

GA 기반 고급 자기구성 퍼지 다항식 뉴럴 네트워크의 최적화

Optimization of GA-based Advanced Self-Organizing Fuzzy Polynomial Neural Networks

박호성, 박건준, 오성권
원광대학교 전기전자및정보공학부

Ho-Sung Park, Keon-Jun Park, Sung-Kwun Oh
School of Electrical and Electronic Engineering, Wonkwang University
E-mail : ohsk@wonkwang.ac.kr

요 약

기존의 SOFPNN은 데이터 수가 적고 비선형 요소가 많은 시스템에 대한 체계적 이고 효율적인 최적 모델을 구축할 수 있었으며 각 층 노드의 선택 입력을 변화시킴으로써 네트워크 구조 전체의 적응능력을 향상 시켰다. SOFPNN의 구조는 퍼지 다항식 뉴론(FPN)들로 구성되어 있으며, 층이 진행되는 동안 모델 스스로 노드의 선택과 제거를 통해 최적의 네트워크 구조를 생성할 수 있는 유연성을 가지고 있다. 그러나, 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. 본 논문에서는 자기구성 퍼지 다항식 뉴럴네트워크(Self-Organizing Fuzzy Polynomial Neural Networks : SOFPNN)을 최적화시키기 위해 유전자 알고리즘을 이용하여 자기구성 퍼지 다항식 뉴럴 네트워크의 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.

1. 서론

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[1]. 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) [2]와 자기구성 퍼지 다항식 뉴럴 네트워크(Self-Organizing Fuzzy Polynomial Neural Networks) [3]가 제안되었다. 그러나 기존 SOFPNN 구조는 성능을 향상시키기 위해서, 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. 따라서 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 요구되

어진다.

본 논문에서는 고급 자기구성 퍼지 다항식 뉴럴네트워크(SOFPNN)을 최적화시키기 위해서 유전자 알고리즘[4, 5]을 이용하여 고급 자기구성 퍼지 다항식 뉴럴 네트워크의 입력변수의 수와 이에 해당되는 입력변수 그리고 규칙 후반부 다항식의 차수를 탐색하여 최적의 자기구성 퍼지 다항식 뉴럴 네트워크를 구축한다.

본 논문의 평가를 위해서 Box와 Jenkins가 사용한 가스로 입출력 데이터[6]을 이용하여 제안된 모델의 타당성, 정확성 및 예측 성능 등을 기존 지능 모델들과 비교를 통해 그 우수성을 보인다.

2. 퍼지 다항식 뉴런(FPN)

FPN은 SOFPNN의 가장 기본이 되는 처리 단위의 일반적인 형태로서, 2개의 기본적인 기능 모듈로 구성되어 있다. 처음 하나는 "F"로 표기되어진 것으로, 이는 퍼지 집합의 집합체로서 입력 수의 변수들과 뉴런에 의해 실현된 처리 부분사이를 상호 연결해 준다. "P"로 표기된 두 번째 모듈은 함수 기반 비선형(다항식) 처리에 관한 것이다. 이 비선형 처리는 어떤 입력 변수들을 포함한다. 특별히, 비선형성의 다항식 형태인 퍼지 다항식 처리 유닛을 사용한다. 특히, 사용되는 다항식에는 상수 그리고 다항식의 특별한 경우로써의 선형 변환을 포함한다.

3. GA 기반 고급 자기구성 퍼지 다항식 뉴럴 네트워크 설계

본 장에서는 기존의 자기구성 퍼지 다항식 뉴럴 네트워크를 유전자 알고리즘을 이용하여 고급 자기구성 퍼지 다항식 뉴럴 네트워크 구조를 최적화하는 알고리즘 및 설계절차를 다룬다.

[단계 1] 출력변수 y 에 관계하는 n 개의 시스템 입력변수(x_1, x_2, \dots, x_n)를 결정한다.

[단계 2] N 개의 입출력 데이터 $(x_i, y_i) = (x_{1i}, x_{2i}, \dots, x_{ni}, y_i), i=1, 2, \dots, N$ 을 학습용 데이터와 평가용 데이터로 분할한다.

[단계 3] SOFPNN 구조를 구축하는 있어서 기본적인 정보를 결정한다. 즉, SOFPNN의 퍼지 추론 방법 및 동정을 설계하는 부분과 SOFPNN 모델의 구조를 결정하는 초기 정보를 결정한다.

[단계 4] 유전자 알고리즘을 이용한 SOFPNN 구조를 결정한다.

유전자 알고리즘은 유전학과 자연 진화를 흉내낸 적응 탐색법으로 1975년 Holland [4]에 의해 개발되었다. 본 논문에서는 유전자 알고리즘을 사용

레이션하기 위해 한 개체에 여러 변수의 정보를 담은 직렬 방식을 사용한다. 재생산 연산으로 룰렛 휠 방식, 교차 연산은 one-crossover 방식, 돌연변이 연산은 선택된 비트를 반전시키는 invert 방식을 사용하였다. 그러나 룰렛 휠 방식은 확률적 속성 때문에 선택 과정에서 최적자를 반드시 선택하지 못하는 단점을 가진다. 이를 보완해줄 수 있는 한 방법으로 엘리트 전략을 사용하였다[5].

[단계 4-1] 입력변수의 수 선택

[단계 4-2] 퍼지규칙 후반부 다항식 차수 선택

[단계 4-3] 입력변수 선택

[단계 4-1]부터 [단계 4-3]까지 수행해서 얻은 선택된 입력변수와 퍼지규칙 후반부 다항식 차수를 이용하여 퍼지 추론을 적용한다. [단계 4-2]에서 결정된 퍼지 추론 방법인 간략 추론 및 회귀 다항식 추론의 후반부 구조는 표 1에서 표현한 것과 같이 Type 1~4를 사용한다.

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

표 1 퍼지규칙 후반부에서 회귀다항식의 서로 다른 형태

퍼지 추론 규칙과 파라미터 동정을 위한 매개변수 추정은 다음과 같다.

i) 간략 추론

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

$$R^j: \text{If } x_1 \text{ is } A_{j1} \text{ and } \dots \text{ and } x_n \text{ is } A_{jn} \text{ then } y_j \text{ is } a_{j0} \quad (1)$$

i 번째 데이터의 추론된 값 \hat{y}_i 는 다음과 같다.

$$\hat{y}_i = \frac{\sum_{j=1}^m w_{ji} y_j}{\sum_{j=1}^m w_{ji}} = \frac{\sum_{j=1}^m w_{ji} a_{j0}}{\sum_{j=1}^m w_{ji}} = \sum_{j=1}^m \hat{w}_{ji} a_{j0} \quad (2)$$

$$w_{ji} = A_{j1}(x_{1i}) \wedge \dots \wedge A_{jn}(x_{ni})$$

여기서, i 는 $i(i=1, \dots, m)$ 번째 데이터이고, 후반부 파라미터 a_{j0} 의 동정은 입출력 데이터가 주어졌을 때 최소자승법에 의해 구해진다.

ii) 회귀다항식 추론

후반부가 다음과 같이 1, 2, 변형된 2차 다항식 형태로 표현된 것으로 회귀다항식 추론이라 한다.

$$R^j : \text{If } x_1 \text{ is } A_{j1} \text{ and } \dots \text{ and } x_n \text{ is } A_{jn} \\ \text{then } y_j = f_j(x_1, \dots, x_n) \quad (3)$$

[단계 5] 모델의 구축과 테스트 그리고 노드 선택을 한다. 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, 돌연변이를, 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.

Step 3) Population에 의해서 생성된 노드의 성능 테스트를 위해서 기존의 SOFPNN에서와 같이 모델의 테스트 데이터만을 이용하여 노드를 테스트한다. 유전자 알고리즘의 적합도 함수로 계산하기 위해서 다음 식 (4)와 같이 변환하여 유전자 알고리즘의 적합도 함수로 사용한다.

$$F(\text{Fitness Function}) = 1/(1+EPI) \quad (4)$$

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

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

Step 6) 정렬된 적합도 값들 중에서 중복된 적합도 값들은 제거하고 하나의 적합도 값으로 처리한다.

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

Step 8) Step 7)에서 선택되어진 적합도 값들을 가진 노드들을 다시 초기 population 번호 순서에 따른 오름차순으로 재정렬한다.

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 Step2)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 대치하는 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.

[단계 6] [단계 5]의 현재 층에서 얻어진 최적 노드의 최대 적합도 값 F_1 이 다음 부등식을 만족하는 경우에 알고리즘을 종료한다.

$$F_1 \geq F^* \quad (5)$$

단, F^* 는 이전 층의 최적 노드인 최대 적합도 값이다.

본 논문에서의 모델의 성능지수식으로 식 (6)과 같이 MSE(Mean Squared Error)을 사용해서 구해진다.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - y_i^*)^2 \quad (6)$$

[단계 7] 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, $x_{1j}=z_{1i}, x_{2j}=z_{2i}, \dots, x_{wj}=z_{wi}$ 으로 구성하고, [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 입력이 되는 전 층의 출력을 대입하고 동일한 조작을 제 1층까지 반복하여 최종 추정 모델 \hat{y} 를 얻는다.

4. 시뮬레이션 및 결과 고찰

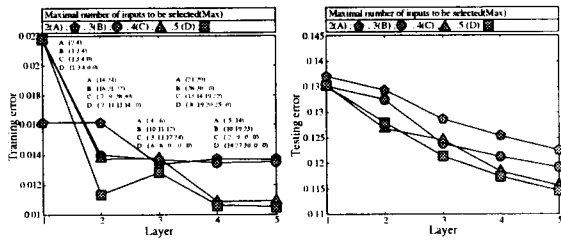
제안된 모델의 성능을 테스트하기 위해서 가스로 시계열 데이터[9]를 이용하였다. 1입력 1출력의 가스로 시계열 데이터 296쌍을 시뮬레이션 하기 위해 전체 시스템 입출력을 [$u(t-2), u(t-1), y(t-2), y(t-1) : y(t)$]와 같이 설정하여 사용한다. 표 2는 유전자 알고리즘의 염색체 정보와 고급 자기구성 다항식 뉴럴 네트워크 구축을 위한 정보를 나타낸 것이다.

	Parameters	1층~5층
GA	Maximum gen.	100
	Total population size	60
	Selected population size	30
	Crossover rate	0.65
	Mutation rate	0.1
	String length	3+3+30
SOFPNN	Maximal no. of inputs to be selected(Max)	$1 \leq l \leq \text{Max}(2 \sim 5)$
	Polynomial Type(Type T) of the consequent part of rules	$1 \leq T \leq 4$
	Membership function(MF) type	Gaussian
	No. of MFs per each input	2

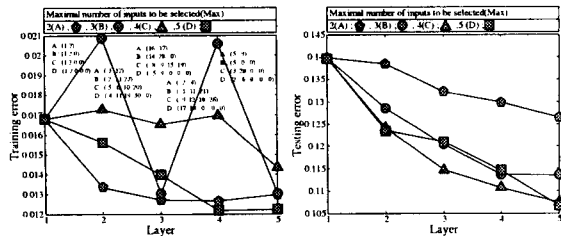
l, T ; integer

* ; 규칙 후반부 다항식의 입력이 전체 시스템 입력이 사용된 경우

표 2 유전자 알고리즘과 자기구성 퍼지 다항식 뉴럴 네트워크 정보



(a) Triangular MF



(b) Gaussian-like MF

그림 1 층의 증가에 따른 SOFPNN의 성능지수

그림 1은 선택된 입력의 변화에 따른 SOFPNN 각 층의 성능지수를 나타낸 것이다. 그림 2는 선택된 입력변수의 수가 4입력이고 멤버십 함수로는 삼각형 함수를 사용한 경우로 각 층에서 generation 수행 과정에 따른 Training 성능지수와 Testing 성능지수의 최적화 과정을 나타낸 것이다.

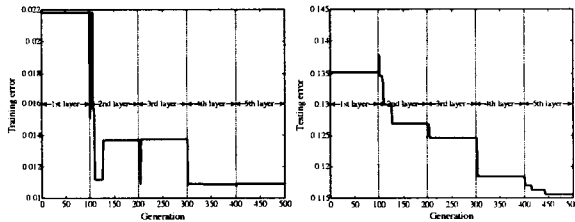


그림 2 유전자 알고리즘에 의한 각 성능지수들의 최적화 과정

표 3은 가스로 공정 데이터를 사용하여 성능지수를 식 (6)에 의해 계산된 값으로 기존의 모델과의 성능비교를 한 것이다.

모델	PI	EPI
Oh and Pedrycz's model[7]	0.271	0.020
Kim, et al.'s model[8]	0.244	0.034
SONN[3]	Basic	0.016
	Modified	0.018
Our model	Triangular	0.010
	Gaussian-like	0.012

표 3 기존의 퍼지모델과 동정오차비교

5. 결론

본 논문에서는 유전자 알고리즘을 이용하여 고급 자기구성 퍼지 다항식 뉴럴 네트워크의 효율적인 설계 방법을 제안하였고 이를 기반으로 주

어진 비선형 예제를 통하여 GA기반 SOFPNN의 우수성을 입증하였다. 기존 SOFPNN 모델의 구조의 한계를 극복하고자 유전자 알고리즘을 이용하여 노드의 입력변수의 수와 그에 해당하는 입력변수 그리고 퍼지규칙 후반부 다항식 차수 등의 파라미터를 선택하여 더욱 정확한 예측 능력을 가진 최적화된 SOFPNN 모델 구조를 구축할 수가 있었다.

감사의 글

본 연구는 산업자원부의 지원에 의하여 기초전력공학공동연구소(R-2003-B-274) 주관으로 수행된 과제임.

6. 참고문헌

- [1] A. G. Ivakhnenko, "Polynomial theory of complex systems", *IEEE Trans. on Systems, Man and Cybernetics*, Vol. SMC-1, pp. 364-378, 1971.
- [2] S.-K. Oh and W. Pedrycz, "The design of self-organizing Polynomial Neural Networks", *Information Science*, Vol. 141, pp. 237-258, 2002.
- [3] 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.
- [4] Holland, J. H., *Adaptation In Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, 1975.
- [5] 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.
- [6] G. E. P. Box and G. M. Jenkins, *Time Series Analysis, Forecasting, and Control*, 2nd edition Holden-Day, SanFrancisco, 1976.
- [7] 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.
- [8] E.-T. Kim, et al, "A simple identified Sugeno-type fuzzy model via double clustering", *Information Science*, Vol. 110, pp. 25-39, 1998.
- [9] 오성권, "프로그래밍에 의한 컴퓨터 지능(퍼지, 신경회로망 및 진화알고리즘을 중심으로)", 내하출판사, 2002.