

퍼지다항식 뉴론 기반의 유전론적 최적 자기구성 퍼지 다항식 뉴럴네트워크

論 文
53D-8-2

Genetically Optimized Self-Organizing Fuzzy Polynomial Neural Networks Based on Fuzzy Polynomial Neurons

朴 浩 成* · 李 東 潤** · 吳 聖 權***
(Ho-Sung Park · Dong-Yoon Lee · Sung-Kwun Oh)

Abstract - In this paper, we propose a new architecture of Self-Organizing Fuzzy Polynomial Neural Networks (SOPFNN) that is based on a genetically optimized multilayer perceptron with fuzzy polynomial neurons (FPNs) and discuss its comprehensive design methodology involving mechanisms of genetic optimization, especially genetic algorithms (GAs). The proposed SOPFNN gives rise to a structurally optimized structure and comes with a substantial level of flexibility in comparison to the one we encounter in conventional SOPFNNs. The design procedure applied in the construction of each layer of a SOPFNN deals with its structural optimization involving the selection of preferred nodes (or FPNs) with specific local characteristics (such as the number of input variables, the order of the polynomial of the consequent part of fuzzy rules, and a collection of the specific subset of input variables) and addresses specific aspects of parametric optimization. Through the consecutive process of such structural and parametric optimization, an optimized and flexible fuzzy neural network is generated in a dynamic fashion. To evaluate the performance of the genetically optimized SOPFNN, the model is experimented with using two time series data (gas furnace and chaotic time series). A comparative analysis reveals that the proposed SOPFNN exhibits higher accuracy and superb predictive capability in comparison to some previous models available in the literatures.

Key Words: Genetically Optimized Self-Organizing Fuzzy Polynomial Neural Networks (SOPFNN), Fuzzy Polynomial Neuron (FPN), Multi-Layer Perceptron (MLP), Genetic Algorithms (GAs), Group Method of Data Handling (GMDH)

1. 서 론

최근들어, 쉽게 이해할 수 있을 뿐만 아니라 일반화와 근사화 능력을 갖춘 시스템의 모델을 구축하기 위한 연구가 계속 진행되고 있다. 모델링 분야에서 신경회로망, 퍼지 집합 그리고 진화론적 알고리즘은 매우 광범위하게 발전하고 있는 것과 동시에, 시스템 모델링에 대한 새로운 방법론적 이슈가 지속적으로 등장하고 이에 대한 우리의 인식 또한 증가하고 있다[1, 2]. 그러나 모델의 차원이 증가하면(즉, 변수들의 수가 증가한다면), 시스템을 모델링하는데 어렵다. 즉, 비선형 실 시스템의 동적 분석을 표현하는데 있어서의 수학적 모델들은 많은 입·출력 변수들 사이에서 모델을 구축하는 변수들과 모델 구조의 선택에 많은 문제를 가지고 있다.

이러한 문제를 풀기 위해, 시스템의 입력과 출력사이의 비선형 관계를 체계적으로 설계하기 위한 분석적인 기술로 A. G. Ivakhnenko는 Group Method of Data Handling(GMDH) 알고리즘을 소개하였다[3, 4]. 그러나 GMDH는 비교적 간단한 시스템에 대해서 매우 복잡한 다항식을 생성하는 경우가

있으며, 2입력, 2차 회귀다항식의 형태로 복잡한 비선형 시스템을 표현하는데는 어느 정도 한계가 있다. GMDH의 이러한 단점을 극복하고자, Oh 등에 의해 자기구성 다항식 뉴럴 네트워크(Self-Organizing Polynomial Neural Networks : SOPNN)[5, 6]가 개발되었다.

GMDH 알고리즘과 더불어 퍼지 추론 방법, 특히 규칙 베이스 퍼지모델에 의한 모델링과 예측은 동정 방법에 따라 많은 연구가 수행되어 왔다[7, 8]. 그러나 이들 모든 퍼지 동정 알고리즘은 주어진 공정 데이터에 기초함으로써 보다 좋은 성능 즉, 근사화 능력에 초점이 맞추어져있으나, 많은 최적화 기법과 함께 모델화된 특별하고 상세한 규칙들의 제한적 표현능력으로 말미암아, 일반적인 퍼지 근사 이론의 실공정으로의 적용은 유용치 못하다.

이러한 퍼지추론 시스템의 단점은 물론 GMDH의 단점을 극복하기 위해 Oh 등에 의해 SOPNN[5, 6]과 퍼지 추론 시스템을 결합한 자기구성 퍼지 다항식 뉴럴 네트워크(Self-Organizing Fuzzy Polynomial Neural Networks : SOPFNN) [9-12]가 제안되었다. SOPFNN은 데이터 수가 적고 비선형 요소가 많은 시스템에 대한 체계적이고 효율적인 최적 모델을 구축할 수 있다. 그러나 기존 SOPFNN 구조는 성능을 향상시키기 위하여, 노드의 입력변수의 수와 규칙 후반부 다항식 차수 그리고 입력변수는 설계자의 경험 또는 반복적인 학습을 통해 선호된 네트워크 구조를 선택하였으나, 최적의 네트워크 구조를 구축하는데는 어려움이 내재되어 있었다. 따

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

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

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

接 受 日 字 : 2003 年 11 月 21 日

最 終 完 了 : 2004 年 6 月 5 日

라서 최적 네트워크 구조와 그와 관련된 다양한 방법론적 설계가 필요하였다. 즉 모델 구축에 있어서 객관적이고 신뢰성이 있는 최적 모델 구조를 구축하기 위해 노드의 입력변수의 수와 이에 해당되는 입력변수의 선택 그리고 규칙 후반부 다항식 차수의 최적 선택이 요구되어진다.

본 논문에서는 자기구성 퍼지 다항식 뉴럴네트워크(SOFPNN)을 최적화시키기 위해서 최적화 탐색 방법인 유전자 알고리즘[13-15]을 이용하여 자기구성 퍼지 다항식 뉴럴네트워크의 입력변수의 수와 이에 해당되는 입력변수 그리고 규칙 후반부 다항식의 차수를 탐색하여 최적의 자기구성 퍼지 다항식 뉴럴 네트워크를 구축한다. 따라서 모델 구축에 있어서 유연성과 정확성을 가지며 객관적이고 좀 더 정확한 예측 능력을 가진 SOFPNN 모델 구조를 구축할 수가 있다.

본 논문의 평가를 위해서 Box와 Jenkins가 사용한 가스로 입출력 데이터[7-10, 16-28]와 Mackey-Glass 시계열 공정[10, 29-32]을 이용하여 제안된 모델의 타당성, 정확성 및 예측 성능 등을 기존 지능 모델들과 비교를 통해 그 우수성을 보인다.

2. 자기구성 퍼지 다항식 뉴럴 네트워크의 구조와 이의 최적화

2.1. 퍼지 다항식 뉴론(Fuzzy Polynomial Neuron)

이 절에서 소개된 FPN은 SOFPNN의 가장 기본이 되는 처리 단위의 일반적인 형태로써, 그림 1에서 보여진 것처럼 FPN은 2개의 기본적인 기능 모듈로 구성되어 있다(여기서 그림 1은 예를 들어 2입력이고 멤버십 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 입력 변수들로 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.

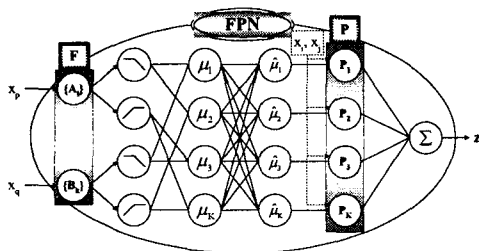


그림 1 FPN 모듈의 일반적인 구조 (F: 퍼지 집합 기반 처리 부분, P: 선형변환의 다항식 형태)

Fig. 1 A general topology of the generic FPN module (F: fuzzy set-based processing part, P: the polynomial form of mapping)

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

그림 1에서 언급했듯이, 각 규칙의 형태는 다음과 같다.

If x_p is A_1 and x_q is B_k then y is $P_k(x_i, x_j, \mathbf{a}_k)$ (1) 여기서 \mathbf{a}_k 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 [8], [23]에 나타난 규칙 베이스 컴퓨팅 중에 있는 무게 중심법과 같은 다음 표현식에 의해 연산처리 된다.

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

여기서, μ_K 는 각 규칙의 활성화 레벨을 표시하고, $K=(1, k)$ 이다. 위 식에서 "K"번째 규칙의 활성화 레벨을 표현하기 위해서 다음과 같이 표준화된 형태로 사용한다.

$$\hat{\mu}_K = \frac{\mu_K}{\sum_{l=1}^{all\ rules} \mu_l}$$
 (3)

표 1에서는 기존의 SOFPNN 각 층에 사용될 FPN의 퍼지 규칙 후반부 다항식을 나타낸다

표 1 퍼지규칙 후반부에서 회귀다항식의 서로 다른 형태 Table 1 Different forms of the regression polynomials standing in the consequence part of the fuzzy rules

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

2.2. SOFPNN의 최적화

유전자 알고리즘(GAs)은 유전학과 자연 진화를 흉내낸 적용 탐색법으로 1975년 Holland[13]에 의해 개발되었다. 유전자 알고리즘은 모든 진화 기반의 탐색 알고리즘 가운데 유연하고 강인한 탐색능력으로 인해 불연속 함수 문제를 포함한 최적화문제를 해결할 수 있는 알고리즘이다[14].

본 논문에서는 유전자 알고리즘을 시뮬레이션하기 위해 설정된 각각의 개체에 존재하는 염색체의 수와 군집의 크기에 대한 파라미터의 정의로써, 한 개체에 여러 변수의 정보를 담

은 직렬 방식을 사용한다. 재생산(Reproduction)연산으로 roulette-wheel 방식, 돌연변이연산은 선택된 비트를 반전시키는 invert 방식, 교차(crossover)연산은 one-crossover 방식을 사용하였다. 그러나 roulette-wheel 방식은 확률적 속성 때문에 선택 과정에서 최적자를 반드시 선택하지 못하는 단점을 가진다. 이를 보완해줄 수 있는 한 방법이 엘리트 전략(Elitist strategy)이다[15].

그림 2는 유전자 알고리즘을 통한 SOFPNN 각 층의 최적화과정을 보여주고 있다.

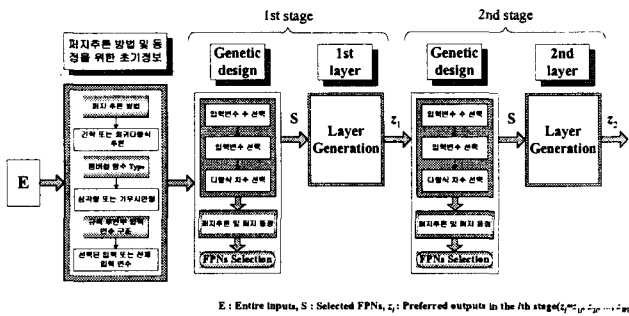


그림 2 유전자 알고리즘을 사용한 SOFPNN 전체 최적화과정
Fig. 2 Overall optimization process of SOFPNN using GAs

본 논문에서는 기존의 자기구성 퍼지 다항식 뉴럴 네트워크가 가지고 있던 단점인 각 노드의 입력변수의 수와 이에 해당하는 입력변수 그리고 다항식 차수의 선택을 최적화 알고리즘인 유전자 알고리즘을 통해 탐색하여 경험자의 경험에 의존한 구조의 최적화를 객관적이고 효율적으로 구축하게 된다.

3. 유전자 알고리즘 기반 자기구성 퍼지 다항식 뉴럴 네트워크 알고리즘과 설계

본 장에서는 기존의 자기구성 퍼지 다항식 뉴럴 네트워크를 유전자 알고리즘을 이용하여 자기구성 퍼지 다항식 뉴럴 네트워크 구조를 최적화하는 알고리즘 및 설계절차를 다룬다. 유전자 알고리즘을 이용한 자기구성 퍼지 다항식 뉴럴 네트워크의 최적화 설계과정은 다음과 같다.

[단계 1] 시스템 입력변수의 결정

출력변수 y 에 관계하는 n 개의 시스템 입력변수를 결정한다. 필요하면 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] SOFPNN 구조 결정

SOFPNN 구조를 구축하는 있어서 기본적인 정보를 결정한다. 즉, SOFPNN의 퍼지 추론 방법 및 동정을 설계하는 부분과 SOFPNN 모델의 구조를 결정하는 초기 정보를 결정한다. 먼저, 퍼지 추론 방법 및 동정 설계를 위한 정보로는 a)

'퍼지 추론 방법 선택 - 간략 또는 회귀 다항식 추론', b) '멤버쉽 함수(MF) 형태 결정 - 삼각형 또는 가우시안', c) '각 입력의 MF 수 선택', d) '규칙 후반부 입력변수 구조 결정 - 선택된 입력 또는 전체 입력변수'가 있으며, SOFPNN 모델 구조를 결정하기 위한 기본적인 정보로는 e) 'SOFPNN 종료 조건의 선택-최적 노드의 적합도 값 비교 또는 최대 층 수 결정', f) '각 층 노드의 입력변수의 수 결정', g) '각 층에 선택될 노드의 수 결정'이 있다.

[단계 4] 유전자 알고리즘을 이용한 SOFPNN 구조 결정 및 FPN의 설계

유전자 알고리즘을 이용한 SOFPNN 구조 결정은 먼저 그림 3에서 보여진다. SOFPNN 모델을 구축하는데 기본 유닛인 FPN 노드들을 유전자 알고리즘에서 각각의 염색체에 의한 구조 선택을 통해 최적 모델을 구축할 수가 있다. 그림 3은 SOFPNN 구조의 각 층의 각 노드에 해당되는 것이다.

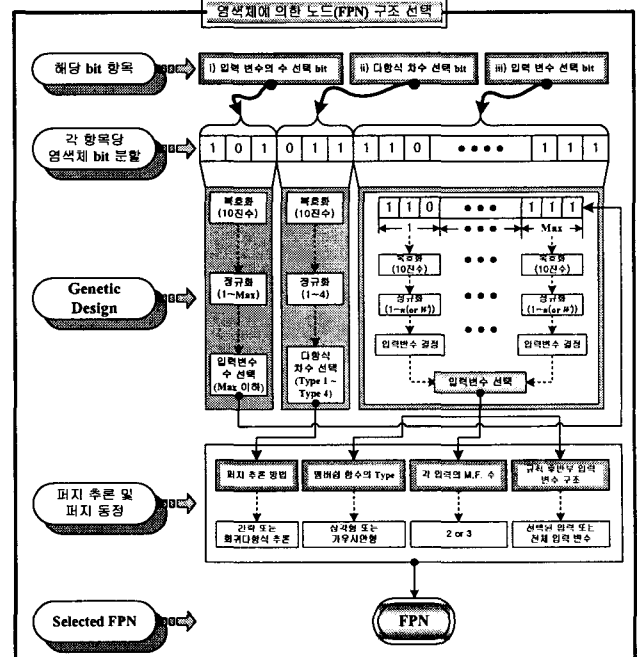


그림 3 유전자 알고리즘의 염색체를 이용한 SOFPNN 구조에서 이용할 수 있는 FPN 설계

Fig. 3 The FPN design available in SOFPNN architecture by using a chromosome of GAs

[단계 4-1] 입력변수의 수 선택

Step 1) 주어진 전체 염색체 중 처음 3비트를 입력변수의 수를 선택하는 비트로 설정한다.

Step 2) 랜덤하게 선택되어진 3비트를 식 (4)을 이용하여 10진수로 복호화 한다.

$$\beta = (2^2 \times \text{bit}(3)) + (2^1 \times \text{bit}(2)) + (2^0 \times \text{bit}(1)) \quad (4)$$

여기서, bit(1), bit(2), bit(3)은 주어진 3비트의 위치를 나타낸다.

Step 3) 복호화된 10진수 β 를 설계자가 정한 최대 입력변수의 수(Max)로 식 (5)을 이용하여 얻은 값을 반올림하여 정규화 한다.

$$\gamma = (\beta / \alpha) \times (\text{Max} - 1) + 1 \quad (5)$$

여기서, Max는 FPN 노드에 들어가는 입력변수의 최대 수이고, a 는 입력변수의 수 선택 비트에 해당되는 비트들이 모두 1로 설정된 경우에 10진수로 복호화한 값이다.

Step 4) 정규화된 값이 하나의 노드를 구축하는 입력변수의 수로 선택되어진다.

여기서, 입력변수의 최대 수(Max)는 전체 입력변수인 x_1, x_2, \dots, x_n 의 n 보다 같거나 작은 수로 설정된다. 즉, $Max \leq n$ 이 된다.

[단계 4-2] 퍼지규칙 후반부 다항식 차수 선택

Step 1) 입력변수의 수 선택 비트 다음의 3비트를 퍼지규칙 후반부 다항식 차수 선택을 위한 비트로 설정한다.

Step 2) 랜덤하게 선택되어진 3비트를 식 (4)을 이용하여 10진수로 복호화 한다.

Step 3) 복호화된 10진수를 식 (5)을 이용(Max 대신 4을 대입)하여 1~4 사이의 값으로 정규화 한다.

1이 선택된 경우 -> Type 1이 사용 즉, 퍼지규칙 후반부 다항식이 간략식으로 구성된다.

2가 선택된 경우 -> Type 2가 사용 즉, 퍼지규칙 후반부 다항식이 선형식으로 구성된다.

3이 선택된 경우 -> Type 3가 사용 즉, 퍼지규칙 후반부 다항식이 2차 다항식으로 구성된다.

4가 선택된 경우 -> Type 4가 사용 즉, 퍼지규칙 후반부 다항식이 변형된 2차 다항식으로 구성된다.

각 Type별 퍼지 규칙 후반부 다항식은 표 1에서 보여준다.

Step 4) 정규화된 값이 하나의 노드를 구축할 때 다항식 차수로 설정된다.

[단계 4-3] 입력변수 선택

Step 1) 위 두 부 염색체(Sub-chromosome) bit를 제외하고 나머지 비트들은 입력변수 선택을 위한 비트로 사용한다.

Step 2) 입력변수 선택을 위한 비트를 [단계 4-1]에서 구한 입력변수의 수로 균등 분할한다. 균등분할이 안될 때에는 다음과 같이 비트를 분할한다. 즉 입력변수 선택을 위한 비트가 22비트이고 입력변수의 수가 4라 하면, 22를 4로 나누어 반올림 처리를 한다. 그러면 첫 번째, 두 번째, 세 번째 입력변수 선택을 위한 비트는 각각 6비트씩 할당을 받고 나머지 비트가 마지막 네 번째 입력변수 선택을 위한 비트로 할당이 된다.

Step 3) 분할된 각 비트들을 식 (4)을 이용하여 10진수들로 각각 복호화 한다.

여기서, 1층에서의 Max값은 전체 입력의 수로 대치하고, 2층 이상부터는 선택된 입력의 수인 W 로 대치한다.

Step 4) 복호화된 각각의 10진수들을 식 (5)을 이용하여 전체 입력변수의 수로 정규화 한다.

Step 5) 정규화된 값들이 하나의 노드를 구성할 때 노드의 입력변수가 된다. 만약 선택된 입력변수가 중복된다면 중복된 입력변수를 하나의 입력변수로 처리한다.

[단계 4-1]부터 [단계 4-3]까지 수행해서 얻은 선택된 입력변수와 퍼지규칙 후반부 다항식 차수를 이용하여 퍼지 추론을 적용한다. 퍼지 모델의 전반부 멤버십 함수는 [단계 3]에서 결정된 삼각형 혹은 가우시안형으로 나누어서 정의하고, [단계 4-2]에서 결정된 퍼지 추론 방법인 간략 추론 및 회귀 다항식 추론의 후반부 구조는 표 1에서 표현한 것과 같이 상수항 그리고 1차, 2차, 변형된 2차 다항식인 Type 1~4를 사

용한다.

기존의 SOFPNN 구조[9-12]에서는 하나의 층에서 모든 노드가 모두 똑같은 입력변수의 수와 다항식 차수를 가지고 모델 구축에 있어서 다양성과 데이터 특성에 맞는 모델을 구축하지 못하는 단점이 있었다. 이에 비해 본 논문에서 제안한 유전자 알고리즘을 이용한 SOFPNN 구조는 위에서 살펴본 그림 3과 같은 염색체 처리를 SOFPNN 전체 층의 모든 노드에 각각 실행함으로써 기존의 SOFPNN 구조의 시행착오적 입력변수의 수와 다항식 차수의 선택에서 벗어나 입력변수의 수와 그에 해당하는 입력변수 그리고 다항식 차수가 유전자 알고리즘에 의해 선택되어 최적화된 모델 구조를 얻는다.

[단계 5] 모델의 구축과 테스트 그리고 노드의 선택

기존의 SOFPNN에서는 모델의 구축은 앞에서 설명하였듯이 일반적인 설계방법이었으나 본 논문에서는 그림 3에서 보여진 것과 같이 유전자 알고리즘을 이용하여 모델의 최적화에 맞게 각 노드들을 유연하게 구축할 수 있다.

각 층의 생성과정으로부터 유전자 알고리즘에 의한 SOFPNN 구조의 전체 최적화 과정은 다음과 같은 절차와 같이 수행된다.

Step 1) SOFPNN 모델 구축을 위한 유전자의 세대, Population 개수, 돌연변이율, 교배율, 그리고 염색체의 길이와 같은 유전자의 정보를 설정한다.

Step 2) 첫 번째 세대에서 설정된 population 개수만큼 염색체에 의한 즉, 최적 다항식 구조 선택을 위해 노드(FPN)을 구축한다. 여기서, 하나의 population은 SOFPNN 구조에서 하나의 노드와 같은 역할을 하며, 각각의 population은 그림 3에서 나타낸 것과 같이 진화론적 최적 설계에 의해 실행한다. 퍼지규칙 후반부 다항식의 매개변수는 최소 자승법을 이용하여 구한다.

Step 3) Population에 의해서 생성된 노드의 성능 테스트를 위해서 기존의 SOFPNN에서와 같이 모델의 테스트 데이터만을 이용하여 노드를 테스트한다. 또한 테스트 데이터를 이용해서 구해진 성능지수를 유전자 알고리즘의 적합도 함수로 계산하기 위해서 다음 식 (6)과 같이 변환하여 유전자 알고리즘의 적합도 함수로 사용한다.

$$F(\text{Fitness Function}) = \frac{1}{1 + EPI} \quad (6)$$

Step 4) 다음 세대의 생성을 위해서 구해진 population 정보와 적합도 값들을 이용하여 선택연산, 교배, 돌연변이를 실행한다.

Step 5) 구해진 적합도 값들(F_1, F_2, \dots, F_z)을 선호되는 적합도 값 순(내림차순)으로 정렬한다.

Step 6) 정렬된 적합도 값들 중에서 중복된 적합도 값들은 제거하고 하나의 적합도 값으로 처리한다. 이는 모델의 각 층을 구축할 때 중복된 노드를 제거하기 위함이다.

Step 7) 중복된 적합도 값들을 하나로 처리한 후 설계자가 SOFPNN 구조를 설계하기전 결정한 노드의 제한된 개수 W 개만큼 선호되는 적합도 값들을 선택한다.

i) 계산된 적합도 함수의 노드 수가 $z \geq W$ 일 경우, 계산된 적합도 함수의 노드 수를 W 개로 설정하고 나머지 $z - W$ 개의 노드는 버린다.

ii) 계산된 적합도 함수의 노드 수가 $z < W$ 일 경우, 적합도 함

수의 노드 수를 z 개로 설정한다.

Step 8) Step 7)에서 선택되어진 적합도 값들을 가진 노드들을 다시 초기 population 번호 순서에 따른 오름차순으로 재정렬한다. 이는 SOFPNN 최종 모델을 구축할 때 관련 층 및 노드들을 체계적으로 구축하기 위함이다.

Step 9) 다음세대에서의 엘리트 전략을 위해서 선택된 노드들 중에서 최적의 적합도 값을 가지는 노드를 선택한다.

Step 10) Step 4)에서 얻어진 정보를 가지고 다음 세대의 population을 생성한 후 Step 2)에서 Step 9)까지 반복한다. 여기서 Step 3)에서 생성된 적합도 값들 중에서 가장 낮은 적합도 값을 가진 노드를 Step 9)에서 구한 이전 세대의 최적의 적합도 값을 가진 노드로 대체하는 엘리트 전략을 실행한다.

Step 11) 이전 세대에서 구해진 적합도 값들과 현재 세대에서 구해진 적합도 값들을 결합하여 다시 Step 5)부터 Step 8)까지 반복한다.

Step 12) 정해진 세대까지 Step 10)부터 Step 11)을 반복 실행한다.

위에서 살펴본 바와 같이, 마지막 세대까지 유전자 알고리즘이 실행되면 최적의 노드들이 생성되며 생성된 노드들은 SOFPNN 모델을 구축하는 하나의 층을 이루게 된다.

[단계 6] 종료판정

[단계 5]의 현재 층에서 얻어진 최적 노드의 최대 적합도 값 F_1 이 다음의 부등식을 만족하는 경우에 알고리즘을 종료한다.

$$F_1 \leq F_* \tag{7}$$

단, F_* 는 이전 층의 최적 노드인 최대 적합도 값이다.

그러나 위와 같은 종료판정은 네트워크 생성 중에 최대 층수가 과대하게 발생할 우려가 있으며 그럴 경우 프로그램 실행 시 많은 시간과 계산량을 요구하게 된다. 또한 현재 층의 동정오차가 이전 층의 동정오차보다 크더라도 그 다음 연속 층에서의 동정오차가 이전 층들의 동정오차보다 더 적은 값이 나올 수 있는데도 프로그램을 종료시키는 단점이 있다. 이와 같은 단점을 보완하기 위해 알고리즘의 판정을 위의 경우와 병행하여 전체 층의 수를 최대 5층으로 제한하여 네트워크의 복잡성과 우수성의 상호 균형을 고려하여 알고리즘을 종료한다.

본 논문에서의 모델의 성능지수식으로 다음 식들을 이용하였다. 가스로 공정과 비선형 데이터인 경우엔 MSE(Mean Squared Error)를 사용해서 식 (8)과 같이 구해지고,

$$E = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2 \tag{8}$$

Predicting Chaotic 시계열인 경우엔 RMSE(Root Mean Squared Error)를 사용해서 식 (9)와 같이 구해진다.

$$E = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2} \tag{9}$$

[단계 7] 다음 층의 새로운 입력변수 설정

현재 층에서 보존된 노드의 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(여기서, $j=i+1$). 이 후, [단계 4]부터 [단계 7]까지를 반복한다. 알고리즘이 종료할 경우, 마지막 층의 최적 적합도 값 F_* 을 얻은 노드의 다항식에 전 층의 출력을 대입하고 동일한 조작을 제 1층까지 반복하여 최종 추정 모델 \hat{y} 를 얻는다.

4. 시뮬레이션 및 결과고찰

4.1. 가스로 공정

제안된 유전자 알고리즘을 이용한 자기구성 퍼지 다항식 뉴럴 네트워크의 성능을 테스트하기 위해서 많은 연구자들이 제안하는 모델링 기법의 성능을 테스트하는데 널리 사용되는 가스로 시계열 데이터를 이용하였다. Box와 Jenkins[16]에 의해 사용된 가스로 시계열 데이터는 입출력 데이터인 가스 흐름율($u(t)$)과 연소된 이산화탄소 농도($y(t)$)의 가스로 공정으로부터 추출된다. 1입력 1출력의 가스로 시계열 데이터 296쌍을 시뮬레이션 하기 위해 전체 시스템 입력을 표 2에서 보여진 것과 같이 설정하여 사용한다. 또한 전체 시스템 입력 데이터 296개 중 처음 148개의 데이터는 학습 데이터로 이용하고 나머지 148개의 데이터는 테스트 데이터로 나누어 자기구성 퍼지 다항식 뉴럴 네트워크를 모델링하고 유전자 알고리즘을 사용하여 모델을 최적화한다. 본 논문에서는 표 2에서 보여진 것처럼 3가지 시스템 입력 형태 Type I, Type II, 그리고 Type III가 고려된다.

표 2 시스템 입력 변수

Table 2 System input vector formats for the design of the SOFPNN

No. of System Inputs(SI)	Inputs and Outputs
2(Type I)	$[u(t-3), y(t-1) ; y(t)]$
4(Type II)	$[u(t-2), u(t-1), y(t-2), y(t-1) ; y(t)]$
6(Type III)	$[u(t-3), u(t-2), u(t-1), y(t-3), y(t-2), y(t-1) ; y(t)]$

표 3은 유전자 알고리즘을 기반으로 자기구성 퍼지 다항식 뉴럴 네트워크의 다항식을 구성하기 위한 유전자 알고리즘의 염색체 정보와 자기구성 다항식 뉴럴 네트워크 구축을 위한 정보를 나타낸 것이다.

표 3 유전자 알고리즘과 자기구성 퍼지 다항식 뉴럴 네트워크 정보

Table 3 Computational overhaed and a list of parameters of the GAs and the SOFPNN

	Parameters	1층	2층~5층
		GA	Maximum generation
	Total population size	60	60
	Selected population size	30	30
	Crossover rate	0.65	0.65
	Mutation rate	0.1	0.1
	String length	3+3+30	3+3+30
SOFPNN	Maximal no. of inputs to be selected(Max)	$1 \leq l \leq \text{Max}(2 \sim 5)$	$1 \leq l \leq \text{Max}(2 \sim 5)$
	Polynomial Type(Type T) of the consequent part of rules	$1 \leq T \leq 4$	$1 \leq T \leq 4$
	Membership function(MF) type	Triangular Gaussian	Triangular Gaussian
	No. of MFs per each input	2	2

l, T : integer

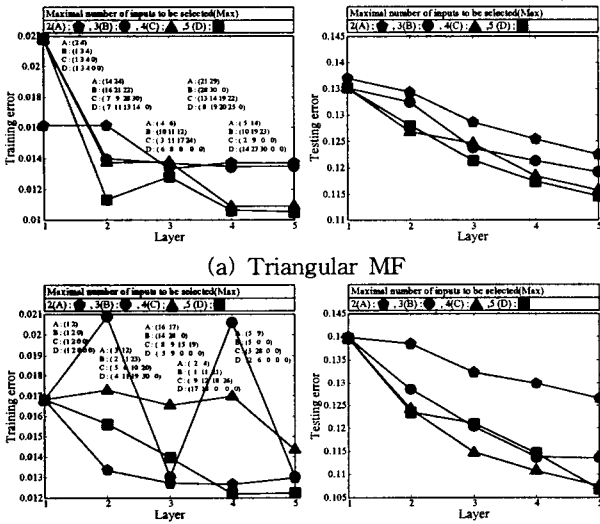
* : 규칙 후반부 다항식의 입력이 전체 시스템 입력이 사용된 경우

표 3에서 보여진 것처럼 초기 정보는 유전자 정보와 SOFPNN 정보의 두 형태로 나눌 수 있다. 유전자 정보중에

서 string length가 3+3+30인 것은 처음 3비트는 입력변수의 수를 선택하기 위한 비트이고, 두 번째 3비트는 퍼지 규칙 후반부 다항식 차수를 선택하기 위한 비트이고, 마지막 30비트는 입력변수를 선택하기 위한 비트를 나타낸 것이다. SOFPNN 정보에서는 멤버십 함수 형태를 삼각형과 가우시안형으로 나누어 모델링하였다.

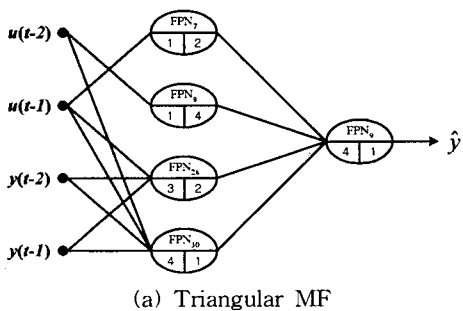
가스로 공정에서의 모델의 특징은, 모델의 근사화와 일반화 사이의 상호 능력을 고려한다면 멤버십 함수를 삼각형을 사용한 것보다는 가우시안형을 사용하는 것이 우수한 모델을 구축하는데 효과적임을 알 수 있다. Max=5에서 가장 최적인 성능지수로는 삼각형 함수를 사용한 경우에는 PI=0.010, EPI=0.114, 가우시안 함수를 사용한 경우에는 PI=0.012, EPI=0.106이다.

그림 4는 선택된 입력의 변화에 따른 SOFPNN 각 층의 성능지수를 나타낸 것이다. 그림에서, A(·)-D(·)는 네트워크의 각 층에서의 최적의 노드 번호들을 나타내고 해당 층에서 최적의 네트워크 출력성능을 가진다. 즉, 그림 4에서 1층에서의 노드 번호는 시스템 입력번호를 나타내고, 2층 이상에서의 노드번호는 현재 층에서 가장 우수한 성능지수를 가지는 최적노드로서 앞선 층의 출력 노드번호를 나타낸다.

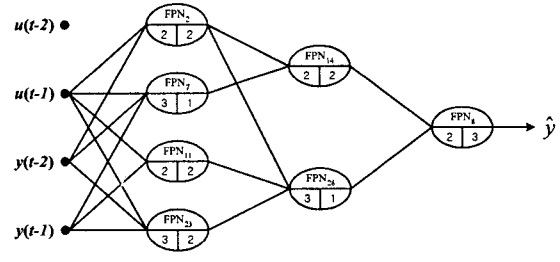


(b) Gaussian-like MF
 그림 4 층의 증가에 따른 Type II에 대한 SOFPNN의 성능지수
 Fig. 4 Performance index of SOFPNN for Type II according to the increase of number of layers

그림 5는 기존 FPNN보다 성능이 우수한 모델(상호 성능지수 값이 비슷한 조건하에서)을 나타낸 것이다.



(a) Triangular MF



(b) Gaussian-like MF

그림 5 진화론적 최적 자기구성 퍼지 다항식 뉴럴 네트워크 구조

Fig. 5 Genetically optimized SOFPNN architecture

즉, 멤버십 함수가 삼각형인 경우, Max=4이고 2층에서의 네트워크를 나타낸 것이며(PI=0.013, EPI=0.126), 가우시안형 함수를 사용한 경우엔 Max=3이고 3층에서의 네트워크를 나타낸 것이다(PI=0.013, EPI=0.120).

그림 5에서 보여준 노드는 다음과 같이 표현된다. $\begin{matrix} \text{FPN}_n \\ \hline \text{NI} \\ \hline \text{T} \end{matrix}$ 여기서, 'FPN_n'는 각 층에서 n번째 FPN 노드를 나타낸 것이고, 'NI'는 해당 노드로 들어오는 입력의 수를, 'T'는 해당 노드에 사용된 규칙 후반부 다항식 차수를 나타낸 것이다. 그림 5에서 알 수 있듯이 기존의 SOFPNN 구조와는 달리 각각의 층을 구성하는 노드들이 기존의 SOFPNN 구조처럼 미리 설계자에 의해서 설정된 네트워크 구축이 아니라 GA의 구조적 동정에 의해 입력변수 수, 입력변수의 선택, 그리고 규칙 후반부 다항식 차수등의 각 파라미터 값을 최적으로 선택하여 네트워크가 구축된다.

그림 6은 그림 5-(a)에서 보여준 네트워크 각 층에서 generation 수행 과정에 따른 Training 성능지수와 Testing 성능지수의 최적화 과정을 나타낸 것이다. 그림 6에서 알 수 있듯이 층이 증가할수록 성능은 개선되지만 계속 층을 증가하면 처리 속도와 계산량 증가와 같은 문제를 발생한다.

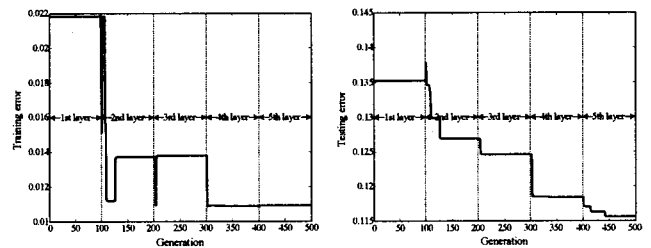


그림 6 유전자 알고리즘에 의한 각 성능지수들의 최적화 과정
 Fig. 6 The optimization process of each performance index by the genetic algorithms

표 4는 가스로 공정 데이터를 사용하여 성능 지수를 식 (8)에 의해 계산된 값으로 기존의 모델과 본 논문에서 제시한 유전자 알고리즘을 이용한 SOFPNN 모델과의 성능비교를 한 것이다. 표 4에서 "PI"는 전체 데이터를 이용하여 모델의 성능지수를 구한 것이고, "PI_s", "EPI_s"는 전체 데이터를 학습 데이터와 테스트 데이터로 나누어 모델의 성능지수 값을 각각 구한 것이다.

표 4 다른 모델링 방법과 성능 비교

Table 4 Comparison of performance with other modeling methods

Model		Performance Index		
		PI	PI _s	EPI _s
Box and Jenkir's model[16]		0.710		
Sugeno and Yasukawa's model[17]		0.355		
Xu and Zailu's model[7]		0.328		
Pedrycz's model[8]		0.320		
Chen's model[18]		0.268		
Gomez-Skarmeta's model[19]		0.157		
Oh and Pedrycz's model[20]		0.123	0.020	0.271
Kim, et al.'s model[21]		0.055		
Kim, et al.'s model[22]			0.034	0.244
Leski and Czogala's model[23]		0.047		
Lin and Cunningham's model[24]			0.071	0.261
NNFS model[25]		0.128		
FPNN[26] (SI=4, 5th layer)	CASE I	0.016	0.116	
	CASE II	0.016	0.128	
PNN[27] (SI=4, 5th layer)	Basic	0.021	0.110	
	Modified	0.015	0.103	
HFPNN[28] (SI=4, 5th layer)	Triangular	0.019	0.134	
	Gaussian	0.021	0.119	
Generic SOPNN[9] (SI=4, 5th layer)	Basic SOPNN	0.027	0.021	0.085
	Modified SOPNN	0.035	0.017	0.095
Advanced SOPNN[10] (SI=4, 5th layer)	Basic SOPNN	0.020	0.119	
	Modified SOPNN	0.018	0.118	
GA-SOPFNN (Our model)	Type I (SI=2)	Triangular 3rd layer(Max=4) 5th layer(Max=5)	0.017	0.271
		Gaussian-like 3rd layer(Max=4) 5th layer(Max=5)	0.014	0.260
	Type II (SI=4)	Triangular 2nd layer(Max=4) 5th layer(Max=5)	0.013	0.126
		Gaussian-like 3rd layer(Max=3) 5th layer(Max=5)	0.013	0.120
	Type III (SI=6)	Triangular 3rd layer(Max=4) 5th layer(Max=5)	0.018	0.122
		Gaussian-like 3rd layer(Max=4) 5th layer(Max=5)	0.020	0.104

4.2. Mackey-Glass 시계열 공정

Mackey-Glass 시계열은 예측 알고리즘의 성능 평가를 위한 대표적인 예로서 사용되는 시계열이다[10 38-41]. 특히 Mackey-Glass 시계열은 비선형 시스템으로부터 발생하는 카오스 신호의 일종으로 신호의 복잡성으로 인해 랜덤 시계열(Random Time Series)처럼 보일 수 있으나 다음 식과 같은 비선형 동력학 방정식에 의해 생성된다.

$$\dot{x}(t) = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \quad (10)$$

본 연구에서는 기존의 방법들과 비교를 위해 미래값 $x(t+6)$ 의 예측 성능을 조사한다. 식 (10)의 수치적 해를 찾기 위한 방법으로 4차 Runge-Kutta 방법을 적용한다. time step 은 0.1, 초기조건 $x(0)$ 는 1.2로, 시간지연상수(Time Delay Constant) τ 는 17로 놓고 $x(t)$ 를 발생시킨다. Mackey-Glass 시계열 $x(t)$ 로부터 $[x(t-18), x(t-12), x(t-6), x(t); x(t+6)]$ 와 같은 4입력 형태의 입력력 데이터 쌍을 구하였다. 여기서 데이터 발생 초기에는 과도영역(Transient Region)이 존재하므로 $t=118$ 에서 $t=1117$ 까지 1000개의 데이터를 추출하였다.

제안된 모델의 학습을 위해 $x(118)$ 부터 $x(617)$ 까지 500개

의 시계열 데이터를 사용하였고, 예측 성능의 평가를 위해 $x(618)$ 부터 $x(1117)$ 까지 500개의 시계열 데이터를 예측치와 비교하였다. 시뮬레이션의 예측 성능을 위해, 식 (9)와 같이 잘 알려진 성능지수인 RMSE(Root Mean Squared Error)를 이용한다. Mackey-Glass 시계열 공정데이터를 유전자 알고리즘 기반 자기구성 다항식 뉴럴 네트워크의 다항식을 구축하기 위해 유전자 알고리즘의 염색체 정보와 자기구성 다항식 뉴럴 네트워크 구축을 위한 정보는 표 3과 같다.

Mackey-Glass 시계열 공정데이터를 제안한 모델을 통하여 얻은 성능지수는, 삼각형과 가우시안형 모두에서 FPN의 선택된 입력과 층이 증가할수록 성능지수가 우수하나 그 증가폭은 둔화됨을 알 수 있다. 또한 모델의 근사화와 일반화 사이의 상호 능력을 고려한다면 멤버쉽 함수를 삼각형을 사용한 것보다는 가우시안형을 사용하는 것이 우수한 모델을 구축하는데 효과적임을 알 수 있다. 가장 선호되는 성능지수로는 삼각형 함수를 사용한 경우엔 Max=5인 경우 PI=0.0009, EPI=0.0010, 가우시안 함수를 사용한 경우에는 Max=5인 경우 PI=0.0003, EPI=0.0005으로 다른 선택된 입력변수의 성능보다 우수함을 알 수 있다.

그림 7은 선택된 입력의 변화에 따른 SOPFNN 각 층의 성능지수를 나타낸 것이다.

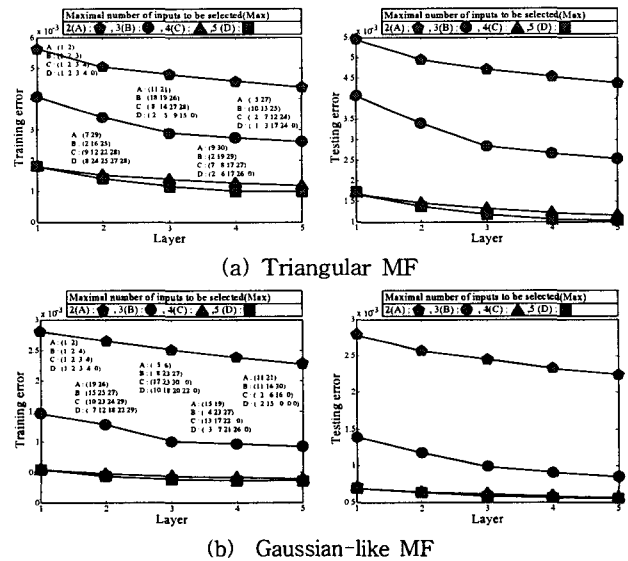


그림 7 층의 증가에 따른 SOPFNN의 성능지수
Fig. 7 Performance index of SOPFNN according to the increase of number of layers

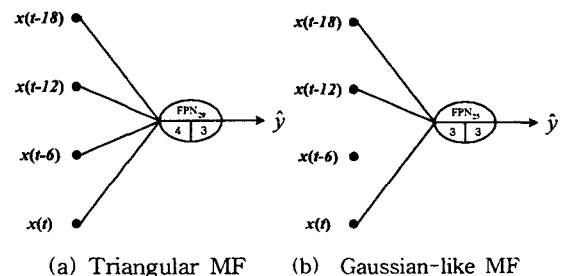


그림 8 진화론적 최적 자기구성 퍼지 다항식 뉴럴 네트워크 구조
Fig. 8 Genetically optimized SOPFNN architecture

그림 8은 비교적 모델이 간단하면서 기존의 SOFPNN보다 성능이 우수한 모델을 나타낸 것이다. 즉, 그림 8-(a)는 멤버쉽 함수가 삼각형인 경우, Max=4이고 1층(PI=0.0017, EPI=0.0016)에서의 네트워크를 나타낸 것이며, 그림 8-(b)는 가우시안 함수를 사용한 경우엔 Max=3이고 1층에서의 네트워크를 나타낸 것이다(PI=0.0014, EPI=0.0013).

그림 9는 Max=4이고 멤버쉽 함수를 가우시안 함수로 사용할 때 네트워크 각 층에서 generation 수행 과정에 따른 Training 성능지수 와 Testing 성능지수의 최적화 과정을 나타낸 것이다. 그림 9에서 알 수 있듯이 3, 4층 부근에서 오차 범위가 줄어드는 것을 알 수 있고, 정해진 오차의 tolerance level에서 최적의 네트워크를 구축 할 수 있음을 알 수 있다.

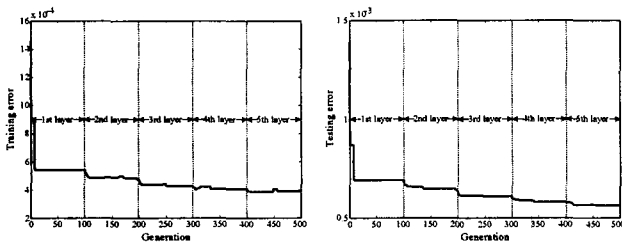


그림 9 유전자 알고리즘에 의한 각 성능지수들의 최적화 과정
Fig. 9 The optimization process of each performance index by the genetic algorithms

표 5는 Mackey-Glass 시계열 공정 데이터를 사용하여 성능 지수를 식 (9)에 의해 계산된 값으로 기존의 모델과 본 논문에서 제시한 유전자 알고리즘을 이용한 SOFPNN 모델과의 성능비교를 한 것이다.

표 5 기존 모델방법과 동정 오차비교
Table 5 Comparison of performance with other modeling methods

Model			Performance Index			
			PI	PI _s	EPI _s	NDEI
Wang's model[29]			0.044			
			0.013			
			0.010			
ANFIS[30]			0.0016	0.0015	0.007	
FNN model[31]			0.014	0.009		
Recurrent neural network[32]			0.0138			
SONN[10]	Basic (5th layer)	Case 1	0.0011	0.0011	0.005	
		Case 2	0.0027	0.0028	0.011	
	Modified (5th layer)	Case 1	0.0012	0.0011	0.005	
		Case 2	0.0038	0.0038	0.016	
GA-SOFPNN (Our model)	Triangular	2nd layer(Max=4)	0.0014	0.0015	0.0014	0.0061
		5th layer(Max=4)	0.0010	0.0011	0.0011	0.0045
	Gaussian-like	2nd layer(Max=5)	0.0011	0.0014	0.0013	0.0051
		5th layer(Max=5)	0.0007	0.0009	0.0010	0.0031
	1st layer(Max=3)	1st layer(Max=4)	0.0005	0.0005	0.0006	0.0023
		1st layer(Max=5)	0.0005	0.0005	0.0006	0.0023

여기서, NDEI(Non-Dimensional Error Index)[33]는 RMSE에 의해 구해진 모델의 출력 데이터가 시스템 출력 데이터의 표준편차에 의해 나눈 것으로 정의된다.

5. 결 론

본 논문에서는 최적화 알고리즘인 유전자 알고리즘을 이용하여 진화론적 최적 자기구성 퍼지 다항식 뉴럴 네트워크의 효율적인 설계 방법을 제안하였고 이를 기반으로 주어진 비선형 예제들을 통하여 GA기반 SOFPNN의 우수성을 입증하였다.

기존의 SOFPNN 알고리즘은 데이터 수가 적고 비선형 요소가 많은 시스템에 대해서 경험적인 방법에 의해 효율적인 가장 선호된 모델을 구축할 수 있었다. 그러나 SOFPNN 구조의 성능을 향상시키는데 크게 영향을 미치는 네트워크 층을 구성하는 노드내의 입력변수의 수와 퍼지규칙 후반부 다항식의 차수 그리고 고정된 입력변수 선택을 설계자의 경험에 의해서 결정함으로써 객관적이고 주어진 데이터의 특성에 맞는 최적의 모델을 구축하지 못하였고, 또한 최적에 근접한 모델을 구축하기 위해서 많은 시행착오를 필요로 하였다. 이러한 SOFPNN 모델의 구조의 한계를 극복하고자 본 연구에서는 최적화 방법인 유전자 알고리즘을 이용하여 노드의 입력변수의 수와 그에 해당하는 입력변수 그리고 퍼지규칙 후반부 다항식 차수 등의 파라미터를 선택하여 모델 구축에 있어서 유연성과 정확성을 가지며 더욱 정확한 예측 능력을 가진 최적화된 SOFPNN 모델 구조를 구축할 수가 있었다.

더욱이 주어진 데이터 특성에 맞게 멤버쉽 함수의 형태 결정, 규칙 후반부 입력변수 구조 결정 그리고 유전자 알고리즘에 의해서 결정된 규칙 후반부의 다항식 차수에 의한 퍼지추론 규칙을 통해서 네트워크 구조 전체의 적응능력을 향상시켰다.

각 공정의 시뮬레이션을 통해서 알 수 있듯이 제안한 모델은 유전자 알고리즘을 이용한 각 노드들의 유기적인 조직으로부터 기존의 다른 모델들보다 성능이 우수함을 알 수 있으며, 데이터 특성이 비선형성이 강하거나 또는 데이터의 수가 적더라도 체계적이고 효율적인 최적의 자기구성 퍼지 다항식 뉴럴 네트워크를 구축함을 보였다.

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. Dickerson and B. Kosko, "Fuzzy function approximation with ellipsoidal rules", IEEE Trans. Syst., Man, Cybernetics, Part B, Vol. 26, pp.542-560, Aug., 1996.
- [3] A. G. Ivakhnenko, "Polynomial theory of complex systems", IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-1, pp. 364-378, 1971.
- [4] A. G. Ivakhnenko and H. R. Madala, "Inductive

- Learning Algorithms for Complex Systems Modeling”, CRC Press, Boca Raton, FL, 1994.
- [5] S.-K. Oh and W. Pedrycz, “The design of self-organizing Polynomial Neural Networks”, *Information Science*, Vol. 141, pp. 237-258, 2002.
- [6] S.-K. Oh, W. Pedrycz and B.-J. Park, “Polynomial Neural Networks Architecture: Analysis and Design”, *Computers and Electrical Engineering*, Vol. 29, Issue 6, pp. 703-725, 2003.
- [7] 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.
- [8] W. Pedrycz, “An identification algorithm in fuzzy relational system”, *Fuzzy Sets Syst.*, Vol. 13, pp.153-167, 1984.
- [9] 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, May, 2003.
- [10] S.-K. Oh, W. Pedrycz and T.-C. Ahn, “Self-organizing neural networks with fuzzy polynomial neurons”, *Applied Soft Computing*, Vol. 2, Issue 1F, pp. 1-10, Aug. 2002.
- [11] S.-K. Oh, W. Pedrycz and H.-S. Park, “Self-organizing Networks in Modeling Experimental Data in Software Engineering”, *IEE Proc.-Computers and Digital Techniques*, Vol. 149, Issue 3, pp. 61-78, May, 2002.
- [12] S.-K. Oh and D.-Y. Lee, “Advanced Self-organizing Neural Networks with Fuzzy Polynomial Neurons: Analysis and Design”, *KIEE International Transactions on Systems and Control(SC)*, Vol. 12D, No. 1, pp. 12-16, March, 2002.
- [13] Holland, J. H., *Adaptation In Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor. 1975.
- [14] D. E. Goldberg, *Genetic Algorithm in search, Optimization & Machine Learning*, Addison wesley, 1989.
- [15] K. De Jong. Are genetic algorithms function optimizers? In *Proc. of PPSN II (Parallel Problem Solving from Nature)*, pages 3-13, Amsterdam, North Holland, 1992.
- [16] D. E. Box and G. M. Jenkins, *Time Series Analysis, Forecasting and Control*, California: Holden Day, 1976.
- [17] M. Sugeno and T. Yasukawa, “Linguistic modeling based on numerical data”, *IFSA 91*, Brussels, Computer, Management & Systems Science, pp. 264-267, 1991.
- [18] J. Q. Chen, Y. G. Xi, and Z.J. Zhang, “A clustering algorithm for fuzzy model identification”, *Fuzzy Sets and Systems*, Vol. 98, pp. 319-329, 1998.
- [19] 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.
- [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] E.-T. Kim, et al, “A new approach to fuzzy modeling”, *IEEE Trans. Fuzzy Systems*, Vol. 5, No. 3, pp. 328-337, 1997.
- [22] E.-T. Kim, et al, “A simple identified Sugeno-type fuzzy model via double clustering”, *Information Science*, Vol. 110, pp. 25-39, 1998.
- [23] J. Leski and E. Czogala, “A new artificial neural networks based fuzzy inference system with moving consequents in if-then rules and selected applications”, *Fuzzy Sets and Systems*, Vol. 108, pp. 289-297, 1999.
- [24] Y. Lin, G. A. Cunningham III, “A new approach to fuzzy-neural modeling”, *IEEE Trans. Fuzzy Systems*, Vol. 3, No. 2, pp. 190-197, 1995.
- [25] Yin Wang and Gang Rong, “A self-organizing neural-network-based fuzzy system”, *Fuzzy Sets and Systems*, Vol. 103, pp. 1-11, 1999.
- [26] H.-S. Park, S.-K. Oh, and Y.-W. Yoon, “A New Modeling Approach to Fuzzy-Neural Networks Architecture”, *Journal of Control, Automation and Systems Engineering*, Vol. 7, No. 8, pp. 664-674, Aug., 2001.
- [27] S.-K. Oh, D.-W. Kim and B.-J. Park, “A Study on the Optimal Design of Polynomial Neural Networks Structure”, *The Transactions of The Korean Institute of Electrical Engineers*, Vol. 49D, No. 3, pp.145-156, Mar., 2000.
- [28] S.-K. Oh, W. Pedrycz, and D.-W. Kim, “Hybrid Fuzzy Polynomial Neural Networks”, *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 10, No. 3, pp. 257-280, June, 2002.
- [29] L. X. Wang, J. M. Mendel, “Generating fuzzy rules from numerical data with applications”, *IEEE Trans. Systems, Man, Cybern.*, Vol. 22, No. 6, pp. 1414-1427, 1992.
- [30] J. S. R. Jang, “ANFIS: Adaptive-Network-Based Fuzzy Inference System”, *IEEE Trans. System, Man, and Cybern.*, Vol. 23, No. 3, pp. 665-685, 1993.
- [31] L. P. Maguire, B. Roche, T. M. McGinnity, L. J. McDaid, “Predicting a chaotic time series using a fuzzy neural network”, *Information Sciences*, Vol. 112, pp. 125-136, 1998.
- [32] C. James Li, T. -Y. Huang, “Automatic structure and parameter training methods for modeling of mechanical systems by recurrent neural networks”, *Applied Mathematical Modeling*, Vol. 23. pp. 933-944.

1999.

- [33] A. S. Lapedes and R. Farber. "Nonlinear signal processing using neural networks: prediction and systems modeling". Technical Report LA-UR-87-2662, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, 1987.
- [34] 오성권, "C 프로그래밍에 의한 퍼지모델 및 제어시스템", 내하출판사, 2002. 1.
- [35] 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



이 동 윤 (李 東 潤)

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