

경쟁적 퍼지다항식 뉴런에 기초한 고급 자기구성 뉴럴네트워크

論 文

53D-3-1

Advanced Self-Organizing Neural Networks Based on Competitive Fuzzy Polynomial Neurons

朴 浩 成* · 朴 建 俊** · 李 東 潤*** · 吳 聖 權§

Ho Sung Park, Keon-Jun Park, Dong-Yoon Lee, and Sung Kwun Oh

Abstract - In this paper, we propose competitive fuzzy polynomial neurons-based advanced Self-Organizing Neural Networks(SONN) architecture for optimal model identification and discuss a comprehensive design methodology supporting its development. The proposed SONN dwells on the ideas of fuzzy rule-based computing and neural networks. And it consists of layers with activation nodes based on fuzzy inference rules and regression polynomial. Each activation node is presented as Fuzzy Polynomial Neuron(FPN) which includes either the simplified or regression polynomial fuzzy inference rules. As the form of the conclusion part of the rules, especially the regression polynomial uses several types of high-order polynomials such as linear, quadratic, and modified quadratic. As the premise part of the rules, both triangular and Gaussian-like membership function are studied and the number of the premise input variables used in the rules depends on that of the inputs of its node in each layer. We introduce two kinds of SONN architectures, that is, the basic and modified one with both the generic and the advanced type. Here the basic and modified architecture depend on the number of input variables and the order of polynomial in each layer. The number of the layers and the nodes in each layer of the SONN are not predetermined, unlike in the case of the popular multi-layer perceptron structure, but these are generated in a dynamic way. The superiority and effectiveness of the proposed SONN architecture is demonstrated through two representative numerical examples.

Key Words : SONN(Self-Organizing Neural Networks), activation node, FPN(Fuzzy Polynomial Neuron), regression polynomial fuzzy inference, the basic and the modified type, the generic and the advanced type.

1. 서 론

입출력 데이터쌍의 집합으로부터 미지의 함수를 평가하는 것은 계속 진행되고 있으며 이는 다양한 과학과 공학분야 [1-4]에서 여전히 중요한 논점이 된다. 전통적인 방정식 기반 근사화 이론들은 잘 개발되고 실제 문제에서 성공적으로 실행되고(특히 선형인 경우에)있다. 동시에 비선형 함수 근사화에 대해서 신경망과 퍼지 기술과 같은 방법에 상당한 관심이 집중되고 있다[5]. 특히, 고계 비선형과 다변수 방정식을 가진 수학적 모델을 처리할 때는 그들의 파라미터 모두를 평가할 수 있는 방대한 양의 데이터들이 필요하다. 이러한 문제를 풀기 위한 하나의 방법으로 A. G. Ivakhnenko[6]는 GMDH(Group Method of Data Handling) 알고리즘을 소개하였다. GMDH는 시스템의 입력과 출력사이의 비선형 관계를 동정하기 위한 분석적인 기술이며, 방대한 데이터의 양이 나 또는 시스템의 특별한 지식이 없어도 복잡하고 비선형 시

스템에 대한 모델을 구축할 수 있다. 또한 기존의 표준 신경망보다 더 적은 수의 노드를 가지고 적응성면에서 더 우수한 특성 때문에 과학과 공학[7,8] 분야에서 우수한 예측 능력을 가진 모델에 이용되어 왔다. GMDH 알고리즘은 데이터의 부분 표현식의 사용에 의해서 각 층에서의 노드들과 층들이 자기 스스로 연속적인 생성을 통해 최적의 구조를 구축하는 알고리즘이다. 그리고 퍼지 추론 방법, 특히 규칙 베이스 퍼지 모델에 의한 모델링과 예측은 동정 방법에 따라 많은 연구가 수행되어 왔다. 언어적 접근방식에서 Tong은 논리적 조사 방법에 의해 가스로 공정을 동정하였고[9], B. Li등은 이 방법의 수정으로 더 좋은 결과를 얻는 방법[10]과 결정 테이블에 기초한 자기 학습 알고리즘을 제안하였다. 퍼지 관계 방정식에 기초한 접근 방식에서, Pedrycz는 퍼지 관계 방정식에 의해 새로운 합성규칙인 기준 퍼지 집합과 Zadeh의 제한적 가능성 분포의 개념에 기초해서 동정하였고[11], Xu는 이 두 가지 방법에 기초한 기준 퍼지 집합 이론과 자기 학습 알고리즘에 의해 퍼지 관계 모델을 구성하여 동정하였으나[10,12], 직접 추론 방식을 사용했기 때문에 혼합(선형)추론 방식보다 그 성능이 떨어지는 단점이 있다. GMDH 방법을 이용한 Tanaka등[13,14]은 확신도 인자를 가진 퍼지 후반부 추론 규칙을 구성하기 때문에 좋은 특성을 얻지 못한다. 이들 모든 퍼지 동정 알고리즘은 주어진 공정 데이터에 기초함으로써 보다 좋은 성능 즉, 근사화 능력에 초점이 맞추어져있고, 많은 최적화 기법과 함께 모듈화된 특별하고 상세한 규칙들의

* 正 會 員 : 圓 光 大 學 制 御 計 測 工 學 科 博 士 課 程

** 正 會 員 : 圓 光 大 學 制 御 計 測 工 學 科 碩 士 課 程

*** 正 會 員 : 中 部 大 學 情 報 通 信 工 學 科 助 教 授 · 工 博

§ 正 會 員 : 圓 光 大 學 電 氣 電 子 工 學 部 副 教 授 · 工 博

接 受 日 字 : 2003년 9월 03일

最 終 完 了 : 2003년 12월 23일

제한적 표현능력으로 말미암아, 일반적인 퍼지 근사 이론에 대한 실공정으로서의 적용은 얼마간 제한되고 있다.

본 논문에서는 고급 자기 구성 뉴럴네트워크(Self-Organizing Neural Networks : SONN)라 불리는 새로운 뉴로퍼지 형태를 이용한 새로운 모델 구조를 설계한다. 제안된 SONN은 확장된 GMDH 알고리즘과 퍼지 추론시스템의 융합으로부터 얻어지는 네트워크로, 각각의 노드 즉, 퍼지 다항식 뉴론(Fuzzy Polynomial Neuron : FPN)은 퍼지 추론 시스템으로써 동작한다. 규칙 후반부에서는 간략 및 선형, 2차, 변형된 2차와 같은 회귀 다항식의 몇 가지 형태를 이용하여 시스템 환경에 보다 유연하게 적용할 수 있다. 각 입력 변수의 전반부 퍼지 규칙은 삼각형 또는 가우시안형으로 멤버쉽 함수를 정의하였다. 제안한 SONN 구조는 크게 기본 구조와 변형된 구조로 나뉘어지며 각 공정의 시스템 입·출력 변수에 따라서 일반적(기본) 형태(Generic type)와 고급형태(Advanced type)로 더욱 세분화되어 생성과정과 설계 방법을 토의한다. 이와같은 SONN 구조의 다양성으로부터 시스템 환경에 유연한 적응력을 가진 최적설계를 도모한다. 성능 평가를 위해서 Box와 Jenkins가 사용한 가스로 입출력 데이터[15]와 비선형 시스템 데이터[16-19]를 이용하여 제안된 모델 구조의 타당성과 정확성을 시뮬레이션을 통해 확인하고, 근사화와 일반화 능력사이의 균형을 고려한 구조선택을 위하여 안정도 측도에 대해 토의한다.

2. SONN 알고리즘과 모델구조

이 절에서는 SONN의 기본 처리 유니트으로써 퍼지 다항식 뉴론(FPN)을 소개하고, 이를 바탕으로 구성된 SONN의 알고리즘과 구조에 대해서 자세히 설명된다. 그리고 이 네트워크의 학습구조에 의한 단계별 알고리즘이 상세히 소개되고, 그 네트워크 내에서 생성되는 다양한 구조에 대해 조사한다.

2.1. 퍼지 다항식 뉴론(Fuzzy Polynomial Neuron)

이 절에서 소개된 FPN은 제안한 SONN의 가장 기본이 되는 처리 단위의 일반적인 형태으로써, 퍼지 집합과 뉴럴 네트워크의 개념에 기초해서 자세히 설명된다. 그림 1에서 보여진 것처럼 FPN은 2개의 기본적인 기능 모듈로 구성되어 있다 (여기서 그림 1은 예를 들어 2입력이고 멤버쉽 함수의 수는 2x2인 경우이다). 처음 하나는 "F"로 표기되어진 것으로, 이는 퍼지 집합의 집합체로써 입력 수의 변수들과 뉴론에 의해 실현된 처리 부분사이를 상호 연결해 준다. 그림에서 x_p 와 x_q 를 입력변수들로 표시한다. "P"로 표기된 두 번째 모듈은 함수 기반 비선형(다항식) 처리에 관한 것이다. 이 비선형 처리는 어떤 입력 변수들(x_i 와 x_j)을 포함한다. 특별히, 비선형성의 다항식 형태인 퍼지 다항식 처리 유니트를 사용한다. 특히, 사용되는 다항식에는 상수 그리고 다항식의 특별한 경우로써의 선형 변환(linear mapping)(규칙 기반 시스템에서 종종 사용되는)을 포함한다.

입력 변수들의 수는 데이터의 특성에 따라 변할 수 있다. 앞서 언급했듯이, 퍼지 집합(여기서는 $\{A_i\}$, $\{B_k\}$)에 의해서 변환되어진 입력 변수들의 집합은 뉴론의 두 번째 다항식 모듈에 의해서 처리되는 변수들과 다르거나, 같거나 그렇지 않으면 부분적으로 중복될 수 있다. 이러한 입력처리 방법은

전체 처리 유니트에 특별한 유연성을 더해준다. 입력변수들을 변환시키기 위해 사용된 퍼지 집합들을 이용하여, FPN은 다중입력 단일출력(MISO) 규칙들을 실현시킨다.

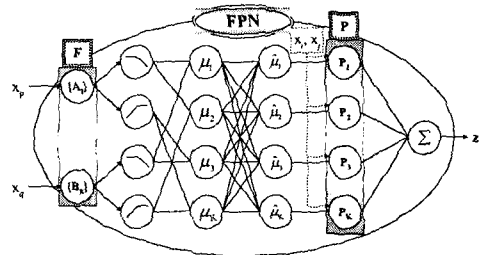


그림 1 FPN 모듈의 일반적인 구조

(F : 퍼지 집합 기반 처리 부분, P : 선형변환의 다항식 형태)

Fig. 1 A general topology of the generic FPN module

(F : fuzzy set-based processing part, P : the polynomial form of mapping)

그림 1에서 언급했듯이, 각 규칙의 형태는 다음과 같다.

$$\text{If } x_p \text{ is } A_i \text{ and } x_q \text{ is } B_k \text{ then } y \text{ is } P_k(x_i, x_j, \mathbf{a}_{ik}) \quad (1)$$

여기서 \mathbf{a}_{ik} 는 규칙 후반부의 파라미터 벡터이다.

단일 FPN 구조에서 수용할 수 있는 규칙의 최대 수는, 각 변수에 대해 정의된 퍼지 집합의 수와 변수 자신들의 수에 의해서 결정된다. FPN의 입력 변수들이, 특히 두 함수 블록과 상호 작용하는 방법을 언급하기 위해, $FPN(x_p, x_q ; x_i, x_j)$ 으로 표기하여 사용한다.

FPN의 처리는 참고문헌 [11], [20]에 나타난 규칙 베이스 컴퓨팅 중에 있는 무게 중심법과 같은 다음 표현식에 의해 연산처리 된다.

$$z = \frac{\sum_{K=1}^{\text{all rules}} \mu_K P_K(x_i, x_j, \mathbf{a}_K)}{\sum_{K=1}^{\text{all rules}} \mu_K} = \sum_{K=1}^{\text{all rules}} \hat{\mu}_K P_K(x_i, x_j, \mathbf{a}_K) \quad (2)$$

여기서, μ_K 는 각 규칙의 활성화 레벨을 표시하고, $K=(l, k)$ 이다.

$$\hat{\mu}_K = \frac{\mu_K}{\sum_{L=1}^{\text{all rules}} \mu_L} \quad (3)$$

2.2. SONN 알고리즘

SONN는 층들의 구성과 노드들 사이의 상호 연결성뿐만 아니라 각각의 노드들의 유연성 때문에 매우 다양한 구조 생성이 가능하다. 결국 이러한 특성들은 네트워크 구조의 유연한 적응성에 기여하지만 보다 신중한 설계 방법론과 학습 메카니즘을 필요로 한다. SONN이 기존의 잘 알려진 뉴로-퍼지 구조와 구별되는 가장 중요한 점은 그 구조의 학습이 네트워크 구조상의 전개에 중요한 역할을 하는 확장된 GMDH 알고리즘을 기반으로 한다는 것이다. 제안된 SONN은 확장된 GMDH 알고리즘과 퍼지추론, 특히 회귀다항식 퍼지 추론 알

그리즘의 상호결합에 의한 학습을 통하여 네트워크 생성을 실현한다. GMDH 알고리즘은 모델링하고자 하는 대상 시스템의 입력변수 x_i 와 x_j , 출력 y 의 쌍에 대해서 2차 회귀방정식을 계산함으로써 이 알고리즘을 시작한다.

$$y = A + Bx_i + Cx_j + Dx_i^2 + Ex_j^2 + Fx_ix_j \quad (4)$$

식 (4)를 사용하여 기존의 m 개의 입력 x_1, x_2, \dots, x_m 에서 출력 y 를 예측하는 2개의 입력을 선택하여 $m(m-1)/2$ 개의 변수 수 즉, 노드의 수를 얻는다[6]. 이 새로운 입-출력 관계로부터 2차 회귀다항식을 찾은 후 이 가운데에서 y 를 가장 잘 예측하는 2차 회귀 다항식 구조만이 판단기준을 통하여 선택한 다음 선택된 2차 회귀 다항식 구조의 결과값들을 연속적인 GMDH 모델의 구축을 위하여 다음 층의 입력으로 사용한다. 이 각각의 2차 회귀다항식을 통한 결과값을 이용하여 새로운 독립적 관찰결과를 얻는다. 다시 말해서, 대수적 대입 법칙을 사용해서 아래의 식과 같은 매우 복잡한 형태의 다항식을 얻을 수 있다.

$$v = a + \sum_{i=1}^m b_i x_i + \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_i x_j + \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^m d_{ijk} x_i x_j x_k + \dots \quad (5)$$

이 식을 Ivakhnenko 다항식이라 한다.

위에서 살펴본 GMDH 알고리즘을 기반으로 한 단계별 상계 SONN 설계 알고리즘을 설명한다.

[단계 1] 시스템 입력변수의 결정

출력변수 y 에 관계하는 n 개의 시스템 입력변수를 결정한다. 필요하다면 데이터를 정규화한다. 이 시스템 입력변수를 x_1, x_2, \dots, x_n 이라 한다.

[단계 2] 데이터 분할

N 개의 입출력 데이터 $(x_i, y_i) = (x_{1i}, x_{2i}, \dots, x_{ni}, y_i), i=1, 2, \dots, N$ 을 학습 데이터와 테스트 데이터로 분할한다. 학습 데이터는 모델을 동정하기 위해 사용하는 데이터이고, 테스트 데이터는 모델을 평가하기 위해 사용하는 데이터이다. 학습 데이터와 테스트 데이터의 개수를 각각 N_L 개와 N_C 개로 한다. 여기서, $N=N_L+N_C$ 이다.

[단계 3] SONN 구조 결정 및 입력변수 수와 퍼지규칙 후반부 다항식 차수 선택

제안된 SONN 알고리즘 구조는 그림 2에서 보여진 것과 같이 다양한 구조 선택을 통하여 주어진 시스템 환경의 변화에 적응할 수 있는 유연성을 가진 모델을 구축할 수가 있다.

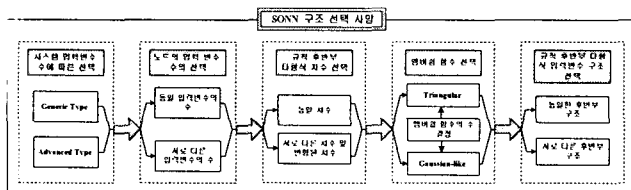


그림 2 SONN 구조에서 이용할 수 있는 설계대안
Fig. 2 The design alternative available in SONN architecture

표 1 전체 SONN 구조

Table 1 A family of the topology of the SONN

층	입력변수 수	규칙 후반부 다항식 차수 및 입력변수 구조	SONN 구조
1층	p	Type P i) 선택된 입력 ii) 전체 시스템 입력	(1) p=q : Basic SONN a) P=Q : CASE I b) P≠Q : CASE II
2층 이상	q	Type Q i) 선택된 입력	(2) p≠q : Modified SONN a) P=Q : CASE I b) P≠Q : CASE II

(p, q=1, 2, 3), (P, Q=1, 2, 3, 4)

입력변수 x_1, x_2, \dots, x_n 을 가지고 선택한 모델에 따라 1층과 2층 이상부터 최종 출력층까지 각 노드의 입력변수 수와 퍼지 추론의 후반부 다항식을 결정한다. 각 층에 사용될 퍼지규칙 후반부 다항식은 다음 표 2와 같다.

표 2 구조퍼지규칙 후반부에서 회귀다항식의 서로 다른 형태

Table 2 Different forms of the regression polynomials standing in the consequence part of the fuzzy rules

다항식 Type(차수)	입력수	1	2	3
0(Type 1)		Constant		
1(Type 2)		Linear	Bilinear	Trilinear
2(Type 3)			Biquadratic	Triquadratic
2(Type 4)		Quadratic	Modified Biquadratic	Modified Triquadratic

[단계 4] 퍼지추론 및 매개변수 추정을 위한 퍼지동정

입력변수 x_1, x_2, \dots, x_n 중 각 노드의 선택된 입력에 대해 퍼지 추론을 적용한다.

선택된 입력의 개수에 따라 총 노드의 개수는 $n!/((n-r)!r!)$ 과 같이 정의된다(n :전체 입력의 수, r :선택한 입력의 수). 퍼지 모델의 전반부 멤버십 함수는 [단계 3]에서 결정된 삼각형 혹은 가우시안형으로 나누어서 정의하고, 퍼지 추론 방법인 간략 추론 및 회귀다항식 추론의 후반부 구조는 표 2에서 표현한 것과 같이 상수항 그리고 1차, 2차, 변형된 2차 다항식인 Type 1~4를 사용한다. 퍼지 추론 규칙과 파라미터 동정을 위한 매개변수 추정은 다음과 같다.

i) 간략 추론

후반부가 단일의 상수항만을 가지는 것으로 이와 같은 추론법을 간략 추론법이라 한다.

이 모델은 아래와 같은 형태를 가지는 구현 규칙들로 구성된다.

$$R^j: \text{If } x_1 \text{ is } A_n \text{ and } \dots \text{ and } x_n \text{ is } A_m \text{ then } y_j \text{ is } a_0 \quad (6)$$

여기서, n : 입력변수의 수, R^j 는 $j(j=1, \dots, z)$ 번째 규칙, $x_l(l=1, \dots, n)$ 는 입력변수, j : 규칙 번호, $A_{jl}(j=1, \dots, z; l=1, \dots, n)$ 는 퍼지 집합의 멤버십 함수, a_0 는 상수이고, z 는 규칙수이다. i 번째 데이터의 추론된 값 \hat{y}_i 는 다음과 같다.

$$\hat{y}_i = \frac{\sum_{j=1}^z w_{ji} y_j}{\sum_{j=1}^z w_{ji}} = \frac{\sum_{j=1}^z w_{ji} a_0}{\sum_{j=1}^z w_{ji}} = \sum_{j=1}^z \hat{w}_{ji} a_0 \quad (7)$$

$$w_{ji} = A_n(x_{1i}) \wedge \dots \wedge A_m(x_{ni})$$

여기서, \hat{w}_{ij} 는 w_{ij} 의 정규화된 값이고, i 는 $i(i=1, \dots, m)$ 번째 데이터, m 은 데이터 총 수이다.

후반부 파라미터 a_0 의 동정은 입출력 데이터가 주어졌을 때 최소자승법(Least square method)에 의해 구해진다. 최소자승법에 의한 후반부 파라미터의 동정은 다음 식과 같다.

$$a = (X^T X)^{-1} X^T Y \quad (8)$$

$$X = [x_1, x_2, \dots, x_m]^T, \quad x_i^T = [\hat{w}_{1i}, \hat{w}_{2i}, \dots, \hat{w}_{zi}],$$

$$Y = [y_1 \dots y_m]^T, \quad a = [a_{10} \dots a_{20}]^T$$

ii) 회귀다항식 추론

후반부가 1, 2, 변형된 2차 다항식 형태로 표현된 것으로 회귀다항식 추론이라 한다. 이 경우의 퍼지 모델은 다음과 같은 형태를 가지는 구현 규칙들로 구성된다.

R^l : If x_1 is A_{j1} and ... and x_n is A_{jn} then $y_j = f_j(x_1, \dots, x_n)$ (9)

여기서, n, R^l, x_i, j, A_{ji} 그리고 z 는 간략추론과 같고, $f_z(x_1, \dots, x_n)$ 은 표 2에서 나타낸 다항식이다.

예로 후반부 다항식이 1차식인 경우의 i 번째 데이터의 추론된 값 \hat{y}_i 는 다음과 같다.

$$\hat{y}_i = \frac{\sum_{j=1}^z w_{ji} y_j}{\sum_{j=1}^z w_{ji}} = \frac{\sum_{j=1}^z w_{ji} (a_{j0} + a_{j1} x_1 + \dots + a_{jn} x_n)}{\sum_{j=1}^z w_{ji}} \quad (10)$$

$$= \sum_{j=1}^z \hat{w}_{ji} (a_{j0} + a_{j1} x_1 + \dots + a_{jn} x_n)$$

여기서 i 는 $i(1, \dots, m)$ 번째 데이터, $a_{jl}(j=1, \dots, z; l=0, \dots, n)$ 는 후반부 파라미터이고 w_{ji} 는 식 (7)과 같다.

후반부 파라미터는 최소자승법에 의해 다음 식으로 구해진다.

$$a = (X^T X)^{-1} X^T Y \quad (11)$$

$$X = [x_1, x_2, \dots, x_m]^T, \quad Y = [y_1 \dots y_m]^T,$$

$$x_i^T = [\hat{w}_{1i}, \hat{w}_{2i}, \dots, x_{1i}, \hat{w}_{1i}, \dots, x_{1i}, \hat{w}_{2i}, \dots, x_{mi}, \hat{w}_{1i}, \dots, x_{mi}, \hat{w}_{zi}],$$

$$a = [a_{10} \ a_{20} \dots \ a_{z0} \ a_{11} \dots \ a_{z1} \dots \ a_{1n} \dots \ a_{zn}]^T$$

[단계 5] 모델 평가 및 노드의 선택

선형 회귀분석에는 학습 데이터를 사용하고, 기존의 연구 결과와 비교하기 위하여 성능지수 즉 동정오차는, 가스로 긍정[15]과 비선형 시스템 데이터[16-19]인 경우엔 MSE(Mean Squared Error)를 사용한 식 (12)와 같다.

$$E = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (12)$$

여기서 N 은 데이터 총 수 이다.

추정된 후반부 다항식에 테스트 데이터를 사용하여 식 (12)의 동정오차 E 를 계산한다.

현재 층에서 얻어진 $n!/((n-r)!r!)$ 개의 동정오차들 $E_1, E_2, \dots, E_{n!/((n-r)!r!)}$ 에 대해서 값이 작은 순으로 배열한다.

1층에서 얻어진 노드 수는 $n!/((n-r)!r!)$ 개의 모두를 사용하며, 2층 이상부터는 선택한 임의의 노드 수 W 로부터,

i) 노드 수가 $n!/((n-r)!r!) > W$ 일 경우, 다음 층의 노드 수

를 W 개로 설정하여, E_1, E_2, \dots, E_W 와 같이 재배열하고 나머지 $n!/((n-r)!r!) - W$ 개의 모델은 버린다.

ii) 노드 수가 $E_1, E_2, \dots, E_{n!/((n-r)!r!)} \leq W$ 일 경우, 다음 층의 노드 수를 $n!/((n-r)!r!)$ 개로 설정하여 $E_1, E_2, \dots, E_{n!/((n-r)!r!)}$ 과 같이 재배열한다.

위와 같은 알고리즘을 바탕으로 2층 이상부터의 노드 수를 결정하는 이유는 노드 수가 많을 경우 프로그램의 과도한 연산시간 및 지나친 메모리 용량등의 문제점을 야기한다. W 값을 너무 작게 설정하면 충분한 연산을 수행하지 못하기 때문에 해당 층에서 우수한 노드 출력의 결과값을 얻지 못한다. 따라서 해당 층에서 적절한 노드 수 선택에 의한 최적 노드 출력을 얻기 위해 적당한 W 의 값을 설정한다.

[단계 6] 종료판정

[단계 5]의 현재 층에서 얻어진 최소 동정오차 E_1 이 다음의 부등식을 만족하는 경우에 알고리즘을 종료한다.

$$E_1 \geq E. \quad (13)$$

단, $E.$ 는 이전 층의 최소 동정오차이다.

본 논문에서는 알고리즘의 판정을 위의 경우와 병행하여 전체의 층 수를 최대 5층으로 제한하여 네트워크가 너무 복잡하지도 않음과 동시에 모델의 복잡도에 비해 성능이 우수한 모델을 구축하여 알고리즘을 종료한다.

[단계 7] 다음 층의 새로운 입력변수 설정

현재 층에서 보존된 퍼지 다항식 뉴런(FPN)의 출력들($z_{1i}, z_{2i}, \dots, z_{wi}$)에서 다음 층의 새로운 입력들($x_{1j}, x_{2j}, \dots, x_{wj}$)로, $x_{1j} = z_{1i}, x_{2j} = z_{2i}, \dots, x_{wj} = z_{wi}(j=i+1)$ 으로 구성되고, [단계 4]로 간다. 이 후, [단계 4]부터 [단계 7]까지를 반복한다. 알고리즘이 종료할 경우, 최소 동정오차 $E.$ 을 얻은 퍼지 다항식 뉴런의 퍼지규칙에 전 층의 출력(중간변수 값)을 대입하고 동일한 조작을 제 1층까지 반복하여 최종 추정 모델 \hat{y} 를 얻는다.

3. SONN 모델 구조

본 논문에서 제안한 SONN 모델 구조는 앞 절에서 살펴본 알고리즘에서 설명한 것 같이 체계적이고 효율적인 구조를 설계함으로써 주어진 시스템 환경에 유연히 적용할 수 있다. 그림 3은 제안한 SONN 구조의 선택을 나타낸 것이다.

TYPE I - Generic type : 전체 시스템 입력변수의 수가 4 입력 이상인 경우

TYPE II - Advanced type : 전체 시스템 입력변수의 수가 3입력 이하인 경우

구조 I - 기본 자기 구성 뉴럴네트워크 : FPN 노드에서 퍼지 규칙의 입력변수의 수가 네트워크의 각 층에서 같은 경우

구조 II - 변형된 자기 구성 뉴럴네트워크 : FPN 노드에서 퍼지 규칙의 입력변수의 수가 네트워크 1층과 2층 이상에서 서로 다른 경우

CASE I - 각 층의 노드 규칙 후반부에서 동일 차수를 가지는 경우 : 표 1과 같이 1층과 2층 이상에서 서로 동일 차수를 사용하는 경우

CASE II - 2층 이상의 노드 규칙 후반부에서 1층과 비교

서 서로 다른 차수 및 변형된 차수를 가지는 경우 : 표 1과 같이 1층과 2층 이상에서 상호 서로 다른 차수 및 변형된 차수를 사용하는 경우

- (A) 모든 층에서 각 노드의 규칙 후반부는 선택된 입력을 사용한 경우
- (B) 각 노드의 규칙 후반부는 1층에서 전체 시스템 입력을 사용하고 2층 이상부터 선택된 입력을 사용한 경우

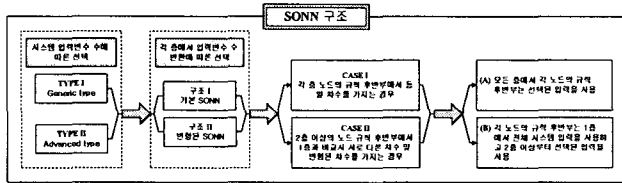


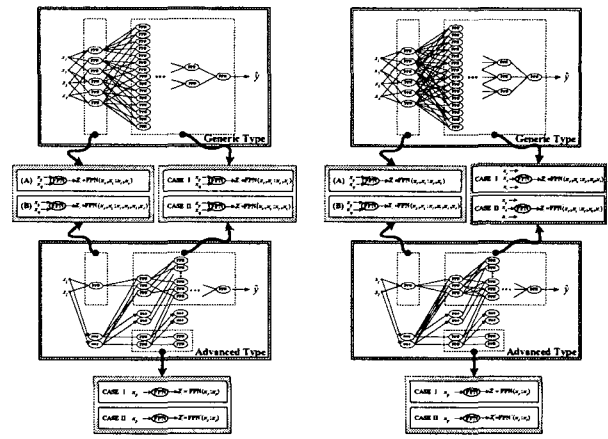
그림 3 SONN 구조 분류
Fig. 3 Taxonomy of SONN architecture

그림 4는 SONN의 기본 구조와 변형된 구조에 대한 설계를 나타낸 것이다. 그림 3에서 보였듯이 먼저 시스템 입력변수의 수에 따라 'TYPE I-Generiс type'과 'TYPE II-Advnaced type'으로 나뉘어지고, 차수의 선택에 있어서 CASE I-각 층 노드의 규칙 후반부에서 동일 차수를 가지는 경우'와 'CASE II-2층 이상의 노드 규칙 후반부에서 1층과 비교시 서로 다른 차수 및 변형된 차수를 가지는 경우'에 대한 설계를 그림 4와 같이 나타낼 수 있다.

또한 각 노드 즉 FPN의 규칙 후반부 다항식 입력변수 구조에 따라 '(A) 모든 층에서 각 노드의 규칙 후반부는 선택된 입력을 사용'한 경우와 '(B) 각 노드의 규칙 후반부는 1층에서 전체 시스템 입력을 사용하고 2층이상부터 선택된 입력을 사용'한 경우로 나누어 모델을 구축하였다.

여기서 2층 이상에서의 Z_i (CASE I)와 Z_i' (CASE II)는 각 노드 즉 퍼지 다항식 뉴런의 규칙 후반부 다항식 차수가 1층에서의 (A)의 Z_i 와 (B)의 Z_i' 와 각각 비교했을 때, CASE I인 경우엔 전 층이 같은 차수로 표기되고 CASE II인 경우엔 1층과 2층 이상에서의 차수가 다르거나 변형된 차수로 표기된다. 또한 그림 4에서 FPN(·)내의 전자는 전방부 입력변수(x_p, x_q)를, 후자는 퍼지 규칙의 후반부 회귀다항식에서의 입력변수(x_p, x_q 또는 x_1, x_2, x_3, x_4)를 각각 나타낸다. 특히, 간략 퍼지 추론 경우엔 FPN(·)의 후자는 변수들 대신 상수가 된다.

그림 4에서, 노드 NOP 은 No operation의 의미로써 FPN 뉴런과는 달리 아무런 연산을 수행하지 않고 단지 대응되는 이전 층의 노드를 현재 층에 다시 사용하여 모델을 구축하는 역할을 한다. 이와 같은 노드를 사용하는 이유는 모델을 구축하기 위해서는 각 층에 어느 정도의 노드가 필요한데 시스템의 입력변수가 적을 경우엔 생성되는 FPN 노드들이 적기 때문에 모델을 구축할 수가 없다. 이러한 단점을 극복하고자 본 논문에서는 NOP 노드를 사용하여 모델을 구축하였다. 네트워크 각 노드의 연결선 중 \rightarrow 은 이전 층에서 대응되는 동일한 노드가 현재 층으로 이동하는 것을 나타낸다.



(a) Basic architecture (b) Modified architecture
그림 4 SONN 구조의 구성
Fig. 4 Configuration of the SONN architecture

4. 시뮬레이션 및 결과고찰

본 논문에서 제안한 SONN 알고리즘을 다음과 같이 단계별로 설계하여 가스로 공정의 시계열 데이터[15]와 비선형 시스템 데이터[16-19]를 적용하여 모델의 타당성 및 유용성을 평가한다. 전체 데이터 집합을 이용하여 학습과 테스트 데이터로 분할하여 모델의 근사화와 일반화(예측) 능력을 상호 평가하며, 이들 성능에 의한 다양한 모델 구조의 유용성 평가를 위해 안정도 측도를 토의한다.

4.1. 가스로 공정

SONN 알고리즘을 이용한 각 단계별 설계절차를 Box와 Jenkins의 가스로 시계열 입출력 데이터[15]에 적용한다.

[단계 1] 시스템 입력변수의 결정

시스템 입력변수의 결정은 다음 표 3과 같다

표 3 시스템 입력 변수
Table 3 System input variables

시스템 입력 형태	입·출력 관계
SI(System Input)-I	$\{u(t-3), y(t-1); y(t)\}$
SI(System Input)-II	$\{u(t-2), y(t-2), y(t-1); y(t)\}$
SI(System Input)-III	$\{u(t-2), u(t-1), y(t-2), y(t-1); y(t)\}$
SI(System Input)-IV	$\{u(t-3), u(t-2), u(t-1), y(t-3), y(t-2), y(t-1); y(t)\}$

여기서, $u(t)$ 는 가스 흐름율이고 $y(t)$ 는 연소된 이산화탄소의 농도이다.

[단계 2] 데이터 분할

총 296개의 데이터 중 처음 148개의 데이터는 학습에 이용하고 나머지 148개의 데이터는 테스트에 이용한다.

[단계 3] SONN 구조 결정 및 입력변수 수와 퍼지규칙 후반부 다항식 차수 선택

SONN 구조는 그림 4와 같이 다양한 SONN 구조로부터 설계된다.

[단계 1]에서 설정한 전체 입력변수와 선택한 모델 구조에 따라 1층과 2층 이상부터 최종 출력층까지 각 노드의 입력변수 수와 퍼지 추론의 후반부 다항식을 결정한다.

각 층에 사용될 후반부 다항식은 다음 표 2와 같다.

2입력과 3입력 모두에서 Type 1은 상수항, Type 2는 선형 다항식, Type 3는 2차 다항식, Type 4는 변형된 2차 다항식을 나타낸다.

[단계 4] 퍼지추론 및 매개변수 추정을 위한 퍼지 동정

[단계 3]에서 선택된 입력변수 수와 후반부 다항식 형태에 따른 퍼지 추론규칙으로부터 학습용 데이터를 사용하여 최소자승법에 의한 규칙 후반부 파라미터를 구한다.

[단계 5] 모델 평가 및 노드의 선택

학습 데이터를 사용하여 모델의 동정오차 즉 성능지수 (Performance Index)를 식 (12)의 MSE(Mean Squared Error)를 사용하여 구한다. 또한 얻어진 모델로부터 테스트 데이터를 사용하여 동정오차 E 를 계산하여 모델을 평가한다.

현재 층에서 얻어진 $n!/((n-r)!r!)$ 개의 $E_1, E_2, \dots, E_{n!/((n-r)!r!)}$ 에 대해서 값이 작은 순으로 배열한다.

1층 노드의 수는 $n!/((n-r)!r!)$ 이며, 2층부터 최종 출력층까지 각 층 노드의 수 즉, W 는 최대 30개로 제한하였으며, 만약 노드의 수가 30개보다 적을 경우엔 $n!/((n-r)!r!)$ 개를 그대로 적용한다.

[단계 6] 종료판정

제안한 SONN 알고리즘은 식 (13)의 종료판정 기준과 병행하여 전체 층 수를 5층으로 제한하여 알고리즘을 수행한다. 따라서, 프로그램 실행시 소요되는 시간과 계산량을 단축할 수 있고, 제한된 전체 층 수로부터 복잡한 구조 생성을 방지할 수 있다.

[단계 7] 다음 층의 새로운 입력변수 설정

현재 층에서 보존된 FPN의 출력값을 다음 층의 입력값으로 하는 새로운 인출력관계를 구성하고, 2층부터 최종 출력층까지는 [단계 4]부터 [단계 7]까지를 반복한다. 알고리즘이 종료할 경우, 동정오차 E 을 얻은 퍼지 다항식 뉴런의 퍼지 규칙에 전 층의 출력(중간변수값)을 입력값으로 대입하고 동일한 조작을 제 1층까지 반복하여 최종 추정 모델 \hat{y} 를 얻는다.

본 공정에서는 표 3에서 나타났듯이 4가지의 입·출력 변수 관계에서, SI-I와 SI-II는 전체 구조에서 TYPE II인 고급 형태(Advanced type)만을 적용하였고, SI-III와 SI-IV는 전체 구조에서 TYPE I인 일반적인 형태(Generic type)만을 적용하였다. 시뮬레이션에 대한 결과 고찰은 TYPE II의 SI-I에 대해서 살펴본다.

SI-I인 경우 그림 3에서 마지막 선택 사항으로써 경우(A)와 경우(B)중에서 시스템 입력이 2입력이므로 단지 경우(A)만 고려된다. 아래 보여진 그림 5와 6은 각 구조별 성능지수 비교에서, 멤버십 함수의 선택에 있어서 삼각형 형태보다 가우시안 형태를 이용했을 때 성능지수가 대체로 우수하여 가우시안 형태를 사용한 경우에 대해서만 도시한 것이다. 각 층에서 각 Type별 가장 우수한 평가 성능지수를 가진 최적 생성 노드로 앞 층의 노드 번호가 표기되어 있고, 또한 그 최적 생성 노드의

PI, E_PI값이 그림에 도시되어 있다. 여기서, 각 Type별 최적 생성노드를 괄호 안의 노드 번호로 나타낸다.

구조 I - 기본 자기 구성 뉴럴네트워크

그림 5에서 (a)는 "CASE I - 각 층의 노드 규칙 후반부에서 동일 차수를 가지는 경우"에 대한 동정오차를 나타낸 것이고, (b)는 "CASE II - 2층 이상의 노드 규칙 후반부에서 1층과 비교시 서로 다른 차수 및 변형된 차수를 가지는 경우"에 대한 동정 오차를 나타낸 것이다. CASE I인 경우엔 후반부 다항식이 Type 4를 사용했을 때 PI=0.0163, E_PI=0.2667로써 다른 후반부 다항식 Type을 사용했을 때 보다 성능이 우수하였다. CASE II인 경우엔 1층은 Type 3을, 2층 이상부터는 Type 4를 사용하여 PI=0.0161, E_PI=0.2654의 동정오차를 얻었다.

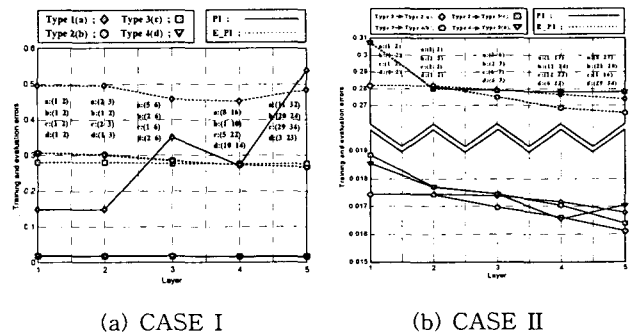


그림 5 구조 I(TYPE II)
Fig. 5 Basic SONN with advanced type

구조 II - 변형된 자기 구성 뉴럴네트워크

구조 II는 구조 I과 달리 각 층에서 서로 다른 입력 변수를 사용한 경우로써 1층에서는 2입력을 사용하고 2층 이상부터는 3입력을 사용하여 모델을 구축하였다. 그림 6에서 (a)는 CASE I에 대한 동정오차를 나타낸 것이고, (b)는 CASE II에 대한 동정 오차를 나타낸 것이다. 이 구조 역시 CASE I과 CASE II의 두경우 모두 멤버십 함수를 가우시안형을 사용했을 때 성능이 우수하였다. CASE I인 경우엔 후반부 다항식이 Type 2를 사용했을 때 PI=0.0133, E_PI=0.2672로써 다른 후반부 다항식 Type을 사용했을 때 보다 성능이 우수하였다. CASE II인 경우엔 1층은 Type 3을, 2층 이상부터는 Type 2를 사용하여 PI=0.0138, E_PI=0.2725의 동정오차를 얻었다.

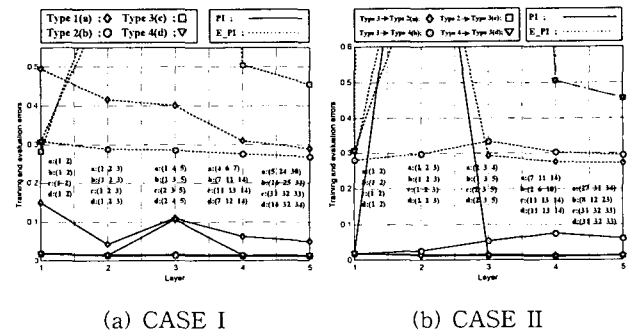


그림 6 구조 II(TYPE II)
Fig. 6 Modified SONN with advanced type

표 4 기존의 퍼지모델과 동정오차비교

Table 4 Comparison of identification errors with conventional fuzzy models

모 델				Performance Index			
				PI	PI _s	E_PI _s	
Tong's model[9]				0.469			
Pedrycz's model[11]				0.320			
Xu's model[12]				0.328			
Park's model[18]				0.055			
Kim's model[19]					0.034	0.244	
Oh and Pedrycz's model[20]				0.123	0.020	0.271	
Sugeno's model[21]				0.068			
Lin and Cunningham's model[22]					0.071	0.261	
Our model	SI-I (2 inputs)	구조 I (Basic)	TYPE II	CASE I	0.1194	0.0163	0.2667
				CASE II	0.1145	0.0161	0.2654
		구조 II (Modified)		CASE I	0.0913	0.0133	0.2672
				CASE II	0.0903	0.0138	0.2725
	SI-II (3 inputs)	구조 I (Basic)		CASE I	0.0409	0.0139	0.1233
				CASE II	0.0479	0.0178	0.1448
		구조 II (Modified)		CASE I	0.0497	0.0219	0.1249
				CASE II	0.0526	0.0203	0.1300
	SI-III (4 inputs)	구조 I (Basic)	TYPE I	CASE I	0.0490	0.0166	0.1161
				CASE II	0.0476	0.0164	0.1283
		구조 II (Modified)		CASE I	0.0570	0.0168	0.1333
				CASE II	0.0599	0.0183	0.1317
	SI-IV (6 inputs)	구조 I (Basic)		CASE I	0.0452	0.0168	0.1168
				CASE II	0.0379	0.0127	0.1255
		구조 II (Modified)		CASE I	0.0369	0.0107	0.1514
				CASE II	0.0402	0.0173	0.1010

표 4는 가스로 공정 데이터를 사용하여 성능 지수를 식 (12)에 의해 계산된 값으로 기존의 모델과 본 논문에서 제시한 SONN 모델과의 성능비교를 한 것이다. 표로부터 제안된 모델은 기존의 모델들에 비해 데이터 특성에 맞는 다양한 모델을 설계함으로써 우수한 성능을 가질 수 있음을 알 수 있다. 또한 표에서 명시된 것 중에서 PI는 전체 데이터를 이용하여 모델의 학습 성능만을 구한 것이다. 전체 데이터를 학습과 테스트 데이터로 나뉘었을 때, 학습 데이터에 대한 성능지수를 PI_s로 표기했으며 테스트 데이터에 대한 성능지수는 E_PI_s로 표기하였다.

4.1.1 안정도 측도

다양한 SONN 모델 중에서 적합한 구조를 선택하는 것에 대한 의문이 증가할 것이다. 분명히, 학습데이터와 테스트 데이터에 대한 모델의 성능들(PI_s, E_PI_s)은 우리가 고려해야 할 두 가지의 중요한 관점들이다. PI_s와 E_PI_s값들은 SONN 모델의 근사화와 일반화 능력을 나타내는 좋은 척도가 된다. 이러한 PI_s와 E_PI_s을 이용해서 안정도 측도를 위한 다음 식을 도입한다.

$$k = \frac{E_PI_s}{PI_s} \tag{14}$$

이 지수는 테스트 데이터에 의해서 모델의 성능이 얼마만큼 저하되는지를 나타낸 모델의 안정도 측도를 나타낸다. 따라서 이 지수가 최소값을 가질 때 그 최소값을 가진 구조를 선택한다. 부가적으로, 만일 SONN 구조의 각 집단에서 k의 값이 그렇게 큰 변화가 없다면, 구조 선택은 그다지 중요하지 않을 수 있다. 특히, 만일 PI_s의 값들이 변화가 심하면 위에

서 제시한 안정도 측도의 유용함이 제한을 받을 것이다. 표 5에서는 표 4에 나타난 PI_s와 E_PI_s에 의해 계산되어진 안정도 측도, k를 나타낸 것이다.

표 5 SONN의 다양한 구조에 대한 안정도 지수

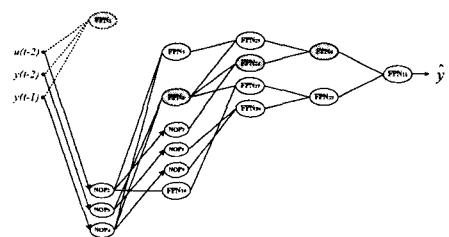
Table 5 Stability index for various architectures of the SONN

SONN Architectures				k	
SI-I	구조 I (Basic)	TYPE II	CASE I	16.361	
			CASE II	16.484	
	구조 II (Modified)		CASE I	20.090	
			CASE II	19.746	
SI-II	구조 I (Basic)		CASE I	8.870	
			CASE II	8.134	
	구조 II (Modified)		CASE I	5.703	
			CASE II	6.403	
SI-III	구조 I (Basic)	TYPE I	CASE I	6.993	
			CASE II	7.823	
	구조 II (Modified)		CASE I	7.934	
			CASE II	7.196	
SI-IV	구조 I (Basic)		CASE I	6.952	
			CASE II	9.881	
	구조 II (Modified)		CASE I	14.149	
			CASE II	5.838	

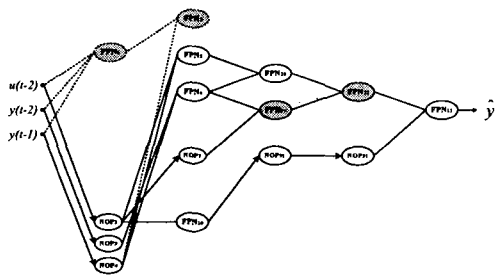
표 5에서 나타난 안정도 측도 지수인 k값에 의해 전체 입력력이 3입력인 경우 즉 SI-II이고 구조 II의 모델을 선택하여 사용했을 때 다른 모델보다 더 안정적이고 효율적인 모델을 구축할 수 있다는 것을 알 수 있다. 이 때 사용된 규칙 후반부 다항식은 CASE I인 경우 Type 4를 사용했으며 PI=0.0219, E_PI=0.1249의 동정오차값을 가지고, CASE II인 경우엔 1층에서는 Type 2를 2층 이상부터는 Type 4를 사용했으며 PI=0.0203, E_PI=0.1300의 동정오차값을 가진다. 두 경우 모두 멤버쉽 함수는 가우시안형을 사용하였다.

그림 7은 이에 대한 각각의 전체 네트워크를 표현한 것이다. 그림에서 PPN_{11} 는 출력 \hat{y} 의 생성을 위해 사용되며, 대응하는 각 층에서 최적인 A번째 노드를 나타낸다. PPN_{12} 는 출력 \hat{y} 의 생성을 위해 사용되며, 대응하는 각 층의 A번째 노드를 나타낸다. PPN_{13} 는 그림 7에서 CASE I의 1층에서처럼 출력 \hat{y} 의 생성을 위해서 사용되지는 않지만, 대응하는 각 층에서 최적인 A번째 노드를 나타낸다.

또한 각 노드를 연결하는 선에 있어서도, —은 출력 \hat{y} 의 생성을 위해 사용되는 선이고, - - -은 출력 \hat{y} 의 생성을 위해 사용되지 않는 선을 나타낸다.



(a) CASE I



(b) CASE II

그림 7 최적 SONN 구조(SI-II, TYPE II, 구조 II, 가우시안형)
Fig. 7 Optimal SONN architecture(SI-II, Advanced, Modified, and Gaussian-like Type)

4.2. 비선형 시스템

식 (15)와 같은 2입력 1출력 관계를 갖는 비선형 시스템 데이터[16-19]를 이용한다.

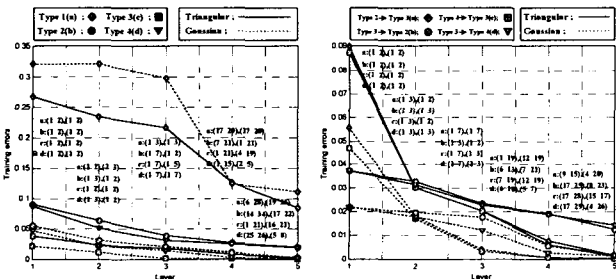
$$y = (1 + x_1^{-2} + x_2^{-1.5})^2, \quad 1 \leq x_1, \quad x_2 \leq 5 \quad (15)$$

식 (15)와 같은 비선형 함수는 참고문헌 [16-19]에서 사용한 것처럼 퍼지 모델의 성능을 평가하는데 널리 사용되고 있다. 비선형 시스템의 입출력 관계식은 식 (15)에서도 알 수 있듯이 입력변수가 적으므로 TYPE II인 Advanced type 그리고 경우(A)의 구조를 이용한다. 그림 8은 각 층에서 각 다항식 Type별 가장 우수한 성능지수를 가진 최적 생성 노드가 앞 층의 노드 번호로 표기되어 있고, 또한 그 최적 생성 노드의 PI 즉 동정오차가 그림에 표시되어 있다. 각 구조에 대한 성능을 평가하면 다음과 같다.

TYPE II - Advanced type

구조 I - 기본 자기 구성 뉴럴네트워크

구조 I인 경우엔 CASE I과 CASE II 모두에서 대체적으로 멤버쉽 함수를 가우시안형을 사용했을 때 성능이 우수하였다. CASE I인 경우엔 후반부 다항식이 Type 3을 사용했을 때 PI=0.00015로 다른 후반부 다항식을 사용했을 때 보다 성능이 우수하였다. CASE II인 경우엔 1층은 Type 2를, 2층 이상부터는 Type 3을 사용하여 PI=0.0002의 동정오차를 얻어 성능이 우수함을 알 수 있었다.



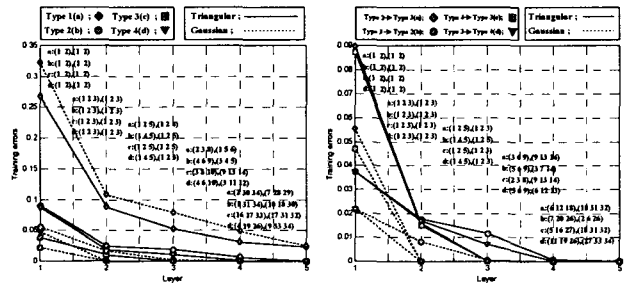
(a) CASE I

(b) CASE II

그림 8 구조 I(TYPE II)
Fig. 8 Basic SONN with advanced type

구조 II - 변형된 자기 구성 뉴럴네트워크

이 구조 역시 CASE I과 CASE II 모두에서 멤버쉽 함수를 가우시안형을 사용했을 때 성능이 우수하였다. CASE I인 경우엔 규칙 후반부 다항식이 Type 3을 사용했을 때 PI=6.89e-20로써 다른 후반부 다항식을 사용했을 때 보다 성능이 우수하였다. CASE II인 경우엔 1층은 Type 2를, 2층 이상부터는 Type 3을 사용했을 때 보다 좋은 동정오차를 얻을 수 있었다.



(a) CASE I

(b) CASE II

그림 9 구조 II(TYPE II)

Fig. 9 Modified SONN with advanced type

표 6 기존의 퍼지모델과 동정오차비교

Table 6 Comparison of identification errors with conventional fuzzy models

모델		PI	
Sugeno and Yasukawa[16]		0.079	
Kim et al.[18]		0.019	
Kim et al.[19]		0.0089	
Gomez-Skarmeta et al.[23]		0.070	
Our model	구조 I (Basic)	CASE I	0.00015
		CASE II	0.00020
	구조 II (Modified)	CASE I	6.89e-20
		CASE II	6.89e-20

표 6은 기존의 모델들과 본 논문에서 제안한 SONN 알고리즘과의 동정오차비교이다. 표에서도 알 수 있듯이 제안한 여러 가지 형태의 알고리즘의 성능이 우수함을 알 수 있다.

그림 10은 비선형 시스템에 대한 최적의 SONN 구조를 나타낸 것이다. 이 경우는 구조 II에서 각 층의 노드 규칙 후반부에서 동일 차수를 가지는 경우인 CASE I을 사용한 경우로써 이때의 후반부 다항식을 Type 3을 사용하고 멤버쉽 함수는 가우시안형을 사용하였다. 이때의 성능지수는 PI=6.89e-20이다.

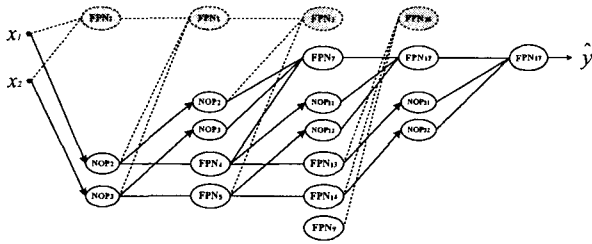


그림 10 최적 SONN 구조(TYPE II, 구조 II, CASE I, 가우시안형)

Fig. 10 Optimal SONN architecture(Advanced, Modified, CASE I, and Gaussian-like Type)

5. 결 론

본 논문에서는 경쟁적 퍼지다항식 뉴런에 기초한 고급 자기 구성 뉴럴네트워크(SONN)에 대한 자세한 설계 절차와 그 특성에 대해서 소개하였다. 특히 데이터 수가 적고 비선형 요소가 많은 시스템에 대한 체계적이고 효율적인 최적 모델을 구축할 수 있었다. 제안된 SONN의 구조는 퍼지 추론 규칙을 기반으로 한 활성노드 즉 퍼지 다항식 뉴런(FPN)들로 구성되어 있다. 규칙의 전반부는 삼각형 및 가우시안형의 멤버쉽 함수를 사용하였고, 규칙의 후반부는 간략 추론과 더불어 1차, 2차 그리고 변형된 2차 방정식과 같은 고계 다항식에 의한 회귀다항식 추론을 이용하였다. 또한 각 층에서 선택된 입력변수의 수가 변화함에 따라 모델을 구축할 수 있으며 더 나아가 일반적인 형태(Generic type)과 고급 형태(Advanced type)으로 나뉘어 모델 구축에 있어서 시스템 입력변수의 수에 따른 한계성을 극복하였다. 또한 주어진 데이터 특성에 맞게 멤버쉽 함수의 형태와 선택된 입력변수의 수에 따른 규칙 후반부의 다항식 차수를 결정하는 동시에 각 층 노드의 선택 입력을 변화시킴으로써 네트워크 구조 전체의 적응능력을 향상시켰다.

SONN의 구조는 일반적인 다층 퍼셉트론 구조의 경우와 달리 미리 정해져 있지 않고 층이 진행되는 동안 모델 스스로 노드의 선택과 제거를 통해 최적의 네트워크 구조를 생성할 수 있는 유연성을 가지고 있다. 이를 통해 제한된 데이터 집합으로도 정교하고 더 정확한 예측 능력을 가진 모델을 구축할 수 있다.

각 공정의 시뮬레이션을 통해서 알 수 있듯이 제안한 모델은 퍼지추론 방법의 선택과 각 층들간의 유기적인 조직으로부터 기존의 다른 퍼지 모델보다 성능이 우수함을 알 수 있었으며 또한 안정도 지수를 이용한 SONN 설계 알고리즘이 모델의 근사화를 위한 overfitting과 일반화 사이의 충돌을 감소시킴으로써 최적 네트워크 구조 선택을 용이하도록 하였다.

향후 수행해야할 연구내용은, 본 연구에서 구축된 SONN 모델은 그 성능을 향상시키는데 크게 영향을 미치는 네트워크 층을 구성하는 노드내의 입력변수의 수와 퍼지규칙 후반부 다항식의 차수 그리고 고정된 입력변수 선택을 설계자의 경험에 의해서가 아니라 최적화 알고리즘을 이용하여 각각의 파라미터들을 선택함으로써 모델을 구축할 필요가 있다.

감사의 글

이 논문은 2003년도 한국학술진흥재단의 지원에 의하여 연구되었음(KRF-2003-002-D00297)

참 고 문 헌

- [1] V. Cherkassky, D. Gehring, and F. Mulier, "Comparison of adaptive methods for function estimation from samples", IEEE Trans. Neural Networks, vol. 7, pp. 969-984, July 1996.
- [2] J. A. Dicherson and B. Kosko, "Fuzzy function approximation with ellipsoidal rules", IEEE Trans. Syst., Man, Cybern. Part B, vol. 26, pp. 542-560, Aug. 1996.
- [3] R. Rovatti and R. Guerrieri, "Fuzzy sets of rules for system identification", IEEE Trans. Fuzzy Syst., vol. 4, pp. 89-102, May 1996.
- [4] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples", IEEE Trans. Syst., Man, Cybern., vol. 22, no. 6, pp. 1414-1427, Nov./Dec. 1992.
- [5] J. H. Nie and T. H. Lee, "Rule-based modeling: Fast construction and optimal manipulation", IEEE Trans. Syst., Man, Cybern. Part A, vol.26, pp. 728-738, Nov. 1996.
- [6] A. G. Ivakhnenko, "The group method of data handling; a rival of method of stochastic approximation", Soviet Automatic Control, 1-3, pp. 43-55, 1968.
- [7] V. Sommer, P. Tobias, D. Kohl, H. Sundgren, and L. Lundstrom, "Neural networks and abductive networks for chemical sensor signals: A case comparison", Sensors and Actuators, B. 28, pp. 217-222, 1995.
- [8] S. Kleinstueber and N. Sepehri, "A polynomial network modeling approach to a class of large-scale hydraulic systems", Computers Elect. Eng. 22, pp. 151-168, 1996.
- [9] R. M. Tong, "The evaluation of fuzzy models derived from experimental data", Fuzzy Sets and Systems., Vol. 13, pp. 1-12, 1980.
- [10] C. W. Xu, "Fuzzy system identification", IEE Proceeding, Vol. 126, No. 4, pp. 146-150, 1989.
- [11] W. Pedrycz, "An identification algorithm in fuzzy relational system", Fuzzy Sets Syst., Vol. 13, pp. 153-167, 1984.
- [12] C. W. Xu, and Y. Zailu, "Fuzzy model identification self-learning for dynamic system", IEEE Trans. on Syst. Man, Cybern., Vol. SMC-17, No.4, pp. 683-689, 1987.
- [13] I. Hayashi and H. Tanaka, "The Fuzzy GMDH algorithm by possibility models and its application", Fuzzy Sets and Systems, Vol. 36, pp. 245-258, 1990.
- [14] Hideo Tanaka, Katsunori and Hisao Ishibuchi, "GMDH

by If-Then Rules with Certainty Factors”, Fifth IFSA World Conference, pp. 802-805, 1993.

[15] Box and Jenkins, “Time Series Analysis, Forecasting and Control”, Holden Day, SanFrancisco, CA, 1976.

[16] M. Sugeno and T. Yasukawa, “A Fuzzy-Logic-Based Approach to Qualitative Modeling”, IEEE Trans. Fuzzy Systems, Vol. 1, No. 1, pp. 7-31, 1993.

[17] H. Nakanishi, I.B. Turksen, M. sugeno, “A review and comparison of six reasoning methods”, Fuzzy sets and Systems, Vol. 57, pp. 257-294, 1992.

[18] E. Kim, M.-K. Park, S.-H. Ji, and M. Park, “A New Approach to Fuzzy Modeling”, IEEE Trans. Fuzzy Systems, Vol. 5, No. 3, pp. 328-337, 1997.

[19] E. Kim, H. Lee, M. Park, M. Park, “A simple identified Sugeno-type fuzzy model via double clustering”, Information Science 110, pp. 25-39, 1998.

[20] S.-K. Oh, and W. Pedrycz, “Identification of Fuzzy Systems by means of an Auto-Tuning Algorithm and Its Application to Nonlinear Systems”, Fuzzy Sets and Systems, Vol. 115, No. 2, pp. 205-230, 2000.

[21] T. Takagi and M. Sugeno, “Fuzzy indetification of systems and its applications to modeling and control”, IEEE Trans Syst. Cybern., Vol. SMC-15, No. 1, pp. 116-132, 1985.

[22] Y. Lin, G. A. Cunningham III, “A new approach to fuzzy-neural modeling”, IEEE Trans. Fuzzy Systems 3, (2), pp. 190-197, 1995.

[23] A. F. Gomez-Skarmeta, M. Delgado and M. A. Vila, “About the use of fuzzy clustering techniques for fuzzy model identification”, Fuzzy Sets and Systems, Vol. 106, pp. 179-188, 1999.

[24] S.-K. Oh and W. Pedrycz, “Fuzzy Polynomial Neuron-Based Self-Organizing Neural Networks”, Int. J. of General Systems, Vol. 32, No. 3, pp. 237-250, 2003

[25] S.-K. Oh and W. Pedrycz, “The Design of Self-organizing Polynomial Neural Networks”, Information sciences, Information Sciences, Vol. 141, Issue 3-4, pp. 237-258, Apr. 2002.

[26] B.-J. Park, W. Pedrycz and S.-K. Oh, “Fuzzy Polynomial Neural Networks: Hybrid Architectures of Fuzzy Modeling”, IEEE Trans. on Fuzzy Systems, Vol. 10, No. 5, pp 607-621, Oct. 2002.

[27] 오성권, “C 프로그래밍에 의한 퍼지모델 및 제어시스템”, 내하출판사, 2002. 1.

[28] 오성권, “프로그래밍에 의한 컴퓨터지능(퍼지, 신경회로망 및 유전자알고리즘을 중심으로)”, 내하출판사, 2002. 8.

저 자 소 개



박 호 성 (朴 浩 成)

1999년 원광대 제어계측공학과 졸업. 동대학원 석사(2001), 2001~현재 동대학 박사과정. 관심분야는 퍼지, 뉴럴네트워크 및 뉴로-퍼지등의 지능형 모델링과 제어. 시스템 자동화 등.

Tel : (063) 850-6342

E-mail : neuron@wonkwang.ac.kr



박 건 준 (朴 建 俊)

2003년 원광대 전기전자공학부 졸업. 2003~현재 동대학 석사과정. 관심분야는 퍼지추론시스템, 신경망, 유전자 알고리즘 및 최적화이론, 컴퓨터 및 인공지능, 자동화시스템 및 제어 등

Tel : (063) 850-6342

E-mail : bird75@wonkwang.ac.kr



이 동 운 (李 東 潤)

1963년 11월 22일 생. 1987년 원광대 공대 전기공학과 졸업. 1990년 연세대 대학원 전기공학과 졸업(석사). 2001년 동대학원전기전자공학과 졸업(공학박). 1990년~2000년 Intellect, J-Tek, KIST 근무. 2001년 3월~2002년 2월 원광대 전기전자 및 정보공학부 B.K. 교수. 현재 중부대학교 정보통신 S/W공학과 조교수.

Tel : (041)750-6209

E-mail : dylee@joongbu.ac.kr



오 성 권 (吳 聖 權)

1981년 연세대학교 전기공학과 졸업. 1983-89년 금성산전연구소(선임연구원). 1993년 연세대 대학원 전기공학과 졸업(공학박). 1996-1997년 캐나다 Manitoba대학 전기 및 컴퓨터공학과 Post-Doc. 1993년~현재 원광대 전기전자 및 정보공학부 부교수. 2002년~현재 대한전기학회 및 제어·자동화·시스템공학회 편집위원, 관심분야는 시스템자동화, 퍼지이론 및 뉴로퍼지네트워크 응용, 계산지능 및 소프트 컴퓨팅 등.

Tel : (063) 850-6342

E-mail : ohsk@wonkwang.ac.kr