

신경회로망을 이용한 개념구조의 인식 및 처리에 관한 연구*

박상무 · 이수동
컴퓨터정보통신공학부

<요 약>

연결주의 지식표현을 위한 개념구조의 인식 및 처리에 있어서 신경망 모델들은 병렬처리로 인한 실시간 연산 수행과 오류 내인성(fault tolerance)에는 큰 장점이 있는 반면 주소 개념의 부재로 인하여 기호주의에 비해 변수의 바인딩, 구조적 표현, 지식 순계, 참조기능의 구현, 순차적 제어등의 어려움이 있다. 본 논문에서는 개념 그래프와 프레임으로 표현되는 개념구조를 학습하고 인식할 수 있는 신경회로망 모델인 개념 신경 회로망(CPNN : Conceptual Processing Neural Network) 모델을 제안하였다. 제안된 신경망은 개념 그래프로 표현된 단일 개념구조에 대한 처리와 그로 인해 야기되는 참조기능 및 변수의 바인딩 등의 문제를 망의 확장을 통해 프레임 구조와 같은 상이한 개념구조를 처리하는 과정을 도입함으로써 해결하고 있다. 따라서, 연결주의 지식표현이 지닌 다양한 개념구조의 처리에 대한 문제점을 해결하며, 개념 신경망이 다양한 지식을 습득하고 처리함에 있어서 융통성있는 인공지능 모델임을 보였다.

A Study on Conceptual Structure Recognition and Processing Using Neural Network

Sang-Moo Park · Soo-Dong Lee
Division of Computer Engineering & Information Technology

<Abstract>

Neural network models for connectionism knowledge representation have advantages of fault tolerance and real-time operation due to parallel processing but have difficulties

* 이 논문은 1997학년도 울산대학교 학술연구조성비에 의하여 연구되었음

in variable binding, structured representation, knowledge inheritance, implementation of reference and sequential processing owing to wickness of addressing concept. In this paper, we suggest a CPNN(Conceptual Processing Neural Network) model that is able to train and recognize a conceptual structure which is represented a conceptual graph and frame structure. For solving problems with a variable binding and implementation of reference in single conceptual structure, the proposed neural network model extends a network and accepts another conceptual structure as a frame structure. Therefore, we show that CPNN is adaptable AI model in knowledge acquirement and processing.

1. 서 론

인공 지능에 있어서 지식의 중요성이 점차로 증대됨에 따라 컴퓨터 내에 지식을 표현하는 원리와 방식등에 대한 연구가 매우 활발히 진행되어 왔다. 과거 인공지능 프로그램이나 시스템에서 제어를 어떻게 할 것인가 하는 문제는, 이제는 지식을 어떻게 표현하고 잘 사용해서 컴퓨터가 지적인 행동을 수행하도록 하는 가에 크게 의존하게 되었다.

지식 표현이라는 것은 단순히 지식 그 자체를 표현함을 의미하는 것이 아니라 표현된 지식에 대하여 어떻게 제어할 것인가 까지를 포함하고 있다. 즉 지식 표현이란 지식을 구성하는 각각의 개념구조와 그것을 해석하여 컴퓨터가 지적인 행위를 할 수 있게 하는 해석 과정의 결합이라고 말할 수 있다. 지식 표현을 위한 방식으로는 논리, 의미 네트워크(Semantic Network), 프레임(frame)이나 스크립트(script), 개념 그래프(Conceptual Graph)등이 가장 널리 쓰이는 방식들이다.

본 논문에서는 논리적인 도식 언어의 일종인 개념 그래프와 자료 구조에 의한 방식인 프레임임을 이용하여 지식을 표현하는 각각의 개념 구조를 생성하고 표현된 지식에 대한 제어는 인간 두뇌의 신경 생리학적 모델에 기초를 둔 정보처리 패라다임인 신경회로망을 이용함으로써 인간의 두뇌 활동이 보이는 동작과 유사성을 보이는 인공지능 모델을 제안하였다. 각각의 개념 구조들을 신경회로망 모델에 학습시킴으로써 개념 그래프와 프레임이 자연스럽게 지식으로 수용될 수 있음을 보였으며, 인식 과정을 통해 지적인 행위를 가능하게 하는 융통성 있는 인공 지능 모델임을 보였다.

2. 본 론

2.1 개념 구조의 생성

2.1.1 개념 그래프(Conceptual Graph)

근대 철학자이며 논리학자인 Peirce는 사람이 계산을 할 때 기호에 의한 것보다 도형에 의한 것이 더 효과적이라고 생각하여 논리를 도식으로 표현하는 존재그래프(Existential Graph)라는 도식언어를 창안하였다.^[1] 도식언어는 논리언어(eg. Predicate Calculus)보다 사람이 읽기

에 쉽고 자연어보다 컴퓨터가 처리하기에 편리한 형태로 문장의 의미를 표현할 수 있다. 개념 그래프는 플로우 차트, 유한상태 기계(Finite State Machine), 또는 페트리 넷처럼 도형을 사용하여 의미를 전달하는 일종의 논리적인 도식언어(graphical language)로써 특정분야만을 위한 지식표현언어가 아닌 보다 더 일반적인 지식표현언어로 사용가능하다.^[2] 따라서, 개념그래프는 언어학, 심리학, 철학등에 기초한 지식 표현 언어의 일종이며 그래프에서 개념노드(concept node)는 실체와 속성, 상태, 사건등을 표현하며, 관계노드(relation node)는 어떻게 개념들이 상호 연결되어 있는가를 나타낸다. 개념노드는 그래프 상에서 사각형으로 표시되며, 관계노드는 원으로 표시된다.^[3] 하나의 개념노드는 그 자체만으로 개념그래프를 형성하지만(그림 1.(a)), 관계노드는 그 자체만으로 개념그래프가 될 수 없다(그림 1.(b)). 따라서, 개념그래프안에 있는 모든 관계노드는 개념노드와 연결되어 있는데 이 때 연결을 위하여 방향성을 갖는 아크(화살표)를 사용하며(그림 1.(c)), 개념노드와 관계노드 사이에서만 존재가 가능하다. 따라서, 같은 종류의 노드끼리 아크를 사용하여 연결되면, 이는 개념그래프가 아니다(그림 1.(d)).

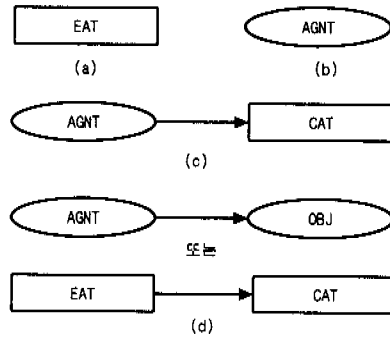


그림 1. 개념 그래프의 표현 양식

Fig. 1. Basic representations of conceptual graph

추상적 또는 사실적 실체(속성, 상태, 사건, 실체 등)를 나타내는 개념노드는 그래프상에서 사각형으로 표현되는데, 표기의 간결성을 위하여 그래프를 선형표현(linear form)으로 표현할 때는 대괄호 '[']를 사용한다. 개념노드간의 관계를 말해 주는 관계노드는 그래프상에서 원으로 표시되며, 선형표현에서는 소괄호 '()'를 쓴다. 대부분의 관계노드는 두 개의 개념노드간의 관계를 정의하는 이원(binary)관계를 나타내지만 때로는 일원(unary) 또는 다원의 관계를 나타내기도 한다. 다음 그래프는 '서울은 올림픽을 개최하였다.'를 나타내는데

(PAST) --> [서울] <-- (AGNT) <-- [개최하다] --> (OBJ) --> [올림픽]

이때 (PAST)는 일원관계를 나타내는 관계노드이고 그 밖의 관계노드 (AGNT), (OBJ)등은 이원관계를 나타내며 각 관계노드들은 화살표를 이용하여 관련된 개념노드에 연결된다.

주어진 이미지(image)를 이해하기 위한 개념 그래프를 예를 들어 보자. 매트 위에 고양이가 앉아서 우유를 먹고 있는 장면이 주어졌을 때 이러한 물리적 영역의 개념 공간(concept space)에서 있어야할 일반적인 개념들과 관계들은 표 1과 같다.

표 1. 개념 공간 속의 개념과 관계

Table 1. Concepts and Relations in concept space

Concept	CAT, SIT, MAT, EAT, MILK
Relation	STAT - state of LOC - location of AGNT - agent of OBJ - object

추출된 개념과 관계들을 사용하여 주어진 상황을 개념 그래프로 표현하면 그림 2와 같다.

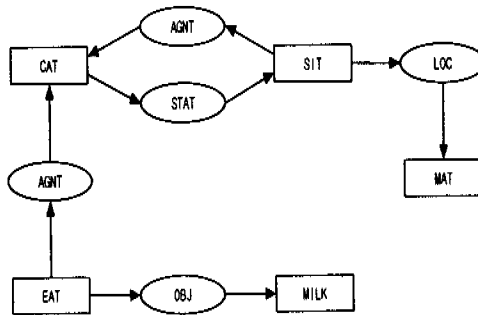


그림 2. 주어진 이미지의 개념 그래프
Fig. 2. Conceptual graph of input image

2.1.2 프레임 구조(Frame Structure)

1975년 M. I. T. 의 M. Minsky가 프레임 이론^[4]을 발표한 이후 프레임은 지식 표현의 수단으로는 가장 널리 쓰이는 것이 되었다. 프레임은 지식을 구성하는 효율적인 자료구조로 생각할 수 있다. 현재의 지식 표현 기법중 많은 것이 이 프레임 이론과 함께 발전해 왔으며 프레임에 바탕을 둔 많은 시스템이 개발되었다.

일반적으로 프레임은 일단의 슬롯(slot)-필러(filler)들로 구성되는데, 각 슬롯들은 프레임이 표현하는 개념 기술의 한 측면을 나타낸다. 이 슬롯들의 값을 이루는 필러는 전형적인 값을 갖거나, 특정 상황에 맞는 특정한 값을 나타낸다. 이러한 전형적 프레임이나 특정 상황에 맞는 값으로 채워진 프레임들을 바탕으로, 과거의 경험에서 얻어진 개념을 통해서 새로운 데이터를 이해하는 구조가 제공되는 것이다. 예를 들어, 차에 대한 일반적인 개념을 나타내는 프레임을 '자동차'라고 하면, '자동차'프레임에는 생산자, 모델, 색깔, 엔진의 용량, 구동 방식등의 슬롯들이 있게 된다. 그런데, 특정한 차에 대한 프레임의 경우, 예를 들어 '철수의 자동차'는 일반적인 자동차 프레임과 같은 슬롯들을 상속 받지만, 그 내용은 더 자세히 명시되게 된다. 즉 생산자는 '현대', 모델은 '엑셀', 색깔은 '상아색'등의 값을 가지게 되는 것이다. 이 과정을 실증화

(instantiation)라고 하며, 차의 전형적 개념인 ‘자동차’와의 비교에 의해 수행되는 것이다. 이와 같이 프레임이 전형적인 개념을 나타내고, 비교에 의해 특정한 개념을 표현할 수 있기 때문에, 프레임은 심리학에서 발전된 원형(prototype) 이론을 쉽게 반영할 수 있다. 즉 일반적인 프레임은 바로 객체의 원형을 나타내며, 실증화는 이 원형과의 비교에 의해 이루어지는 것이다. 이것이 프레임 이론이 각광을 받는 가장 큰 장점 중의 하나이다. 왜냐하면, 원형 이론은 현재 인간의 심리적 행위의 가장 근본적인 형태를 나타낸다는 것으로 인정받고 있으며, 프레임이 이 이론을 뒷받침한다는 것은 바로 프레임이 인간의 정보처리에 보다 적합한 지식 표현임을 나타내는 것이기 때문이다. 이와 같이 원형을 나타내는 프레임의 슬롯값들은 특정한 값 뿐만 아니라, 디폴트(default) 값을 갖는 경우가 많다. 실제로 원형은 거의 디폴트값으로 구성된다고 볼 수 있다. 그러나 이러한 디폴트값은 슬롯에 약하게 부착되어 있어서 현재 상황에서 디폴트값이 모순이 되면, 그 값은 입력 정보에 의해 대치된다. 프레임 역시 의미 네트워크(semantic network)^[5]와 마찬가지로 상속이 가능하다. 즉 일반적인 프레임에서 산출된 모든 후속 프레임은 상위 프레임이 갖는 성질이나 값을 상속받을 수 있다.

프레임이 갖는 두 번째 장점은 프레임에 바탕을 둔 많은 시스템들이 여러 가지의 응용 분야에서 매우 효율적으로 이용될 수 있음이 증명된 것이다. 예를 들어, 자연어 처리, 질의 응답, 예정화 등에 있어서 프레임은, 인간 정보 처리의 심리적 과정을 뒷받침하기 때문에 아주 효과적임이 증명되었다.

그러나, 이러한 연구들은 모두 물리적 기호나 배경 이미지를 컴퓨터 내부에서 기호나 이미지의 의미와 무관한 부호(code)로 바꾸어서 지식을 표현하기 때문에, 기호 이미지 자체가 보유하고 있고 어느 경우에는 상당히 중요한 변수가 될 수 있는 어떤 특질을 간과하게 되어, 전반적인 이해에 있어 오류를 범할 수 있다는 약점이 따르게 된다는 문제들을 가지고 있다. 예를 들어 실내의 장면을 인식하게 위해서 프레임용 이용하여 표현하면 그림 4와 같다. 그림에서 각각의 프레임, 슬롯, 필러에 사용되는 이름들은 인간의 편의에 의해 할당되어진 것일 뿐 이것들이 컴퓨터 내부에서 어떤 의미를 갖는 것은 아니며 그저 하나의 부호(e.g. ASCII code)로 표시되는 문자열일 따름이다. 따라서 지식 표현시 응용 시스템에서 원하는 모든 기호간의 관계를 명백히 표현해 주어야 하며 우리가 표현하지 않은 관계를 필요로 할 때 시스템은 잘못 동작하게 되는 현상이 일어나게 된다.

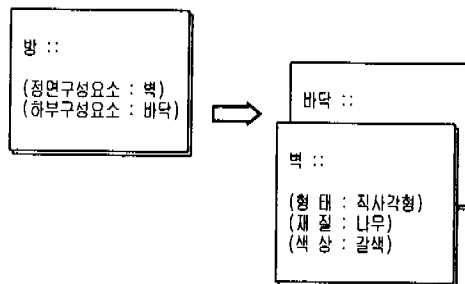


그림 4. 프레임을 이용한 지식표현

Fig. 4. Knowledge representation using frame

2.2 개념 신경회로망 모델(CPNN : Conceptual Processing Neural Network)

2.2.1 CPNN의 구조 및 동작특성

순차적인 시퀀스로 주어지는 개념과 각각의 개념들 사이의 관계들을 학습시키기 위하여 제안된 개념 신경망은 입력층과 은닉층 사이에 두 개념의 관계를 규정하는 관계층을 추가하여, 학습시와 인식시에 목적 개념을 발생 시키기 위한 관계를 인가하는 역할을 하도록 한다. 관계층은 순차적인 시퀀스로 작용하는 입력층과는 달리, 원하는 개념을 발생시키는 동안은 고정된 값을 가진다.

은닉층으로부터 내부 상태층에 이르는 귀환루프(feedback loop)와 문맥 출력층(context output layer)으로부터 은닉층에 이르는 귀환루프를 두어 회로망이 강력한 회귀구조를 가짐으로써 시계열적 정보를 강하게 흡수하고, 계획된 연속 순차를 발생시킬 수 있도록 하였다.

회로망의 결합형태는 입력층과 은닉층간과 내부상태층과 은닉층간에는 임의의 부분결합형태를 이루고 은닉층과 출력층간에는 완전 결합형태를 이룬다(그림5 참조). 여기서 은닉층의 처리요소는 입력층과 내부상태층의 처리요소로부터 무작위로 균일하게 선택된다. 이렇게 회로망이 부분결합형태를 가짐으로써 계통(network dimension)의 용장도(redundancy)가 감소되어 완전 결합망에 비해 빠르게 안정상태로 수렴될 수 있다.^[6]

이 회로망의 각 처리요소(processing unit)의 상태는 매 시간스텝에서 식 (1)로 표현된다.

$$X_j = f \left(\sum_{i=1}^n W_{ji} X_i + \theta_j \right) \quad (1)$$

- X_j : 처리요소 j의 출력값
- W_{ji} : 처리요소 i에서 j로 가는 연결강도(weight)
- θ_j : 처리요소 j의 바이어스(bias)
- N : 처리요소 j의 입력으로 결합된 처리요소들의 총수

함수 f는 처리요소의 활성화 함수이고 은닉층과 출력층의 함수로는 시그모이드(sigmoid) 함수를 사용하였다.

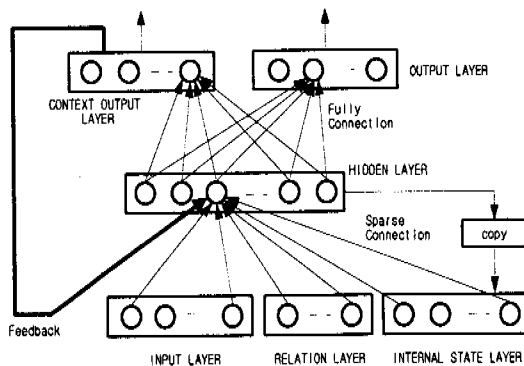


그림 5. 개념 신경망 모델

Fig. 5. A Conceptual Processing Neural Network

2.2.2 학습

개념 신경망의 학습과정에 사용된 학습 알고리즘은 Rumelhart, Hinton, 그리고 Williams에 의해 개발된 오류역전파(EBP) 학습 알고리즘^[7]이 이용되었다. 오류역전파 알고리즘은 다층 전향 신경회로망(Multi-Layer Feedforward Neural Network)의 학습 알고리즘으로 개발 되었으나 회귀 신경망에도 이를 또한 적용할 수 있다. 그러나 회귀망에서 역전파 과정이 적절하게 진행되기 위해서는 회귀루프를 가지는 처리요소가 이전 시간 스텝에서의 활성값을 보존해야 한다. 개념 신경망에서는 은닉노드와 출력노드의 활성값이 내부상태층과 문맥 출력층에 각각 보존됨으로써 역전파 과정이 정상적으로 진행되게 된다.

노드의 활성화 함수(activation function)로는 0에서 1사이의 출력값을 갖는 시그모이드함수를 사용했으며 학습속도를 높이기 위해서 관성항(momentum) α 를 사용했다. 순차적인 시퀀스로 주어지는 개념과 각각의 개념들 사이의 관계들을 학습시키기 위해서 제안된 신경회로망 모델은 임의의 지식에 대한 구조적 표현으로 개념 그래프를 이용하여 지식을 신경망 내부에 수용시켜 원인개념과 관계코드에 의해 결과개념이 도출되게 된다. 그러나, 지식은 개개의 인스턴스(instance)로 표현되는 것이 아니라 형(type)을 표시할 수 있는 변수로 표현된다. 따라서, 변수를 특정 인스턴스로 대치하는 변수 바인딩(binding)이 필요하다. 구조적으로 표현된 각각의 개념들 사이에 변수 바인딩을 구현하기 위해서는 한 구조를 통해서 다른 구조에 접근할 수 있는 참조기능을 가져야 하는데, 기호주의에서는 포인터에 의한 참조가 자연스럽지만, 연결주의에서는 주소개념이 없기 때문에 참조의 구현이 어렵다.

그림 6에서와 같이 동일한 두 개의 신경회로망을 사용하여 개념 그래프 처리부와 프레임 구조 처리부를 두어 학습시에는 개별적으로 동작하여 개념 그래프와 프레임 구조를 신경망 내부에 수용시키게 된다. 개념 그래프 처리부는 개념 그래프로 표현된 지식들을 원인개념과 관계코드를 입력으로 결과개념을 학습시키는 반면, 프레임 구조 처리부는 각각의 프레임과 그에 따른 슬롯(slot), 필러(filler)들을 관계로 학습시키게 된다. 그리고, 인식시에는 개념 그래프 처리부의 출력을 프레임 구조 처리부의 입력층으로 인가시켰다. 프레임 구조 처리부의 입력층을 개념 그래프 처리부의 출력층과 동일한 크기 및 형식으로 구성함으로써 개념 그래프 처리부의 결과개념이 프레임 구조 처리부의 결과개념과 바인딩이 가능하게 됨으로서 참조기능 및 변수의 바인딩등의 문제를 처리할 수 있다.

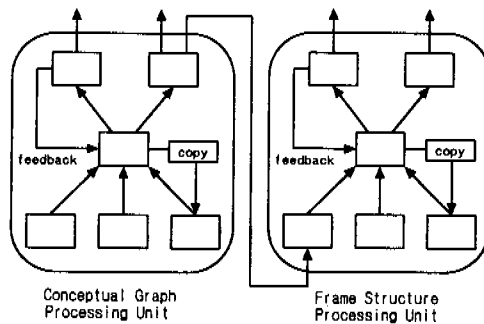


그림 6. 확장된 개념 신경망 모델

Fig. 6. A Extended Conceptual Neural Network

5. 시뮬레이션 및 결과

5.1 시스템 모델의 구성

본 논문에서 개념 그래프 등의 단일 개념 구조를 학습 시키기 위한 시스템 모델은 그림 7과 같다. 입력층에 개념들의 시퀀스와 관계들을 순차적으로 획득하고 입력하기 위해서 개념 습득 오토마타를 연결하여 입력부를 구성하였고, 학습과 인식 단계를 설정하기 위한 플래그(flag)를 두었고 이 플래그에 의해 학습시 교육패턴이 인가된다.

그림 8은 개념 그래프와 함께 프레임 구조를 학습시키기 위한 다중 개념 시스템 모델로의 확장을 보여주고 있다. 개념 그래프 처리부의 입력부에 인과될 시퀀스로는 9*9로 구성된 문자 이미지를 사용하였고, 출력으로는 한글일 경우 2바이트 조합형으로, 영문일 경우 각각의 문자를 나타내는 26개의 출력과 공백문자를 표시하는 1개의 출력을 더해 27개의 출력라인으로 구성하였다. 또한, 프레임 처리부의 입력부에 인가될 시퀀스로는 개념 그래프 처리부의 출력 형태로 인해 9*3의 이미지가 사용되며 출력부는 개념그래프 처리부와 동일하다. 그리고 관계코드는 미리 정의한 관계코드를 사용하였다.

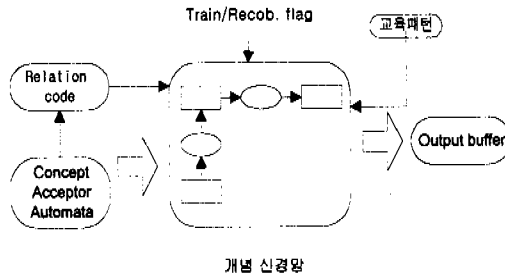


그림 7. 개념 그래프 인식 시스템
Fig 7. Conceptual Graph Recognition System

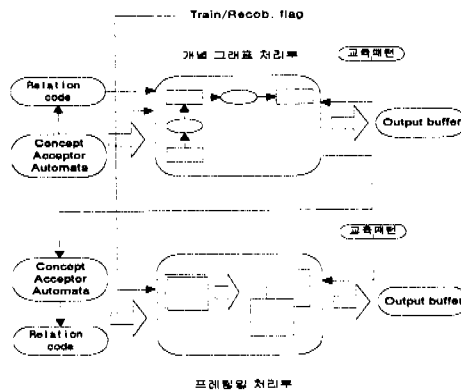


그림 8. 다중 개념 인식 시스템
Fig. 8. Multiple Concept Recognition System

본 논문에서 추구하는 목적에 부합하는 매개변수(parameter)들은 반복적인 실험을 통해서 학습시의 중요한 변수인 학습률은 0.1로 관성항(momentum)은 0.6로 설정하였다.

표 3은 본 논문에 적용된 신경망의 계통(network dimension)의 크기를 보여주고 있다.

표 3. 복합 개념 신경망의 계통크기
Table 3. Network dimension of MCNN

* 개념 그래프 처리부

입력층	은닉층	관계층
81/9	80/x	4/4

출력층	문맥 출력층	내부 상태층
27/27	27/27	80/60

* 프레임 처리부

입력층	은닉층	관계층
27/9	80/x	4/4

출력층	문맥 출력층	내부 상태층
27/27	27/27	80/60

A/B : A는 각 층에 있는 처리요소의 수를 나타내고, B는 은닉층에 결합된 각 층의 처리요소의 수를 나타낸다.

x : 없음(none)

5.2 관계 코드(relation code)의 설정

본 논문에서는 주어진 문제 영역(problem domain) 내에서 만날 수 있는 가능한 기본 개념들과 이들 사이의 관계에 관한 단어들을 미리 추출하고, 이들을 순차 처리가 가능한 개념 신경망으로 구성한 개념 그래프 처리부와 프레임 처리부에 학습에 의해 흡수 수용시켜 개념 인식 및 추출의 기본 모델로 삼았다.

프레임 구조가 프레임 처리부에 학습, 수용되기 위한 기본 개념 및 관계의 추출은 주어진 문제 영역내에서 만날 수 있는 프레임 구조의 형태를 살펴 봄으로 인해 가능하다. 프레임 구조는 하나의 프레임 아래 슬롯-필러의 구조로 된 여러개의 요소를 가질 수 있으며, 각각의 슬롯과 필러는 또 다른 프레임을 나타낼 수도 있도록 되어 있다. 그러므로, 하나의 프레임 구조에서 나타낼 수 있는 관계는 프레임과 슬롯과의 관계, 슬롯과 필러와의 관계, 프레임과 필러와의 관계로 나타낼 수가 있다.

표 4에 문장 역할에 따른 기본 개념들의 분류와 주로 조사와 동사 또는 형용사의 어미로부터 추출 가능한 개념들 사이의 관계와 프레임 구조에서 나타낼 수 있는 관계들을 보였다.

표 4. 관계코드 정의표

Table 4. Relation Code Definition Table

Conceptual Relation	Relation Code
AGNT : agent-of	0
IS : is-a	1
CHRC : characteristic-of	2
OBJ : object-of	3
POSS : possession-of	4
SLOT : frame-slot	5
INST : frame-filler	6
FILL : slot-filler	7

5.3 개념 그래프와 프레임의 학습 및 인식

5.3.1 학습

다음에 주어진 것과 같이 문장으로 표시된 지식들을 학습시켜 수용시키기 위해서는 먼저 각 문장에서 기본이 되는 개념을 추출하고 각 개념들 사이의 관계를 추출하여야 한다.

“대전은 도시이다.”(a)

“엑스포는 과학기술을 전시하는 박람회이다.”(b)

“과기원은 사물로봇을 전시하였다.”(c)

주어진 문장에서 추출된 기본 개념들과 기본 개념들간의 관계들을 이용하여 관계코드표를 구성하고(표 5), 그림 7과 같은 기본적인 개념그래프를 얻었다.

표 5. 기본 개념들간의 관계코드표

Table 5. Relation Code between Basic Concepts

번호	개념 원인	결과 개념	관계	관계 코드
1	대전	도시	is-a	1
2	엑스포	박람회	is-a	1
3	박람회	전시하다	characteristic	2
4	전시하다	과학기술	object	3
5	전시하다	과기원	agent	0
6	전시하다	사물로봇	object	3

이와 같은 기본적인 개념 그래프들을 신경망 내부에 수용시키기 위해 개념간의 관계코드를 관계층에 인가시킨후 원인개념을 나타내는 시퀀스를 순차적으로 입력층에 인가하고 출력층에

는 결과개념의 시퀀스를 보여 주면서 모든 개념 관계들을 반복해서 학습시킨다.

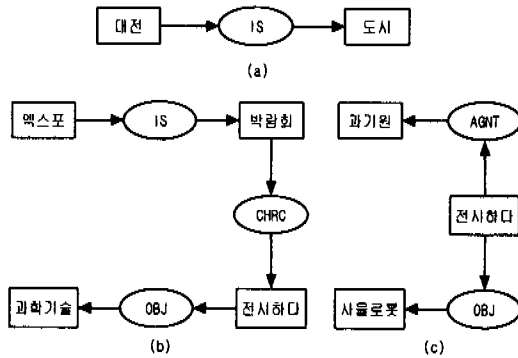


그림 9. 추출된 기본적인 개념 그래프
 Fig. 9. Extracted basic conceptual graph

그러나 그림 9의 기본적인 개념 그래프에서 원인개념 “전시하다”와 관계 “OBJ”에 대한 결과개념으로서 “과학기술”과 “사물로봇”이라는 두 개의 개념이 학습되어 지기 때문에 프레임 구조를 도입하여 두 개념을 바인딩시켜 주어야만 한다. 그림 8와 같은 프레임 구조를 도입함으로써 원인개념 “전시하다”와 관계 “OBJ”에 의해 결과개념으로 “과학기술”을 출력하게 되고, 관계 “AGNT”에 의해 전시하는 주체는 “과기원”라는 것을 알 수 있게 되며, 변수 “과학기술”에는 “사물로봇”이 인스턴스로 출력이 된다. 이와 같이 프레임을 사용함으로써 연결주의에서 바인딩이 가능하게 할 뿐만 아니라 기존의 프레임과 같은 슬롯-필러 구조로 된 자료들을 큰 수정 없이 사용할 수도 있다는 것을 보여준다.

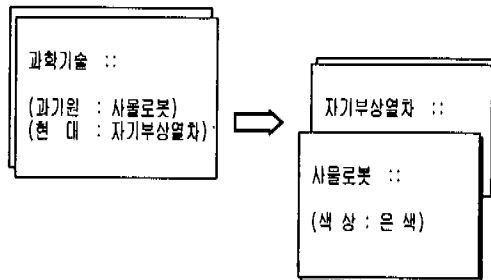


그림 10. 적용된 프레임 구조
 Fig. 10. Applied frame structure

표 6은 기본 개념들간의 관계코드표에 프레임 구조를 도입한 수정된 관계코드표를 나타내며, 그림 11는 신경망 내부에 학습되어 수용되어진 개념 그래프와 프레임 구조를 논리적으로 도식화 한 것이다.

표 6. 복합 개념들간의 관계코드표
Table 6. Relation Code between Hybrid Concep

번호	개념 원인	결과 개념	관계	관계 코드
1	대전	도시	is-a	1
2	엑스포	박람회	is-a	1
3	박람회	전시하다	characteristic	2
4	전시하다	과학기술	object	3
5	전시하다	과기원	agent	0
6	과학기술	과기원	frame-slot	5
7	과학기술	사물로봇	frame-filler	6
8	과기원	사물로봇	slot-filler	7
9	사물로봇	색상	frame-slot	5
10	색상	은색	slot-filler	7

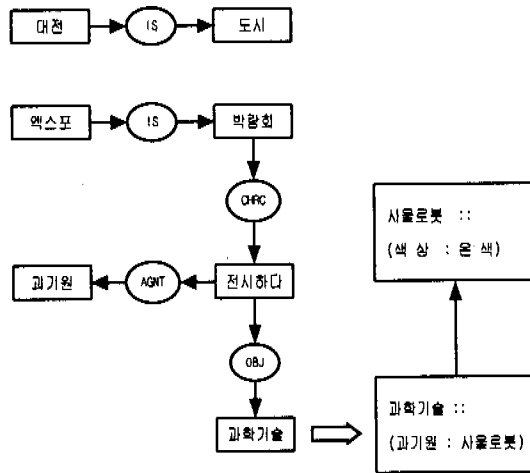


그림 11. 논리적으로 도식화된 개념 그래프와 프레임.
Fig. 11. Logical diagram of conceptual graph and frame.

5.3.2 인식

앞절에서 논리적으로 도식화된 개념 그래프와 프레임들이 실제로 신경망 속에 올바르게 학습되어 수용되었는지를 평가하기 위해서 학습시에 사용되었던 개념과 관계들을 입력으로 사용하여 출력을 살펴보았다.

표 7은 각각의 개념 인식시의 활성값을 나타낸다. 도표에서 출력노드의 활성값은 실제로 출력 노드에서 발생한 값들이고, 우측의 2바이트 조합형 한글 코드들은 임계치(threshold)를 0.9

로 설정해서 출력노드의 활성화들로부터 얻은 결과이다.

표 7의 (d)와 (e)의 결과로부터 원인개념 “전시하다”와 관계 “OBJ”에 의한 출력으로 “과학기술”을 출력하게 되고, 관계 “AGNT”에 의해 전시하는 주체는 “과기원”이라는 것을 알 수가 있으며, “과학기술”에는 “사물로봇”이 인스턴스로 출력이 되어 연결주의에서 다중 개념처리가 가능하게 되었음을 알 수가 있다.

6. 결 론

본 논문에서는 지식 표현에 있어서 연결주의의 장점에 기인한 신경망 모델을 제안함에 있어서 가지는 문제점인 변수의 바인딩, 구조적 표현, 참조기능의 구현, 순차적 제어의 어려움 등을 극복함으로써 자연어 처리(nat-ural language processing)나 전문가 시스템(expert system), 지능로봇, 생산 공정의 무인화 등의 구현시에 지능을 부여하기 위한 지식 베이스(knowledge base)를 생성 시킬 수 있는 가능성을 보여 주었다.

구조적 표현의 문제는 복합 개념 신경망이 개념 그래프와 프레임 구조등을 내부에 수용하여 인식시에 원하는 결과를 도출할 수 있음으로써 극복할 수 있었으며, 다중 개념처리에 있어 변수의 바인딩과 참조기능의 구현 문제는 기호주의의 프레임 구조를 도입하여 개념 그래프로 표현된 각각의 개념들이 학습된 프레임과 연계되어 결과를 나타내게 함으로써 극복하였다. 그러나 기호주의 지식 표현의 지식 승계와 다양한 지식 표현 양식들을 효율적으로 신경망에 이식시켜 학습시키는 문제 등은 아직도 많은 어려움이 있으므로 이를 해결하기 위해서는 전문가 시스템에 다양한 신경망들을 접목시키는 연구가 이루어져야 하겠다.

표 7. 출력노드의 활성화값과 인식된 한글코드

Table 7. Activation value of output node and recognized code

사이클 Cycle	출력노드의 활성화값				2바이트 조합형 한글코드
	1	2	3	4	
1	0.9625	0.0013	0.0069	0.9736	1001
	0.0016	0.9618	0.0081	0.9479	0101
	0.9453	0.0198	0.9769	0.0012	1010
	0.0003	0.0012	0.0076	0.0002	0000
2	0.9850	0.0040	0.9548	0.0341	1010
	0.9568	0.9802	0.9618	0.9786	1111
	0.9730	0.0173	0.9817	0.0000	1010
	0.0035	0.0002	0.0092	0.0005	0000

a) 원인개념 “대전”에 대한 인식결과 (관계코드 : IS) : “도시”

사이클 Cycle	출력노드의 활성값				2바이트 조합형 한글코드
	1	2	3	4	
	5	6	7	8	
	9	10	11	12	
	13	14	15	16	
1	0.9757	0.0031	0.9704	0.0247	1010
	0.0075	0.9621	0.0039	0.0001	0100
	0.0018	0.9793	0.9631	0.0095	0110
	0.0046	0.0016	0.9694	0.0180	0010
2	1.0000	0.0034	0.0266	0.9814	1001
	0.9691	0.9999	0.0165	0.0106	1100
	0.0098	0.9998	1.0000	0.9753	0111
	0.0006	0.0024	0.0258	0.9610	0001
3	0.9813	0.9604	0.0004	0.9681	1101
	0.0329	0.0266	0.9498	0.0004	0010
	0.0002	0.9904	0.0364	0.0220	0100
	0.0016	0.0021	0.0028	0.0233	0000

b) 원인개념 “엑스포”에 대한 인식결과 (관계코드 : IS) : “박람회”

사이클 Cycle	출력노드의 활성값				2바이트 조합형 한글코드
	1	2	3	4	
	5	6	7	8	
	9	10	11	12	
	13	14	15	16	
1	0.9728	0.0073	0.9714	0.9670	1011
	0.9585	0.0115	0.0001	0.0010	1000
	0.9668	0.9885	0.9677	0.0044	1110
	0.0010	0.9733	0.0142	0.9735	0101
2	1.0000	0.0076	0.9920	0.0358	1010
	0.9983	0.9979	0.9634	0.9789	1111
	0.9992	0.0229	1.0000	0.0055	1010
	0.0200	0.0181	0.0013	0.0363	0000
3	0.9993	0.9617	0.0064	0.9798	1101
	0.0141	0.0354	0.0395	0.0200	0000
	0.0245	0.9669	0.9949	0.0023	0110
	0.0004	0.0052	0.0003	0.0001	0000
4	0.9936	0.0411	0.0013	0.9717	1001
	0.0002	0.9538	0.0029	0.0003	0100
	0.0098	0.9724	0.9642	0.0021	0110
	0.0041	0.0020	0.0004	0.0007	0000

c) 원인개념 “박람회”에 대한 인식결과 (관계코드 : CHRC) : “전시하다

사이클 Cycle	출력노드의 활성값				2바이트 조합형 한글코드
	1	2	3	4	
	5	6	7	8	
	9	10	11	12	
	13	14	15	16	
1	0.9653	0.0149	0.0010	0.0020	1000
	0.9611	0.0001	0.0004	0.9614	1001
	0.9478	0.9919	0.0014	0.0010	1100
	0.0057	0.0001	0.0285	0.0000	0000
2	0.9995	0.9579	0.0020	0.9588	1101
	0.0438	0.0015	0.0395	0.0516	0000
	0.0590	0.9523	0.9970	0.0020	0110
	0.0023	0.0022	0.9540	0.0013	0010
3	0.9998	0.0323	0.0210	0.0452	1000
	0.9475	0.0345	0.9734	0.9417	1011
	0.9355	0.0379	0.9931	0.0018	1010
	0.0091	0.0069	0.0428	0.0160	0000
4	0.9846	0.0001	0.9658	0.0001	1010
	0.9724	0.9722	0.9988	0.0255	1110
	0.9977	0.0000	0.0164	0.0001	1000
	0.9649	0.0252	0.0001	0.9676	1001
5	0.9844	0.0019	0.9697	0.0001	1010
	0.9782	0.9648	0.0003	0.0511	1100
	0.0415	0.9773	0.9680	0.0041	0110
	0.0102	0.0043	0.0085	0.0075	0000
6	0.9975	0.0027	0.9927	0.0099	1010
	0.0240	0.0100	0.9435	0.0035	0010
	0.9424	0.0227	0.0514	0.0078	1000
	0.9549	0.0140	0.0042	0.9397	1001
7	0.9994	0.0005	0.0278	0.9602	1001
	0.9516	0.9634	0.0204	0.9778	1101
	1.0000	0.0000	0.9929	0.0245	1010
	0.0259	0.0164	0.0046	0.0580	0000
8	0.9914	0.0007	0.9576	0.0340	1010
	0.0278	0.9972	0.0000	0.9980	0101
	0.9937	0.0000	0.9997	0.9776	1011
	0.0000	0.9617	0.0001	0.9521	0101

d) 원인개념 "전시하다"에 대한 인식결과 관계코드 : OBJ, INST) : "과학기술", "사물로봇"

사이클 Cycle	출력노드의 활성값				2바이트 조합형 한글코드
	1	2	3	4	
	5	6	7	8	
	9	10	11	12	
	13	14	15	16	
1	0.9791	0.0002	0.0100	0.0004	1000
	0.9851	0.0027	0.0010	0.9709	1001
	0.9988	0.9681	0.0061	0.0002	1100
	0.0011	0.0003	0.0008	0.0003	0000
2	1.0000	0.0187	0.0284	0.0192	1000
	0.9961	0.0195	0.9650	0.9901	1011
	0.9983	0.0270	0.9995	0.0006	1010
	0.0029	0.0044	0.0050	0.0060	0000
3	0.9975	0.0010	0.9784	0.9541	1011
	0.0343	0.9679	0.9959	0.0213	0110
	0.9991	0.0003	0.9845	0.0000	1010
	0.0127	0.9611	0.0003	0.9680	0101
4	0.9642	0.0013	0.9605	0.0000	1010
	0.9346	0.9097	0.0140	0.0053	1100
	0.0396	0.9657	0.9265	0.0035	0110
	0.0328	0.0013	0.0487	0.0580	0000
5	0.9998	0.0047	0.9980	0.0530	1010
	0.0428	0.1049	0.9958	0.0660	0010
	0.9979	0.0486	0.0400	0.0013	1000
	0.9593	0.0327	0.0099	0.9601	1001
6	0.9991	0.0005	0.0456	0.9581	1001
	0.9718	0.9608	0.0212	0.9354	1101
	1.0000	0.0000	0.9524	0.0137	1010
	0.0098	0.0183	0.0434	0.0337	0000
7	0.9564	0.0009	0.9597	0.0172	1010
	0.0032	0.9942	0.0000	0.9862	0101
	0.9496	0.0000	0.9991	0.9803	1011
	0.0000	0.9791	0.0001	0.9719	0101

e) 원인개념 "전시하다"에 대한 인식결과 (관계코드 : AGNT, FILL) : "과기원", "사물로봇"

참 고 문 헌

- [1] Peirce. C. S., "Manuscripts on Existential Graphs", Mouton, The Hague, 1973.
- [2] 양기철, "개념그래프의 소개", 정보과학회지 제12권 제9호, 1994. 10.
- [3] J. F. Sowa, "Conceptual Structures", Addison-Wesley Publishing Company, 1984.
- [4] M. Minsky, "A Framework for Representing Knowledge", In P. H. Winston(Ed.) The Psychology of Computer Vision NY : McGraw-Hill, 1975.
- [5] Quillian, M. R., "Semantic memory", in Semantic Information Processing, M Minsky(ed.), The M. I. T Press, Cambridge, MA, 1968.

- [6] 김성석, "순차구조처리를 위한 부분결합 회귀 신경회로망", 공학박사 학위논문, 울산대학교, 1990, 8.
- [7] D. E. Rumelhart and J. L. McClelland, "Parallel Distributed Processing", vol.1, MIT press, 1986.

부 록 A. 2바이트 상용 조합형 한글 코드

상용 조합 한글 코드							
비트	초성	중성	종성	비트	초성	중성	종성
43210	·	·	·	43210	·	·	·
00000	·	·	·	10000	ㅈ	·	ㄹㅎ
00001	Fill	·	Fill	10001	ㅋ	·	ㅍ
00010	ㄱ	Fill	ㄱ	10010	ㅌ	ㅍ	·
00011	ㄲ	ㅏ	ㄱ	10011	ㅊ	ㅍ	ㅍ
00100	ㄴ	ㅑ	ㄱㅏ	10100	ㅎ	ㅌ	ㅍㅏ
00101	ㄷ	ㅓ	ㄴ	10101	·	ㄱ	ㅏ
00110	ㄸ	ㅕ	ㄴㅓ	10110	·	ㅑ	ㅏ
00111	ㄹ	ㅗ	ㄴㅎ	10111	·	ㄱ	ㅇ
01000	ㅍ	·	ㄷ	11000	·	·	ㅓ
01001	ㅑ	·	ㄹ	11001	·	·	ㅈ
01010	ㅑ	ㅑ	ㄹㄱ	11010	·	ㅑ	ㅋ
01011	ㅏ	ㅑ	ㄹㅑ	11011	·	·	ㅌ
01100	ㅏ	ㅑ	ㄹㅑ	11100	·	ㄱ	ㅑ
01101	ㅇ	ㅑ	ㄹㅏ	11101	·	ㅑ	ㅎ
01110	ㅓ	ㅑ	ㄹㅓ	11110	·	·	·
01111	ㅓ	ㅑ	ㄹㅑ	11111	·	·	·