 시스템 개발에 관한 연구

박 영문 윤 지호* 손 동욱
 서울대학교 전기공학과

A Study on the Development of Expert System
 Using Artificial Neural Net

Park Young-Moon Yoon Ji-Ho* Son Dong-Wook
 Seoul National Univ. Dept. of Electrical Engineering

Abstracts

The most difficult, time-consuming, and expensive task in building an ES (Expert System) is constructing and debugging its knowledge base. Our goals are to eliminate the knowledge-acquisition bottle-neck for ES creation in data rich situations and to make an ANN (Artificial Neural Network) model behave as much as possible like an ES. The ANN ES has many benefits: Once it has been learned, inference time is very short. It can provide a reasonable conclusion for insufficient input data. But it has also several demerits: Learning time is too long to converge. We cannot guarantee the convergence of its weights. We introduce an ANN ES model which makes most of its benefits and compensates its shortcomings.

1. 서론

기존의 전문가 시스템은 주어진 분야에서 복잡한 문제를 해결하는 전문가의 사고 과정을 모방하는 전자계산기 하드웨어 및 소프트웨어로 구성된 시스템으로서, 특정 분야에서의 전문가 역할을 모의할 수 있는 컴퓨터 시스템이다 [1,2,3]. 지식공학자는 특정 분야의 전문가와 함께 지식베이스 (Knowledge base)를 구축한다. 사용자가 인간-기계 연결 기구 (Man-machine interface)를 통하여 문제를 입력하면 추론기관 (Inference engine)은 지식베이스에 저장되어 있는 지식을 바탕으로 전문가가 내릴 수 있는 정도의 해답을 얻어낸다. 지식베이스가 구축된 이후에 그 분야의 연구와 계속된 경험에 의하여 이전에는 발견되지 않았던 지식이 생겼을 경우에는 사용자가 인간-기계연결기구를 통하여 지식베이스를 수정 (Modification) 하게 된다. 전문가 시스템을 구축하는데 있어서 가장 시간이 많이 걸리고 어려운 일은 바로 지식 베이스의 구축과 관리이다. 과거의 데이터들만을 가지고, 자동으로 지식 베이스를 구축할 수 있다면 전문가 시스템의 구성이 용이해질 것이다. 이것의 실현을 위해서는 데이터로부터 룰(Rule)을 도출(Induction)해낼 수 있어야 한다. 즉 여러개의 입력 패턴과 그에 대응되는 출력패턴에서 내재되어 있는 룰들을 도출해낼 수 있는 알고리즘의 개발이 이루어져야 한다. 근래 들어서 인공신경회로망의 학습 특성을 사용하여 전문가 시스템의 지식 베이스를 구현하고자하는 연구가 많이 발표되고 있다. 인공신경회로망 전문가시스템의 초기 연구는 James A. Anderson에 의하여 이루어졌다. [4] 그는 Autoassociative memory [6,7] 모델을 사용하였다. 즉 입력 노드와 출력 노드를 대칭적으로 배열한 뒤, 원하는 입출력 관련쌍들을 가지고 학습시

킨다. 일단 학습이 되면 어느 정도의 불완전한 입력패턴이 들어가더라도 학습에 의거한 가장 가까운 패턴을 출력한다. 이후에도 Gallant, Stephan 등에 의하여 뒤 Anderson연구의 문제점들이 일부 보완되었으나 근본적인 문제점들은 해결이 되지 못한 상태이다 [8,9,10].

본 연구는 전문가 시스템의 구성시에 가장 큰 난점이 되고 있는 지식 베이스 구축을 자동적으로 행하는 알고리즘을 제시한다. 문제 분야의 전문가와 지식공학자가 없더라도 장시간에 걸쳐 축적된 데이터에 내재된 룰들을 생성하여 지식 베이스를 구성하고 계속 경험되는 데이터에 의해 스스로 지식 베이스를 확장하는 적응 전문가 시스템(Adaptive Expert System)을 개발하고자 한다. 이를 위해서 축적된 데이터군에서 지식 (또는 룰)을 얻어내는 알고리즘을 개발하고 이를 구현하는 새로운 신경회로망 전문가 시스템 모델을 제시한다.

2. 경험된 민텀 (Minterm) 에 의한 로직 룰 베이스

불확실의 정도를 고려하지 않는 결정적 (Deterministic) 전문가 시스템 (Expert system)의 경우에는 지식 베이스를 로직으로 구성할 수 있다. 한걸음 더 나아가서 새로운 경험에 의하여 룰 (Rule)을 생성하고, 또는 기존의 룰과 배치되는 입력이 들어왔을때 그것을 사용자에게 알려 줄 수 있다면 여러 분야에서 응용될 수 있을 것이다. 룰이란 그 어떤 것이라도 경험된 데이터 상호간에 내재되어 있는 것이므로 'IF A, THEN B'의 형태로 추출해 내지 않아도 되는 시스템에서는 추론에 의하지 않고 로직회로를 통하여 직접 그 결과를 얻어낼 수 있다. 만약 한 작은 시스템에서 상태변수 A, B, C 가 존재하고 'IF A THEN B & IF B THEN C'라는 내재된 룰을 가지고 있다고 가정하자. 장시간에 걸쳐서 변수의 값들을 기록했다면 절대 나오지 않는 항들과 나온 적이 있는 항들로 나눌수 있다. 즉:

	A	B	C	경험	미경험
m ₀	0	0	0	*	
m ₁	0	0	1	*	
m ₂	0	1	0		*
m ₃	0	1	1	*	
m ₄	1	0	0		*
m ₅	1	0	1		*
m ₆	1	1	0		*
m ₇	1	1	1	*	

그림 2-1
기록된 데이터

이러한 경험을 가지고 다음과 같은 기능 시스템 Y를 구성하고자 한다.

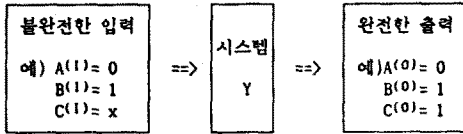


그림 2-2 불완전한 입력이지만 판단 할 수 있는 경우

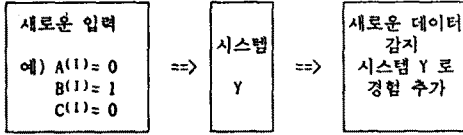


그림 2-3 경험에 없는 새로운 데이터가 들어온 경우

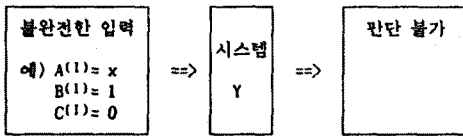


그림 2-4 불완전한 데이터이고 경험한 바가 없는 경우

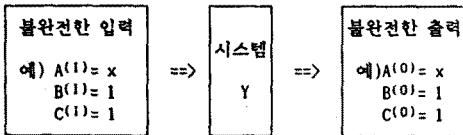


그림 2-5 불완전한 데이터이고 판단할 수 없는 경우

시스템 Y는 경험된 데이터에 의하여 미결정 입력을 결정하거나 새로운 경험임을 인식하여 저장 데이터를 추가한다. 하나의 변수는 True, False, Unknown의 세가지 상태를 가져야 하므로 2개의 비트를 사용한다. 즉

A	\bar{A}	상태
1	0	참
0	1	거짓
1	1	미결정
0	0	모순

그림 2-6

만약 입력값이 참, 거짓, 둘중에 한 값을 갖는다면 다음 식은 자명하다.

$$\begin{aligned}
 A^{(0)} &\Leftarrow A^{(1)}B^{(1)}C^{(1)} \\
 \bar{A}^{(0)} &\Leftarrow \bar{A}^{(1)}\bar{B}^{(1)}\bar{C}^{(1)} + \bar{A}^{(1)}\bar{B}^{(1)}C^{(1)} + \bar{A}^{(1)}B^{(1)}C^{(1)} \\
 B^{(0)} &\Leftarrow \bar{A}^{(1)}B^{(1)}C^{(1)} + A^{(1)}B^{(1)}C^{(1)} \\
 \bar{B}^{(0)} &\Leftarrow \bar{A}^{(1)}\bar{B}^{(1)}\bar{C}^{(1)} + \bar{A}^{(1)}\bar{B}^{(1)}C^{(1)} \\
 C^{(0)} &\Leftarrow \bar{A}^{(1)}\bar{B}^{(1)}C^{(1)} + \bar{A}^{(1)}B^{(1)}C^{(1)} + A^{(1)}B^{(1)}C^{(1)} \\
 \bar{C}^{(0)} &\Leftarrow \bar{A}^{(1)}\bar{B}^{(1)}\bar{C}^{(1)}
 \end{aligned}$$

((0), (1) 은 각각 출력, 입력임을 나타낸다.)

위의 식을 살펴보면 경험된 입력 패턴이 들어오면 출력 패턴은 입력과 같다. 그러나 경험된 것이 아니면 위배되는 입력은 모두 0(예를 들어 $A^{(0)}=0, \bar{A}^{(0)}=0$), 즉 모순이 된다. 이것을 감지하여 입력패턴이 새로운 것임을 알 수 있는 것이다. 만약 A가 미결정 ($A^{(1)}=1, \bar{A}^{(1)}=1$)이면 출력에 아무런 영향을 끼치지 못하게 되며 $A^{(0)}, \bar{A}^{(0)}, B^{(0)}, \bar{B}^{(0)}, C^{(0)}, \bar{C}^{(0)}$ 는 $B^{(1)}, \bar{B}^{(1)}, C^{(1)}, \bar{C}^{(1)}$ 의 상태에 의하여 결정이 된다. 이러한 상태를 식으로 써보면 다음과 같다.

$$\begin{aligned}
 A^{(0)} &\Leftarrow B^{(1)}C^{(1)} \\
 \bar{A}^{(0)} &\Leftarrow \bar{B}^{(1)}\bar{C}^{(1)} + \bar{B}^{(1)}C^{(1)} + B^{(1)}C^{(1)} \\
 B^{(0)} &\Leftarrow B^{(1)}C^{(1)} \\
 \bar{B}^{(0)} &\Leftarrow \bar{B}^{(1)}\bar{C}^{(1)} + \bar{B}^{(1)}C^{(1)} \\
 C^{(0)} &\Leftarrow \bar{B}^{(1)}C^{(1)} + B^{(1)}C^{(1)} \\
 \bar{C}^{(0)} &\Leftarrow \bar{B}^{(1)}\bar{C}^{(1)}
 \end{aligned}$$

((0), (1) 은 각각 출력, 입력임을 나타낸다.)

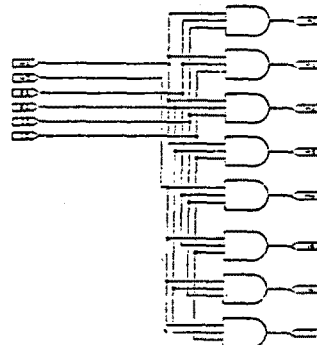
마찬가지로 $B^{(1)}, C^{(1)}$ 등이 이미 경험된 것이면 $B^{(0)}, C^{(0)}$ 등은 입력과 같고 $A^{(0)}$ 는 $B^{(1)}, C^{(1)}$ 등의 상태에 따라 결정이 된다. 이것은 내재되어 있는 물에 의하여 입력된 사실(Fact)에 따라 추론되는 것과 동일하다. 여기서 일반적으로 $A^{(0)}$ 와 $\bar{A}^{(0)}, B^{(0)}$ 와 $\bar{B}^{(0)}, C^{(0)}$ 와 $\bar{C}^{(0)}$ 는 서로 보수관계가 아니지만 ($A^{(1)}$ 등 입력도 마찬가지이다.), 편의상 민텀(Min-term)의 개념을 도입하여 위의 식들을 다시 써보면 다음과 같다.

경험된 민텀: m_0, m_1, m_3, m_7

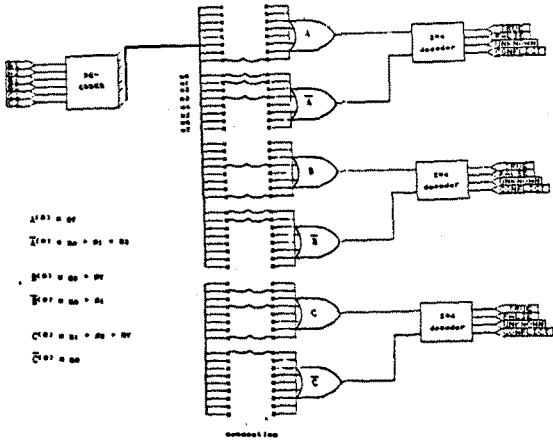
$$\begin{aligned}
 A^{(0)} &\Leftarrow m_7 && A \text{가 포함된 민텀 (Minterm)} \\
 \bar{A}^{(0)} &\Leftarrow m_0 + m_1 + m_3 && A \text{가 포함된 민텀 (Minterm)} \\
 B^{(0)} &\Leftarrow m_3 + m_7 && B \text{가 포함된 민텀 (Minterm)} \\
 \bar{B}^{(0)} &\Leftarrow m_0 + m_1 && B \text{가 포함된 민텀 (Minterm)} \\
 C^{(0)} &\Leftarrow m_1 + m_3 + m_7 && C \text{가 포함된 민텀 (Minterm)} \\
 \bar{C}^{(0)} &\Leftarrow m_0 && C \text{가 포함된 민텀 (Minterm)}
 \end{aligned}$$

((0) 은 출력임을 나타낸다.)

모든 것을 AND, OR, NOT으로 구현하였으므로 이에 따라 조합로직 회로로 구성할수 있다. 우선 민텀(Minterm)을 생성하는 디코더의 회로부분은 다음과 같다:

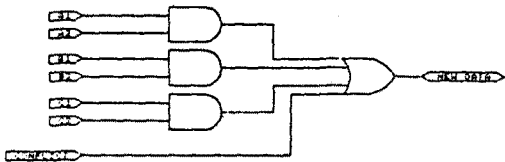


다음의 회로는 식을 로직으로 구성하고 두개의 비트로부터 참,거짓,미결정,모순의 상태를 알아내는 디코더 회로를 첨가한 회로이다:

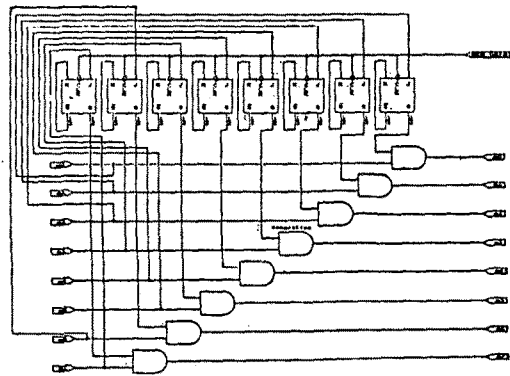


새로이 경험되는 데이터를 자동으로 저장하려면 이것을 감지하여야 한다:

앞에서 고찰한 바와 같이 입력 가운데 미결정(1,1)이 없고 출력 가운데 모순이 생긴다면 그때의 입력은 경험하지 못했던 패턴이다. 새로운 데이터 감지회로는 다음과 같다:



새로운 데이터 감지신호 따라 각 출력에 민첩의 연결을 추가하는 회로는 다음과 같다:



위와 같이 모두 로직회로로 지식 베이스를 구성하면 변수가 증가함에 따라 소자수가 지수적으로 증가한다:

3. 인공 신경 회로망 모델

앞에서 살펴본 알고리즘과 회로는 원리적으로 볼때, 경험되는 데이터를 패턴으로 보고 메모리에 모두 기억시킨 후에, 입력되는 데이터와 비교하여 가장 가까운 것을 출력하는 것이다. 이는 패턴인식 분야에서 많은 장점을 가지고 있는 인공 신경 회로망을 사용하여 구현될 수 있다. 그림 2-1의 예를 기존의 방법을 사용하여 인공 신경회로망으로 구현해 보면 다음과 같다:

1. 각 노드의 구조: Perceptron.
2. 학습 알고리즘: Back propagation.
3. 학습 입출력쌍:

입력	출력
A \bar{A} B \bar{B} C \bar{C}	A \bar{A} B \bar{B} C \bar{C}
0 0 0 0 0	0 0 0 0 0
0 0 0 0 1	0 0 0 0 1
0 0 1 1 1	0 0 0 0 1
1 1 0 0 0	0 0 0 0 0
⋮	⋮

4. Topology

그림 3 - 1

위의 모델은 학습시켜야 하는 입출력쌍들을 모두 구해내야 하므로 그 갯수가 많다. 또한 원하는 결과를 얻기 위한 중간 노드의 갯수를 정확히 알 수 없다. 연결강도의 학습시간이 길며 그 수렴을 보장할 수 없는 단점들이 있다. 이에 (2)의 원리를 바탕으로 한 새로운 모델의 인공 신경 회로망을 제시한다:

1. 각 노드의 구조: Perceptron.
2. 학습 알고리즘:

입력노드의 값: $I_1, I_1, I_2, I_2, \dots$
 출력노드의 값: $O_1, O_1, O_2, O_2, \dots$
 중간층의 노드: $H_1, H_1, H_2, H_2, \dots$
 연결 강도 : W_{ij}, W_{ij}

(1). 학습 데이터의 입력
 만약 $O_i, O_i = 0$ (all i) 이면 New Data
 노드를 증가 ($j = j + 1$)
 $W_{ij} = I_i, W_{ij} = I_i$ (all i)

(2). (1)을 반복

3. 학습 입출력쌍: 그림 3-1과 같다.
4. Topology :

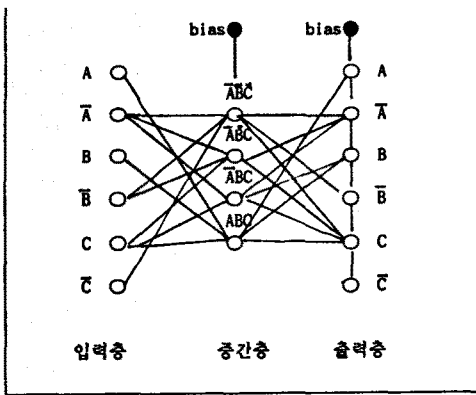


그림 3 - 2

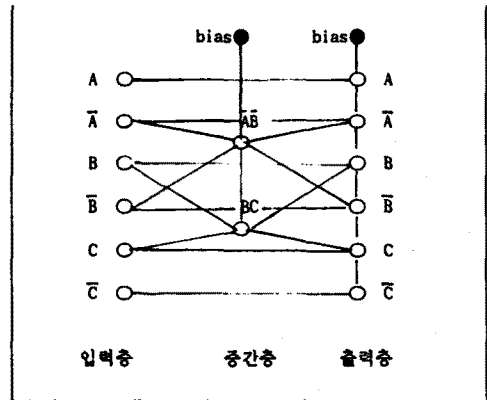
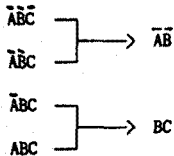


그림 3 - 3

위의 모델에서 중간층의 노드들은 경험되는 각각의 데이터들이다. 이들은 학습과정에서 생성되는데 변수가 N개 일때, 그 갯수는 2^n 보다 작은 수이다. 이는 (2)에서의 로직회로와 마찬가지로 변수가 들어남에 따라서 지수적으로 증가 한다. 그러나 중간 층의 한 노드가 하나의 데이터(패턴)를 나타내게 하지 않고 경험되는 패턴들을 축약해서 나타나는, AND로 묶이는 항으로 하면 중간층의 노드 갯수를 많이 줄일 수 있다. 즉 위의 예에서:

<경험된 데이터> <축약된 항>



중간 레이어가 4개에서 2개로 줄게 된다. 축약된 항은 경험되는 패턴들을 몇개의 그룹으로 나눌때 한 그룹의 특징을 갖는 대표 패턴으로 생각하면 된다. 이때 발생하는 문제는, 중간 노드가 특징이 아닌 변수의 출력을 결정할 수 없다는 것이다. 그러나 그러한 변수의 결과는 입력의 값과 같은 값을 가지므로 입력 노드와 출력 노드를 계산된 연결강도로 연결시켜 주면 문제점이 해결된다. 개선된 모델은 다음과 같다:

1. 각 노드의 구조: Perceptron.
2. 학습 알고리즘 :
 - 입력노드의 값: $I_1, I_1, I_2, I_2, \dots$
 - 출력노드의 값: $O_1, O_1, O_2, O_2, \dots$
 - 중간층의 노드: $H_1, H_1, H_2, H_2, \dots$
 - 연결 강도 : W_{ij}, W_{ij}
 - (1). 학습 데이터들을 축약한다.
 - (2). 학습 데이터의 입력 만약 $O_i, O_i = 0$ (all i) 이면 New Data 노드를 증가 ($j = j + 1$) $W_{ij} = I_i, W_{ij} = I_i$ (all i)
 - (3). (2)을 반복
3. 학습 입출력쌍 : 그림 3-1과 같다.
4. Topology :

4. 결 론

본 연구에서는 전문가 시스템의 지식베이스를, 기록된 데이터를 가지고 자동적으로 구성하는 알고리즘을 제시하였다. 결국 이는 경험되는 데이터를 하나의 패턴으로 보고 메모리에 효과적으로 기억시키는 방법으로써, 입력되는 데이터와 가장 가까운 것을 출력하게된다. 이러한 알고리즘을 패턴인식분야에서 많이 사용하는 인공 신경 회로망으로 구현하였다. 기존의 인공 신경 회로망이 가지고 있는 문제점을 제거하여 쉽게 연결 강도를 결정할 수 있고, 노드의 수를 절약 수 있는 새로운 모델을 제시하였다.

참고 문헌

- [1] B. G. Buchanan, E. H. Shortliffe, "Rule-based Expert Systems", Addison-Wesley, 1984.
- [2] D. D. Volfgam, T. J. Dear, C. S. Galbraith, "Expert Systems for the Technical Professional", John Wiley & Sons, 1987.
- [3] F. Hayes-Roth, D. A. Waterman, D. B. Lenat, "Building Expert Systems", Addison-Wesley, 1983.
- [4] Kosko, B., "Bidirectional Associative Memory", IEEE Trans. Man & Cibern., 1987.
- [5] Kosko, B., "Constructing an Associative Memory", Byte, 1987.
- [6] Gallant, Stephan I., "Connectionist Expert Systems", Communications of ACM, v31 n2, pp 152-170.
- [7] Galinat, Stephan I., "Automatic Generation of Expert Systems from Examples", Proc. of 2nd Intern. Conf. on Artificial Intelligence Applications, IEEE, 1985, pp313-319.
- [8] Gallant, Stephan I., "Matrix Controlled Expert Systems Produccible from Examples", U.S. patent pending 707, 458, 1985.
- [9] Richard P. Lippmann, "An Introduction to Computing with Neural Nets", 1987, 4.
- [10] Yoh-Han Pao, "Adaptive Pattern Recognition and Neural Nets", Addison-Wesley, 1989.