

신경망을 이용한 퍼지 규칙 생성과 추론망 구축

Fuzzy Rule Generation and Building Inference Network using Neural Networks

이 상 령*, 이 현 숙**, 오 경 환***

Sang Ryung Lee*, Hyun Sook Rhee**, Kyung Whan Oh***

※본 논문은 1995년도 서강대학교 학술연구비에 의하여 연구되었음.

요 약

퍼지 시스템의 구현에 있어 가장 어려운 과정 중 하나는 정확한 지식의 획득이다. 이는 퍼지 시스템의 응용 영역이 커지고 그 응용 영역의 입출력 변수가 많아질수록 전문가가 그 변수들 간의 관계를 정확히 파악하는 것이 어렵고, 더구나 복잡한 시스템의 제어 과정을 언어 변수를 표현하는 것이 전문가에게 힘든 일이기 때문이다. 또 하나의 어려운 과정은 퍼지 변수의 적절한 소속함수 정의와 조정이다. 그래서 기존의 언어 변수를 포함하는 퍼지 규칙을 사용하여 설계된 퍼지 시스템에서는 기술된 퍼지 규칙들이 시스템의 특성을 제대로 반영하도록 퍼지 변수의 소속 함수 모양을 조정하는 작업을 필요로 한다.

본 논문에서는 기존의 퍼지 시스템 구현에 있어서 어려운 과정인 지식 획득과 소속 함수 정의부분의 개선을 위한 새로운 퍼지 시스템 구현 방법으로 코호넨 신경망과 역전파 신경망을 이용한 퍼지 규칙 자동 생성과 추론망 구축 방법을 제안한다. 제안된 방법은 시스템의 입력과 출력으로 구성된 데이터들로부터 퍼지 규칙을 신경망의 학습 기능을 이용하여 자동 생성한다. 또한 데이터 변수간의 퍼지 관계에 기반을 두고 추론이 이루어지므로 각 퍼지 변수에 대한 소속 함수 정의가 필요 없게 된다. 따라서 퍼지 시스템의 구현이 쉽게 이루어질 수 있다.

실험에서는 제안된 방법으로 자동차 정속 주행을 위한 추론망을 구축하고 실험차의 단독 주행의 여러 상황을 고려한 도의 주행 실험을 통해 새로운 방법의 타당성을 보인다.

ABSTRACT

Knowledge acquisition is one of the most difficult problems in designing fuzzy systems. As application domains of fuzzy systems become larger and more complex, it is more difficult to find the relations among the system's input-output variables. Moreover, it takes a lot of efforts to formulate expert's knowledge about complex systems' control actions by linguistic variables. Another difficulty is to define and adjust membership functions properly. So

*신세기통신(주) 연구원

**동양공업전문대학 전산경영기술공학부 조교수

***서강대학교 전자계산학과 교수

in conventional fuzzy systems, the membership functions should be adjusted to improve the system performance. This is time-consuming process.

In this paper, we suggest a new approach to design a fuzzy system. We design a fuzzy system using two neural networks, Kohonen neural network and backpropagation neural network, which generate fuzzy rules automatically and construct inference network. Since fuzzy inference is performed based on fuzzy relation in this approach, we don't need the membership functions of each variable. Therefore it is unnecessary to define and adjust membership functions and we can get fuzzy rules automatically. The design process of fuzzy system becomes simple.

The proposed approach is applied to a simulated automatic car speed control system. We can be sure that this approach not only makes the design process of fuzzy systems simple but also produces appropriate inference results.

I. 서 론

Zadeh는 1965년 퍼지 논리(fuzzy logic) 발표 후, 일련의 연구 발표를 통해서 퍼지이론의 수학적인 바탕을 마련하고 퍼지집합 이론을 바탕으로 시스템을 기술하는 방법론을 제창하여 왔는데[1], 이 이론은 특히 퍼지 제어(fuzzy control)분야 등에 많이 응용되어 왔다. 특히 입출력 관계의 정량적인 분석이 어려운 여러 가지 공정의 제어 시스템에서 모델에 기준한 기존 제어기의 부적합한 점을 보완 대체하여 대체로 성공적으로 도입되고 있다[2]. 퍼지 제어는 Mamdani에 의해 처음 도입되었으며[3], 여기서 그는 퍼지제어를 모형 증기기관에의 제어에 집적 응용하였다.

이러한 연구의 흐름 속에서 최근 퍼지 논리는 복잡하거나 정의하기 힘든 시스템의 조작을 위해 많이 사용되어지고 있다[4]. 대부분의 퍼지 제어 시스템(fuzzy control system)은 조작자의 경험이나 엔지니어의 시스템에 대한 지식으로부터 얻은 규칙들로 이루어진 규칙베이스 시스템이다. 그러나 시스템의 복잡도가 좁아질수록 조작자가 자신이 행하는 제어 행동(control action)을 언어적으로 표현하는 데에 어려움을 느끼는 경우가 많다. 따라서 제어를 위한 퍼지 시스템의 구현에 있어 가장 큰 문제로 대두되는 것이 지식 획득(knowledge acquisition) 문제이다. 이런 문제점의 해결을 위한 하나의 방법은 조작자의 행동에 따른 입출력 자료를 근거로 그의 조절 행위의 퍼지 모델을 만드는 것이다. 이에 대한 대표적인 연구는 퍼지 시스템과 신경망의 학습 능력을 결합하는 것으로 여러 연구가들에 의해 다양한 방법들이 제안되었다[5, 6, 7]. 이러한 시도는 시스템의 성능 향상과 함께 시스템

의 입출력 자료로부터 퍼지 시스템 모델링을 목표로 하고 있다.

신경망 이론과 퍼지 이론을 모두 특정 분야에 대해서 인간처럼 일을 할 수 있는 시스템을 만드는 것에 관심이 있다. 퍼지 이론과 신경망 이론을 어떤 형태로든 함께 사용하여 보다 나은 성능을 갖는 시스템을 만들려는 연구가 많이 이루어지고 있고 그런 연구의 방향은 다음과 같이 나누어 볼 수 있다[8, 9]. 첫째로는 신경망을 이용한 퍼지 규칙의 자동적인 생성에 관한 연구이다. 신경망은 병렬처리(parallel processing)와 다단계(multistage) 추론을 하는 구조이다. 따라서 이러한 신경망과 퍼지 시스템에서의 추론과의 결합은 추론 과정에 많은 정보가 필요한 경우 추론 속도를 많이 개선할 수 있을 것이다. 셋째로 퍼지 규칙에 추론 환경에 적응하는 능력을 부여하는 것에 관한 연구이다. 추론 환경이 변함에 따라 추론 규칙에 반영될 수 있는 새로운 학습 방법의 개발이 필요하다. 이런 방법이 개발된다면 사용할수록 성능이 좋아지는 시스템의 구현이 가능해진다. 변화하는 환경에 적응하기 위해서는 새로운 사실을 수용하고, 변화된 환경에 부적절한 사실은 무시해 버리는 학습 방법이 필요하다. 그러나 위와 같은 학습 방법은 신경망에서도 아직 개발되지 않은 상태이다. 현재 신경망에서는 변화하는 환경에 적응하기 위해 완전히 다시 학습을 하거나 과거의 학습 데이터는 유지한 채 새로운 학습 데이터만 추가하는 방법을 사용하고 있다.

본 논문에서는 앞에서 살펴본 연구 방향 중 첫 번째와 두 번째의 연구 방향과 맥을 같이하여 코호넨 신경망(Kohonen neural network)과 역전파 신경망(Back-propagation neural network)을 이용한 퍼지 규칙 생

성과 이 두 신경망의 결합을 통한 추론망(Inference Network) 구축에 관한 새로운 방법을 제안한다. 이 방법에서는 응용 분야의 입출력 데이터 공간의 군집화와 군집화 과정에서 찾은 각 퍼지 규칙군(fuzzy rule patch)의 결론부 함수 근사화로 퍼지 규칙을 생성하고, 각 군집 내 변수간의 퍼지 관계(fuzzy relation)를 추론에 이용한다. 이러한 과정을 통해서 입출력 변수의 퍼지 집합(fuzzy set)과 소속 함수(membership function) 정의 과정을 생략하고도 퍼지 추론이 가능하게 된다. 기존의 퍼지 시스템 구현 과정에서 꼭 필요한 각 변수의 퍼지 집합과 소속 함수 정의 과정과 조정 과정이 생략될 수 있으므로 퍼지 시스템 구현과정이 간단해진다. 또 그 외 시스템의 입출력 변수가 많아질 경우 지금까지 제안된 다른 신경망으로 구성된 추론 시스템[7]에서 보다 적은 수의 노드로 신경망을 구성할 수 있으므로 다중 입력 변수(multi-input variable) 시스템 구현에 쉽게 이용될 수 있게 된다.

II. 신경망을 이용한 퍼지 규칙 생성과 추론망의 구축

본 연구에서 제안한 방법은 퍼지 시스템을 신경망의 학습을 통한 퍼지 규칙 생성 단계와 이 단계에서 학습이 끝난 신경망들의 결합을 통한 추론망 구성 단

계로 나누어 구현한다.((그림 1) 참조) 이 방법의 마지막 단계에서 생성되는 신경망으로 구성된 추론망은 퍼지 규칙 베이스와 추론 엔진의 역할을 함께 하게 된다.

본 연구에서 생성되는 규칙은 다른 추론 방법에 비해 우수한 기술 능력을 가지고 있다고 평가받고 있는 TS fuzzy model에서와 같이 각 규칙의 결론부가 언어적 변수 값이 아닌 조건부 변수들로부터 얻은 함수 값을 갖는다. 그러므로 퍼지 규칙 결론부의 함수 근사화(function approximation) 과정이 필요하다.

규칙 베이스 생성 단계는 두 입력 데이터 공간 군집화 과정과 퍼지 규칙의 결론부 함수 근사화 과정으로 구성된다. 이 두 과정에서는 각각 하나의 신경망을 학습시킨다. 각 신경망은 입력으로 현실 세계로부터 얻은 데이터들을 받아들이고 학습 과정을 통해 구현하고자 하는 시스템에 필요한 규칙을 찾아낸다. 첫 번째 신경망의 학습 단계에서는 비교사 학습인 코호넨 학습 방법을 사용하여 학습 데이터 내에 존재하는 입력 변수간의 관계를 파악하여 비슷한 변수간의 관계들로 이루어진 자료들을 갖는 클래스로 군집화한다. 이 때 형성된 하나의 클래스에 속하는 자료들은 같은 규칙을 가진다고 볼 수 있다. 이렇게 형성된 클래스들 내의 변수들에 대한 퍼지 관계식을 정의함으로써 퍼지 추론을 가능하게 한다. 따라서 이 과정에서는 구현하고자 하는 시스템의 퍼지 규칙 클래스를 찾는다. 다음 단계에서의 신경망은 전 단계에서 형성된 퍼지 규칙 클래스에 대해서 입력에 대한 출력 값을 학습한다. 이 단계에서 학습 방법으로 교사 학습 방법인 역전파 학습 방법이 사용된다. 퍼지 규칙 생성 단계에서 따로 학습된 두 개의 신경망은 구현하고자 하는 퍼지 시스템의 추론망 구축에 이용된다.

퍼지 시스템의 추론망은 전 단계에서 학습된 두 종류의 신경망을 결합하여 구축된다. 각 신경망은 학습 데이터에서 찾아낸 변수들 간의 규칙 정보를 신경망의 연결 강도에 모두 저장하고 있으므로, 두 신경망의 결합에서 구축된 추론망은 퍼지 시스템의 추론 엔진뿐만 아니라 퍼지 규칙 베이스의 역할을 함께 한다. 그러므로 기존의 퍼지 시스템에서 순차적으로 이루어지는 규칙 베이스 탐색이 병렬적으로 이루어질 수 있으며 규칙 베이스 탐색 과정을 거친 후 따로 이루어지던 추론 과정이 탐색과 동시에 이루어지게 되므로

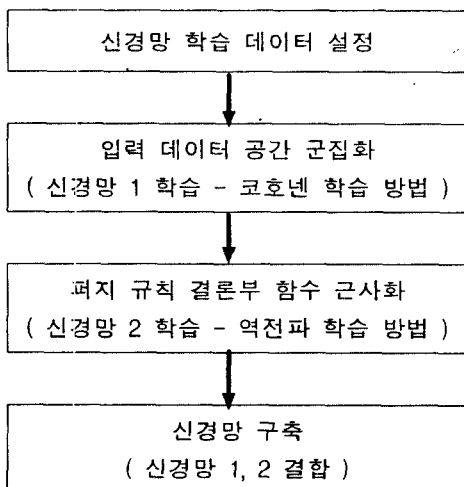


그림 1. 시스템 구현 단계

실시간 처리가 필요한 시스템에 유용하다. 또한 학습 데이터 내 변수간의 퍼지 관계를 기반으로 하여 추론이 이루어지므로 데이터의 모든 변수에 대한 퍼지 소속 함수를 따로 정의하는 대신 하나의 퍼지 관계식만을 정의하면 된다. 따라서 지금까지 퍼지 시스템 구현 과정에서 퍼지 변수의 소속함수 모양 등을 조정해야 하는 과정이 생략될 수 있으므로 퍼지 시스템 구현이 수월해진다고 볼 수 있다. 물론 퍼지 관계 함수의 정의도 잘 이루어져야 하고 함수의 조절 등이 필요한 경우가 발생할 수 있으나 기존의 퍼지 시스템에서 많은 퍼지 변수에 대한 소속 함수의 조정을 하는 것보다는 훨씬 쉽게 퍼지 시스템 구현이 가능하다.

2.1 신경망을 이용한 퍼지 규칙의 생성

본 연구에서 규칙 베이스 생성을 위해 사용하는 신경망 학습 방법은 코호넨 학습 방법과 역전파 학습 방법이다. 코호넨 학습 방법은 입출력 데이터 내의 퍼지 규칙 클래스를 찾는 단계에서 사용되고, 역전파 학습 방법은 각 규칙의 결론부의 함수 근사화를 통해 추정값을 얻는 과정에 사용된다.

코호넨 학습 방법을 이용하는 신경망의 경우는 연결 강도들이 패턴들의 분포에 따라 구조화되므로, 특정 패턴 클래스 정보뿐만 아니라 클래스들 간의 공간적 관계 정보까지도 표현 가능하다. 그러므로, 학습과정을 거쳐 형성된 클래스들은 학습 데이터를 특성별로 분류했다고 볼 수 있다. 따라서, 한 클래스에 속하는 데이터들은 변수간에 존재하는 규칙이 비슷하다고 볼 수 있고, 한 클래스를 대표하는 규칙은 시스템의 입력값이 그 클래스에 속하는 정도를 일반적인 퍼지 시스템에서 퍼지 규칙 조건부 적용도가 반영된 결론부의 소속함수값처럼 사용하여 적절한 출력값을 내줄수 있게 나타내 변수간의 관계를 찾는다. 위에서 살펴 본 바와 같이 퍼지 관계를 기반으로 한 추론 과정은 모든 변수에 대한 퍼지 집합의 정의가 필요 없으므로 이제껏 퍼지 논리 시스템의 구현에서 어려운 과정이었던 퍼지 집합의 정의와 각 퍼지 집합의 소속 함수 정의 부분이 생략될 수 있고, 따라서 퍼지 논리 시스템 구현 과정이 훨씬 간단해지고 시스템 성능 영향 요인의 분석이 용이해진다.

본 연구에서는 퍼지와 되지 않은 입출력 공간의 군집화를 통해 퍼지 관계를 얻고 이 퍼지 관계식을 정

의함으로써 퍼지 추론을 가능하게 한다. 또한 퍼지 규칙의 출력 부분을 언어적 변수값이 아닌 입력에 대한 함수값을 사용하고, 그 함수를 신경망의 학습을 이용하여 얻게 되므로 출력력을 위한 소속함수의 정의를 따로 할 필요가 없다.

2.1.1 입출력 데이터 공간의 군집화를 이용한 퍼지 규칙의 생성을 위한 신경망 학습

신경망의 학습을 위해 주어진 데이터들은 시스템의 입력과 출력으로 이루어지고, 신경망은 이 데이터들을 입력으로 받아들이고 비교사 학습 방법 중 하나인 코호넨의 학습 방법을 이용하여 주어진 입력들 간의 관계를 학습하게 한다. 시스템의 입력과 출력 값으로 구성된 학습 데이터는 퍼지화 과정 없이 정규화 과정만을 거친 후 신경망의 입력으로 사용된다. 입출력 데이터 공간의 군집화를 위한 신경망의 구조는 <그림 2>와 같이 구성된다.

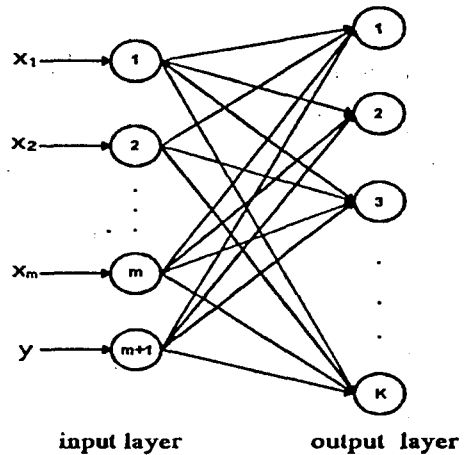


그림 2. 입출력 공간 군집화를 위한 신경망 구조

$X=[x_1, x_2, \dots, x_m]^T$ 를 구현하고자 하는 시스템의 입력 벡터, $Y=[y]^T$ 를 출력 벡터라 할 때, 신경망의 학습 데이터는 이 두 벡터가 합쳐진 벡터, $U=[X_i, y_i]^T$ $i=1, 2, \dots, N$ 으로 구성된다. 신경망의 출력층 노드의 개수는 입출력 데이터 공간 내의 군집의 개수로, 규칙베이스를 구성하는 규칙의 수를 의미한다. 이러한 구조의 신경망은 코호넨의 학습방법을 이용하여 학

습하게 되고 학습 결과 출력층의 각 노드는 학습 데이터 공간에 존재하는 데이터의 변수간의 관계를 파악하여 나누어진 클래스로서, 각각의 퍼지 규칙을 의미한다.

코호넨 학습의 $X \times Y$ 에서 서로 위치적으로 가까운 입출력 데이터들을 같은 덩어리로 묶는 역할을 한다. 만약 K 개의 군집으로 나누었다면, k 번째 노드와 연결된 연결강도 벡터 $W_k = [W_{k1}, W_{k2}, \dots, W_{km+1}]$ 는 그 군집에 속하는 데이터들의 특성을 나타내고 있다. 즉 입출력 공간상에서 연결강도 벡터 W_k 는 그 클래스의 중심값을 갖고 있고, 따라서 입출력 공간의 한 점으로 표현되는 하나의 데이터는 각 클래스의 중심과의 거리에 따라 그 클래스에 속하는 정도를 결정할 수 있다. 이러한 코호넨 신경망의 특성을 이용하여 k 번째 군집에 대한 퍼지 관계 R_k 의 퍼지 관계식을 정의한다. 즉 코호넨 신경망에서 연결강도는 각 군집의 중심값을 나타내므로 주어진 입력 벡터 X 와 연결강도 벡터 W 와의 거리의 차이가 크다면 입력 벡터 X 가 대표하는 클래스에 속하는 정도가 작다는 것이고, 그 값이 작다면 반대의 의미가 된다. 따라서 퍼지 관계식 $\mu_{R_k}(x, y)$ 는 아래의 식과 같이 정의된다.

$$d_k = \left(\sum_{i=1}^m (w_{ki} - x_i)^2 + (w_{km+1} - y)^2 \right)^{1/2}, \quad k=1, \dots, K \quad (1)$$

$$\mu_{R_k}(X, Y) = \exp^{-\alpha \cdot d_k}, \quad k=1, 2, \dots, K \quad (2)$$

<식 1>의 d_k 는 입력 데이터 U 와 k 번째 노드의 연결강도 W_k 와의 공간상의 유클리드 거리(Euclidean distance)를 계산한 값이다. 그리고 <식 2>는 퍼지 관계

식으로서 주어진 입력 U 가 클래스가 나타나는 퍼지 규칙에 속하는 정도를 결정한다. 이 함수의 형태는 <그림 3>과 같다.

<그림 3>에서 볼 수 있듯이 d_k 값이 0에 가까워질수록 μ_{R_k} 값은 1에 가까워지고 d_k 값이 커질수록 μ_{R_k} 값은 0이 되므로 $\mu_{R_k}(X, Y)$ 는 퍼지 관계 $R_k(X, Y)$ 의 식으로 타당하다고 본다. 위의 퍼지관계함수에 의하면, k 번째 군집의 중심값을 갖고 있는 $W_k = [w_{k1}, \dots, w_{km+1}]$ 에 대해 주어진 임의의 입력 $U = [x_1, x_2, \dots, x_m, y]$ 는 U 와 W_k 와의 유클리드 거리를 계산하여 위 그래프에 따라 해당하는 함수값을 W_k 가 대표하는 규칙의 조건부 적합도 $\mu_{R_k}(X, Y)$ 로 결정한다. 이 과정에서 전체 변수에 대한 퍼지화 과정이 이루어진다고 볼 수 있다. 퍼지화가 퍼지 규칙의 조건부 변수 하나 하나에 대해 따로 이루어지고 다시 그 값들을 모아 각 규칙의 조건부 적합도를 결정하는 기존의 퍼지 시스템의 추론 과정이 이 방법에서는 모두 함께 이루어진다. 이것은 입출력 데이터에 대한 군집화가 이루어져서 변수간의 상관 관계를 파악했기 때문에 가능하다고 본다. 지금까지 살펴본 신경망의 학습 과정을 단계별로 살펴보면 다음과 같다.

학습이 충분히 이루어진 후에는 각 노드의 연결 강도 벡터는 노드의 공간적인 위치근처의 점들을 대표하는 값으로 수렴한다. 그러므로, 신경망의 학습이 끝나면 출력층의 노드는 비슷한 특성을 갖는 데이터 변수간의 공통의 규칙을 나타낸다. 다음 단계에서는 입력에 대해 이 단계에서 6서 각 퍼지 규칙의 결론부 추정 값을 결정한다. 다음 단계로 넘어가기 전에 앞서 정의한 퍼지 관계식을 이용하여 학습 데이터에 대한 각 규칙의 반영정도를 구한다. 이 값들은 퍼지 규칙의 결론부 추정값 결정에 이용된다.

2.1.2 퍼지 규칙의 결론부 함수 근사화를 위한 신경망 학습

이 단계에서는 코호넨 학습을 이용해 군집화된 입출력 공간의 각 클래스(퍼지 규칙)에 대해서 입력에 대한 퍼지 시스템의 출력을 별도의 신경망에 학습시킨다. 본 논문에서는 퍼지 규칙의 결론부를 얻어 변수가 아닌, TS fuzzy model과 비슷하게, 함수 $g(x)$ 로 나타낸다. 함수 $g(x)$ 를 얻는 과정을 함수 근사화 과정이라 한다. 앞서서도 살펴보았지만 TS fuzzy model은

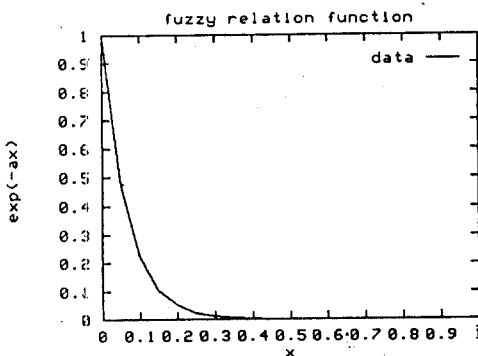


그림 3. 퍼지 관계 함수

1. 신경망의 연결 강도를 초기화한다. 연결 강도는 임의적으로 주어지는데 $-0.5 \sim +0.5$ 의 값의 범위를 갖도록 하였다.
2. 입력 데이터를 설정한다. 신경망의 입력 데이터는 구현하고자 하는 퍼지 시스템의 입력과 출력으로 구성되어 있다. 이 학습 데이터들은 현실 세계에서 얻은 것으로 가정한다.
3. 입력 데이터 U 와 연결 강도 W 간의 거리 d_k 를 계산하여 승자 노드(winner node)를 결정한다.

$$d_k = \left(\sum_{i=1}^m (w_{ki} - x_i)^2 + (w_{k,m+1} - y)^2 \right)^{1/2}, k=1, \dots, K$$

$$\|X - W_c\| = \min_i \{\|X - W_i\|\}$$

첨자 c 는 승자 노드를 의미한다.

4. 승자 노드가 결정되면 이웃 반경(neighborhood distance)안에 있는 노드들과 함께 승자 노드를 학습시킨다.

$$W_i(t+1) = \begin{cases} W_i(t) + a(X - W_i(t)) & i \in N_c \\ 0 & \text{otherwise} \end{cases}$$

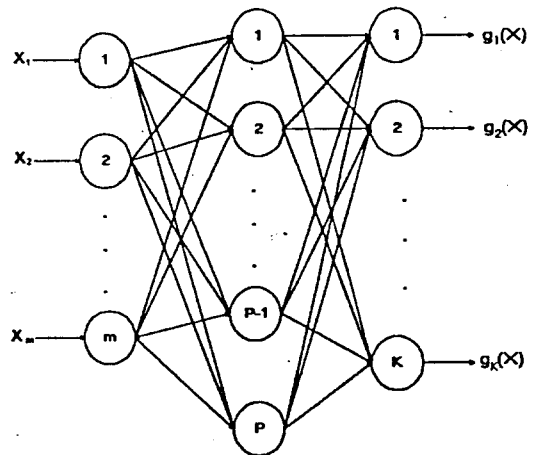
N_c 는 이웃 반경 안에 드는 노드들의 집합이다. 이 연결 강도 수정 과정을 통해서 연결강도 벡터 W 는 입력 벡터 X 를 향해 조금씩 조정된다.

알고리즘 1. 입출력 공간 근집화를 위한 신경망 학습 단계

결론부가 조건부 변수의 함수 형식으로 나타낸다. 그러므로 결론부도 소속함수 정의는 필요하지 않다. 본 연구에서는 이 결론부의 함수역학을 신경망이 하게 된다. TS-model은 매개변수 식별(parameter identification) 문제가 선행되어야 하는 어려움을 갖고 있으나, 본 논문에서는 신경망을 이용함으로써 매개변수 식별 문제를 제거한다. 규칙의 결론부 함수 근사화를 위한 신경망의 구조는 <그림 4>이다. 학습은 역전파 학습 방법을 이용하여 이루어진다. 역전파 학습은 목적 함수(object function)의 정의를 필요로 하는데 여기서서는 다음 식과 같은 목적 함수를 사용한다.

$$J_k(g_k) = \sum_{i=1}^N [\mu_{Rk}(X_i, Y_i) g_k(X_i) - Y_i]^2, k=1, 2, \dots, K \quad (3)$$

신경망은 <식 3>의 목적 함수값을 최소화하는 방향으로 학습한다. $\mu_{Rk}(X_i, Y_i)$ 는 전 단계의 마지막 과정



input layer hidden layer output layer

그림 4. 결론부 함수 근사화를 위한 신경망

에서 구한 학습 데이터에 대한 각 규칙의 조건부 적합도이다. 신경망의 학습은 일반적인 역전파 학습과 단계는 동일하나 연결강도 수정 단계에서 출력층의 노드가 나타내는 퍼지 규칙의 조건부 적합도, 즉 퍼지 관계함수값을 고려한다는 것에서 차이가 난다. 연결 강도 수정은 다음의 <식 4, 5>와 같이 이루어진다.

$$\delta_{pk} = \mu_{Rk}(X_i, Y_i)(Y_i - g_k(X_i))f_k(\text{net}_{pk}) \quad (4)$$

$$W_{kj} = W_{kj} + a\delta_{pk}g_k(X_i) \quad (5)$$

학습은 학습 자료 전체에 대해 ($J_k(g_k) < \theta$)이거나 (반복횟수 > 제한횟수)인 경우가 될 때까지 계속 수행한다. θ 는 임계치로서 제한 횟수와 마찬가지로 신경망이 충분히 학습되도록 정해준다.

<그림 4>에서 보듯이 신경망은 입력층, 은닉층, 출력층으로 구성되고, 입력으로는 입력 데이터 X가 들어가고 출력으로는 입력에 대한 각 규칙의 근사화된 함수값 $g_k(X)$ 가 계산되어 나온다.

2.1.1, 2.1.2 두 단계에서 따로따로 학습된 두 신경망에 입출력 공간에 존재하는 퍼지규칙에 대한 정보가 분산되어 저장되어 있다. 이렇게 나뉘어 저장되어 있는 정보를 다음 단계에서 결합함으로써 퍼지 규칙 베이스가 실질적으로 구축되며 동시에 추론망의 구축도 이루어진다.

2.2 추론망의 구축

2.2에서 두 단계의 학습 과정을 거쳐 생성된 퍼지 시스템의 퍼지 규칙은 실질적으로 두 개의 신경망의 연결 강도에 의해 표시된다. 추론망은 이 두 신경망의 결합으로 생성된 새로운 신경망으로 구성하게 되고, 이 신경망은 학습과정 없이 바로 추론에 사용된다. 이 결합된 신경망의 구조는 <그림 5>와 같다. 퍼지 규칙은 모두 그대로 보존하고 있다.

추론망의 구조를 살펴보면, 시스템의 입력이 두 부분으로 들어간다. 퍼지 규칙의 결론부 추정값 생성을 위한 역전파 신경망으로 들어간 입력은 그에 대한 각 퍼지 규칙의 결과값을 얻고, 그 결과값 $g_i(x)$ 와 입력은 코호넨 신경망으로 입력되어 각 퍼지규칙의 결론부 소속함수값을 얻는다.

다음 단계는 비퍼지화 과정으로 각 규칙의 결과값의 결론부 소속함수값을 이용하여 단일 값인 퍼지 시

스템의 결과를 얻는다. 이 과정에서는 비퍼지화 기법으로 무게중심을 사용한다. 비퍼지화 식은 <식 6>과 같다.

$$Y = \frac{\sum_{k=1}^K \mu_{Rk}(X, g_k(X)) g_k(X)}{\sum_{k=1}^K \mu_{Rk}(X, g_k(X))} \quad (6)$$

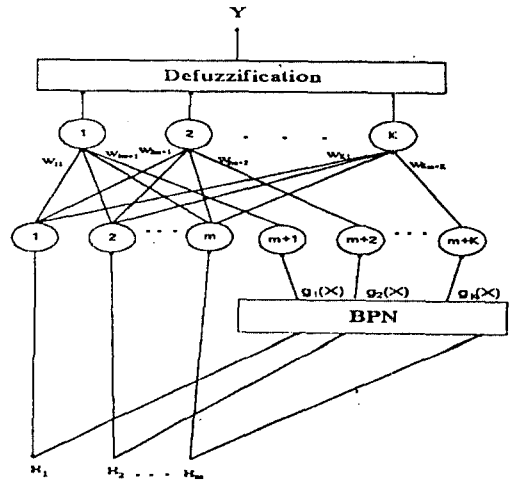


그림 5. 추론망의 구조

<그림 5>에서 연결강도 $w_{1m+1}, w_{2m+2}, \dots, w_{km+k}$ 는 <그림 2>에서의 연결강도 $w_{1m+1}, w_{2m+1}, \dots, w_{km+1}$ 과 각각 같은 값을 갖는다.

III. 실험 및 결과

지금까지 II장에서 살펴본 퍼지시스템 구현방법을 이용하여 실제 퍼지시스템을 구현해 보고, 구현된 시스템의 추론 결과가 타당한지 실험해 본다. 구현하는 퍼지시스템은 자동차 정속 주행 시스템으로 실험은 IBM PC 호환기종인 486 시스템에서 행하였다. 단 규칙 생성과 추론망 구축을 위한 신경망 학습은 workstation 상에서 하였다.

3.1 자동차 정속 주행 시스템에의 적용

자동차 정속 주행 시스템은 제어 대상차의 안정성을

확보하면서 운전자가 지정한 속도로 정속 주행을 하는 시스템이다. 이 시스템에서는 제어 차량과 제어 차량의 앞에서 주행하는 차량만을 고려하여 추론한다. 각 자동차들은 0.1초 단위로 속도, 가속도, 이동거리가 계산된다.

신경망의 학습을 위한 학습 데이터는 자동차 정속 주행 시스템의 입력 변수의 X_1, X_2, X_3 와 출력 변수의 Y 로 구성된다. Y 는 다음 단계의 가속량이다. 각 변수의 의미는 다음과 같다.

- $X_1(t)$ = 제어 대상차의 속도 - 목표 정속 속도
- $X_2(t) = X_1(t) - X_1(t-1)$
- $X_3(t)$ = 앞차의 위치 - 제어 대상차의 위치
- $Y(t)$ = 추론 결과

규칙 생성과 추론망 구축을 위해 신경망은 선택된 430개 데이터로 이루어진 학습 데이터의 일부를 보며 다음 <표 1>과 같다.

<그림 6>에서 보면 제어 차량과 앞차와의 거리가 가까워지면 안전 거리 확보를 위해 제어 차량을 감속하고, 다시 차간 거리가 멀어지면 목표 속도에 도달하기 위해 감속함을 알 수 있다.

시스템 구현에 사용되는 두 개의 신경망 구조는 다음과 같다. 입출력 데이터 공간 군집화 단계에서 사용되는 코호넨 신경망은 입력층 노드 4개, 출력층 노

표 1. 학습 데이터

	$X_1(t)$ km/h	$X_2(t)$ km/h	$X_3(t)$ m	$Y(t)$ km/h
1	-96.4	-1.8	201.0	1.80
2	-96.6	-1.8	201.4	1.80
3	-92.8	-1.8	201.8	1.80
4	-91.0	-1.8	202.0	1.80
5	-89.2	-1.8	202.2	1.80
6	-88.5	-0.7	202.3	0.72
7	-87.8	-0.7	202.3	0.72
8	-83.4	-0.7	201.7	0.72
9	-82.7	-0.7	201.5	0.72
10	-82.0	-0.7	201.2	0.72
11	-77.0	-0.7	197.8	0.72

드 20개로 구성되어 있고, 퍼지 규칙 결론부 함수 근사화를 위한 역전파 신경망은 입력층 3개, 은닉층 7개, 출력층 20개의 노드를 갖는다.

부 함수 근사화를 위한 역전파 신경망은 입력층 3개, 은닉층 7개, 출력층 20개의 노드를 갖는다.

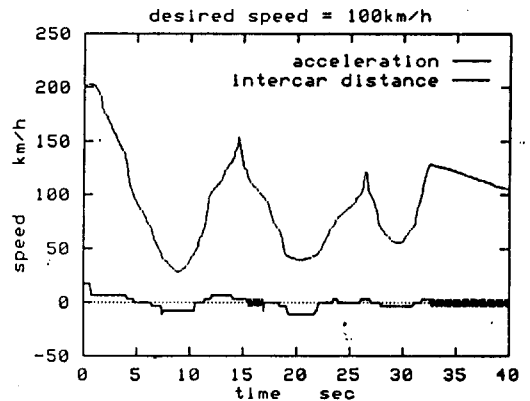


그림 6. 학습 데이터

3.1.1 입출력 데이터 공간의 군집화 결과

<그림 7>은 신경망의 입력으로 사용된 시스템의 입출력 데이터 공간이다. x축은 데이터 변수 중 X_1 , y축은 가속량 Y , z축은 차간거리 X_3 를 나타낸다. X_2 와 Y 값은 절대값이 항상 같기 때문에 X_2 를 제외하고 3차원으로 나타내었다. 그림에서 볼 수 있듯이 시스템의 입출력 데이터들은 공간상에서 명확하게 군집을 이루고 있지 않다. 이러한 데이터들을 군집화하는 것은

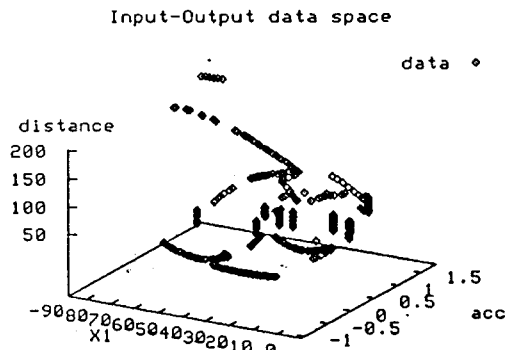


그림 7. 학습 데이터의 분포

어려운 일이고, 또 변수와의 상관 관계를 파악하기도 힘들다. 더구나 이와 같은 특성을 갖는 시스템을 위한 규칙을 퍼지 변수를 이용하여 표현하는 것은 매우 어려운 일이다.

코호넨 신경망 학습을 통해 학습 데이터 공간을 군집화하고 변수간의 상관 관계를 파악한다. <그림 8>는 입출력 데이터 공간 군집화를 위한 신경망의 학습이 끝난 후 신경망의 출력층 노드들, 즉 규칙 노드가 갖는 연결강도를 공간상에 학습 데이터와 함께 표시한 결과이다. 신경망의 출력층 노드의 연결강도는 그 노드가 대표하는 군집의 중심값을 갖는다. 데이터들은 그 주위에 가깝게 위치하는 중심점과 같은 군집에 속한다.

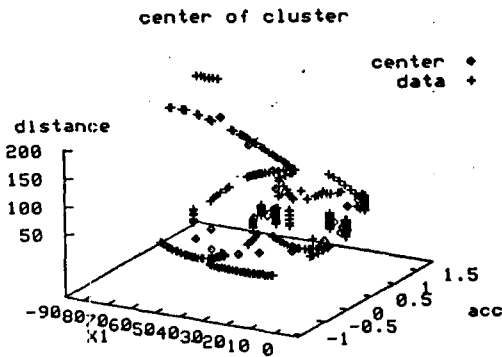


그림 8. 학습 데이터 공간 군집 결과

3.2 실험결과

본 실험은 몇 가지 상황에 대해서 제안한 방법으로 구현된 시스템이 적절한 추론을 하는지 검증하기 위해 실시되었다.

3.2.1. 제어 대상차 단독 주행시 실험 결과

이 실험은 앞차가 없는 가장 기본적인 상황, 즉 제어 대상자가 외부의 상황을 고려하지 않고 정속 주행하는 경우의 추론 결과를 살펴보는 것이 목적이다. 본 실험에서는 제어차의 최초 속도를 50km/h, 목표 속도를 70km/h로 설정한 경우의 실험 결과를 살펴보았다. 실험 결과는 <그림 9>과 <그림 10>에서 볼 수 있다.

<그림 9, 10>는 시간에 대한 제어 대상차의 속도를 나타낸다. 그림에서 보듯이 자동차는 시간이 지남에

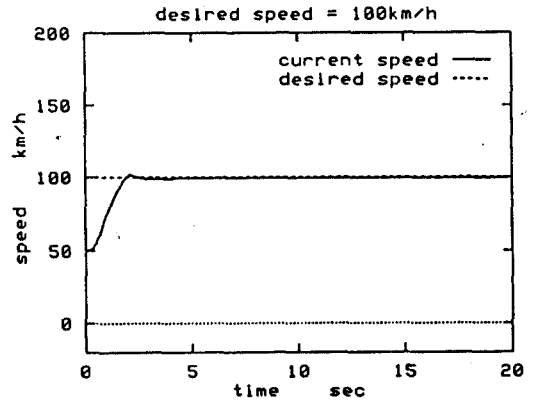


그림 9. 목표 정속 속도가 100km/h인 경우

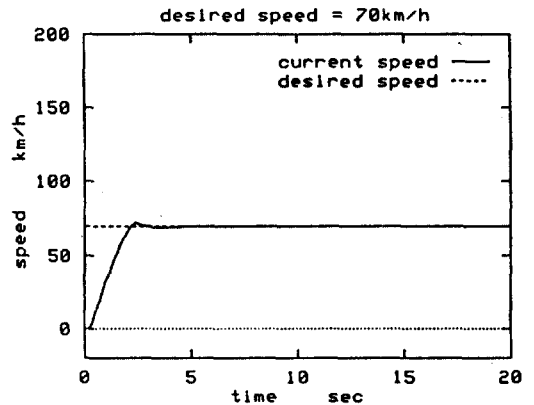


그림 10. 목표 정속 속도가 70km/h인 경우

따라 각각 목표 정속 속도인 100km/h와 70km/h에 근접하게 되고, 목표 속도에 도달한 후에는 속도에 큰 변화 없이 정속 주행을 하고 있다. 따라서 제어 대상차의 단독 주행시 추론이 타당하게 이루어지고 있음을 알 수 있다.

3.2.2 목표 속도의 변화에 따른 실험 결과

이 실험은 운전자가 주행 중에 정속 주행을 위한 목표를 임의로 변화시켰을 때의 시스템의 추론 결과를 살펴보는 것을 목적으로 수행하였다. 시스템의 추론에 영향을 미치는 다른 요인인 앞차와의 차간거리는 충분히 멀다고 가정한다. 제어 대상차의 초기 속도는 50km/h로 설정하였고 목표 속도를 80km/h, 100km/h

로 변환시킨 경우와, 100km/h, 120km/h, 140km/h로 변환시킨 경우로 나누어 실험하였다. 실험 결과는 <그림 11>와 <그림 12>에서 확인할 수 있다.

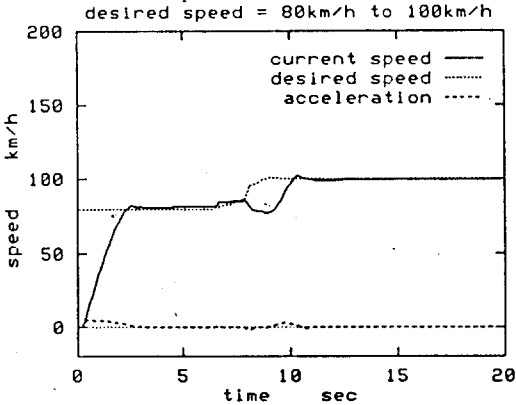


그림 11. 목표 속도가 80km/h, 100km/h로 변환 경우

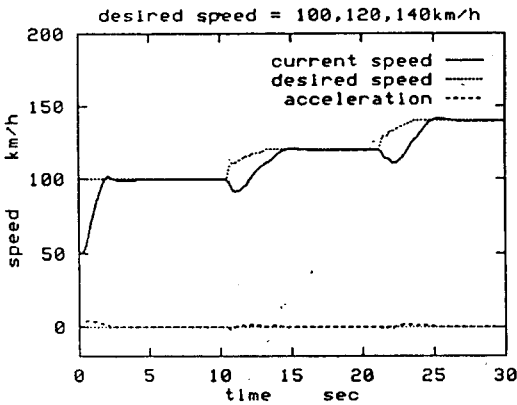


그림 12. 목표 속도가 100km/h, 120km/h, 140km/h로 변환 경우

<그림 11, 12>에서 볼 수 있듯이 단위 시간당 목표 속도의 변화량이 아주 큰 경우 잘못된 추론이 이루어지고 있다. 그 이유는 이러한 경우의 변수간 관계를 신경망이 학습하지 못 했기 때문이다. 즉, 신경망 학습데이터에 이러한 경우에 해당하는 데이터가 포함되지 않아서 신경망이 이 경우를 학습하지 못했기 때문이다. 그러나 목표 속도의 변화율이 작아진 후에는 다시 올바른 추론을 한다. 이 실험에서 신경망 학습데이터의 중요성을 확인할 수 있다.

3.2.3 앞차와 차간 거리를 고려한 실험

이 실험에서는 제어 대상차의 정속 목표속도가 100km/h, 초기 속도는 50km/h이고 앞차의 주행속도는 70km/h이며 최초의 차간 거리는 100m인 상황을 가정하고 실험한다. 제어 대상차의 목표 속도가 앞차의 주행 속도보다 큰 경우, 제어차가 안전 거리 확보를 위해 속도를 조절하는지를 확인한다. 실험 결과는 <그림 13>에서 볼 수 있다.

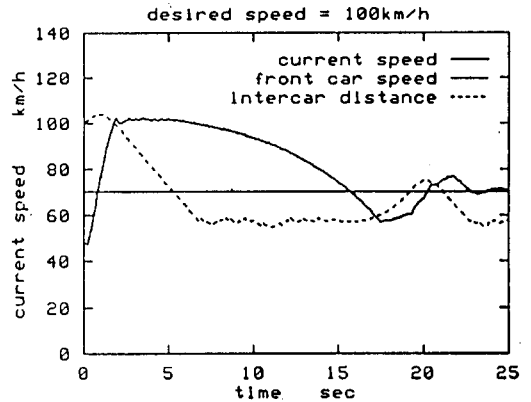


그림 13. 앞차의 주행 속도가 70km/h 일 때 차간 거리를 고려한 경우

<그림 13>에서 보면 제어 대상차는 차간 거리가 100m 이상일 때 이미 목표 속도에 도달하여 정속 주행을 하다가 차간 거리가 약 70m이하가 되면서 앞차와의 충돌을 방지하기 위하여 가속하기 시작한다. 속도가 60km/h 이하로 떨어졌다가 다시 차간 거리가 멀어지면서 약간 속도를 높여 앞차의 주행 속도인 70km/h 근처로 속도를 조절하고 있다. 이후부터 제어 대상차는 차간 거리를 고려하면서 목표 속도에 도달하려는 노력을 계속한다.

이제까지 여러 상황에 대한 시스템의 추론 결과를 살펴보았다. 각각의 실험 결과에서 추론이 타당하게 이루어짐을 확인할 수 있었다.

IV. 결 론

본 연구에서는 일반적으로 퍼지 시스템을 구현하는 데에 있어서 어려운 과정으로 여겨지는 지식 획득

과 퍼지 변수의 소속 함수 정의 과정이 갖고 있는 문제점을 해결하는 퍼지 시스템 구현 방법을 제안하였다. 제안된 방법에서는 자기 조직화 지도(SOM)와 역전파 신경망을 사용하여 주어진 입력 데이터로부터 시스템의 규칙을 자동 생성한다. 또한 각 군집 내 변수간의 퍼지 관계에 기반을 두어 추론을 함으로써 시스템의 입력 변수와 출력 변수들에 대한 퍼지화 과정 없이도 퍼지 추론을 행할 수 있었다. 즉 시스템 변수들의 퍼지 소속함수의 정의 없이 퍼지 추론이 가능했다. 퍼지화 과정을 생략함으로써 퍼지 시스템 구현 과정에서 적절한 소속 함수 정의의 어려움이 없어질 수 있고, 소속 함수 값을 이용하는 다른 신경망 구조에 비해 적은 수의 신경망을 입력 노드 만으로도 다중 입력 변수 시스템의 구현이 가능하다.

규칙의 자동생성과 퍼지화 과정의 생략으로 퍼지 시스템 구현 과정이 간단해졌지만, 제안된 방법이 실현에 있어서 다소 문제가 된 부분은 입출력 데이터 공간의 군집화를 위한 신경망의 출력 노드의 개수를 결정하는 부분이었다. 실제 실험에서는 시행 착오를 거치면서 가장 적절한 노드의 개수를 결정해 주었다. 이 문제의 해결을 위해 앞으로 적절한 군집화에 대한 타당성 있는 기준에 대한 연구 방향이나, 신경망이 학습하는 과정에서 스스로 적절한 출력 노드의 수를 결정할 수 있는 새로운 학습 알고리즘을 개발하는 방향으로 연구가 이루어져야 할 것이다.

참 고 문 헌

1. Abraham Kandel, Lihong Li and Zhiqiang Cao, "Fuzzy inference and its applicability to control systems," Fuzzy Sets and Systems 48, pp. 99-111, 1992.
2. E. H. Mamdani, "Application of fuzzy algorithm for control of simple dynamic plant," IEEE Proc. Control & Science, Vol. 121, No. 12, pp. 1585-1588, Dec 1974.
3. L.A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-3, No. 1, pp. 28-44, jan. 1973.
4. C. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part I and II," IEEE Trans. on System, Man, and Cybernetics, Vol. 20, No. 2, pp. 404-435, march/April 1990.
5. James M. Keller, Ronald R. Yager, and Hossein Tahani, "Neural network implementation of fuzzy logic," Fuzzy Sets and Systems 45, pp. 1-12, 1992.
6. Bart Kosko, Neural Networks and Fuzzy Systems, A Dynamical Systems Approach to Machine Intelligence, Prentice-Hall, 1992.
7. Chin-Teng Lin, C. S. George Lee, "Neural-Network-Based Fuzzy Logic Control and Decision System," IEEE Trans. on Computer, Vol. 40, No. 12, pp. 1320-1336, Dec, 1991.
8. Hideyuki Takagi, "Fusion Technology of Fuzzy Theory and Neural Networks-Survey and Future Directions," proc. of the Int. of Conf. on Fuzzy Logic and Neural Networks(Iizuka), pp. 13-26, July 20-24, 1990.
9. Tomohiro Takagi and Michio Sugeno, "Fuzzy Identification of Systems and Its Application to Modeling and Control," IEEE Trans. on System, Man, and Cybernetics, Vol. SMC-15, No. 1, pp. 116-132, Jan/Feb. 1985.
10. 이광형, 오길록, 퍼지 이론 및 응용 I, II, 홍릉과학출판사, 1991.
11. Richard P Lippmann, "An Introduction to Computing with Neural Nets," IEEE ASSP Magazine, pp. 4-22, April 1987.
12. Korn, G. A., Neural Network Experiments on Personal Computers and Workstations, Massachusetts Inc. of Tehnology, 1991.
13. James A. Freeman, David M. Skapura, Neural Networks Algorithms, Applications, and Programming Techniques, Addison-Wesley, 1992.
14. Helge Ritter, Thomas Martinetz, Klaus Schulten, Neural Computation and Self-Organizing Maps, Addison-Wesley, 1992.
15. 김대수, 신경망 이론과 응용(I, II), 하이테크정보, 1992.

