

FNN 및 PNN에 기초한 FPNN의 합성 다층 추론 구조와 알고리즘

| |
|---------|
| 論 文 |
| 49D-7-6 |

The Hybrid Multi-layer Inference Architectures and Algorithms of FPNN Based on FNN and PNN

朴炳俊* · 吳聖權** · 金賢起***
(Byoung-Jun Park · Sung-Kwun Oh · Hyun-Ki Kim)

Abstract - In this paper, we propose Fuzzy Polynomial Neural Networks(FPNN) based on Polynomial Neural Networks(PNN) and Fuzzy Neural Networks(FNN) for model identification of complex and nonlinear systems. The proposed FPNN is generated from the mutually combined structure of both FNN and PNN. The one and the other are considered as the premise part and consequence part of FPNN structure respectively. As the consequence part of FPNN, PNN is based on Group Method of Data Handling(GMDH) method and its structure is similar to Neural Networks. But the structure of PNN is not fixed like in conventional Neural Networks and self-organizing networks that can be generated. FPNN is available effectively for multi-input variables and high-order polynomial according to the combination of FNN with PNN. Accordingly it is possible to consider the nonlinearity characteristics of process and to get better output performance with superb predictive ability. As the premise part of FPNN, FNN uses both the simplified fuzzy inference as fuzzy inference method and error back-propagation algorithm as learning rule. The parameters such as parameters of membership functions, learning rates and momentum coefficients are adjusted using genetic algorithms. And we use two kinds of FNN structure according to the division method of fuzzy space of input variables. One is basic FNN structure and uses fuzzy input space divided by each separated input variable, the other is modified FNN structure and uses fuzzy input space divided by mutually combined input variables. In order to evaluate the performance of proposed models, we use the nonlinear function and traffic route choice process. The results show that the proposed FPNN can produce the model with higher accuracy and more robustness than any other method presented previously. And also performance index related to the approximation and prediction capabilities of model is evaluated and discussed.

Key Words : 퍼지 다항식 뉴럴네트워크(Fuzzy Polynomial Neural Networks:FPNN), 퍼지 뉴럴네트워크(Fuzzy Neural Networks:FNN), 다항식 뉴럴네트워크(Polynomial Neural Networks:PNN), GMDH (Group Method of Data Handling), BFPNN (Basic FPNN), MFPNN(Modified FPNN)

1. 서 론

컴퓨터와 제어이론의 발전으로 인해 산업계에서는 자동화 범위의 확대가 진행되고 있다. 그러나, 대상 프로세서의 비선형성 때문에 제어범위가 좁고, 모델 파라미터의 결정이 어려워 선형성을 바탕으로 개발된 기존의 방법들을 이용하여 이들을 해석하기란 쉬운 일이 아니다. 또한 시스템 모델링에 관한 연구에서도 입출력 데이터에 의한 선형계의 동정과 예측문제에 관한 회귀분석 기법과 같은 수학적 모델링이 유용하게 사용되어 왔으나 모델링을 하고자 하는 시스템이 복잡하고, 대규모 구조인 경우는 함수식으로 모델의 표현이 한정되지 않기 때문에 회귀분석을 적용하기가 어렵다. 또한

일반적으로 복잡한 다변수계의 모델링은 많은 입출력 변수 중 모델의 구조를 결정짓는 변수 선택에 어려움이 있으며, 추정방법의 계수가 많으면 추정에 필요한 데이터 양이 방대해지는 문제점을 지니고 있다. 이러한 문제들과 어려움을 해결하기 위해 퍼지 모델, 신경회로망과 같은 지능형 모델에 관한 연구가 진행되었으며 최근에는 퍼지 모델과 신경회로망이 가지고 있는 독특한 특성들을 살리면서 지능 모델의 효율과 성능향상을 위한 두 모델의 융합에 관한 연구가 여러 가지 각도에서 다양하게 제시되고 있다. Takagi, Hayashi[5]는 신경회로망으로 구동되는 퍼지추론을 제안하였으나 입출력 수가 적은 경우는 그 효과가 미미하고, 전반부가 신경회로망이라는 특성화된 함수로 구성되어 기존의 퍼지추론보다 시스템 표현에 있어서 정성적으로 이해하기가 힘들다는 단점을 가지고 있다. Horikawa[6]는 퍼지-뉴럴네트워크에 의한 합성방법과 동정에 관한 연구를 하였다. 그러나 퍼지규칙의 전반부인 시그모이드 형태의 멤버쉽함수 초기치 선정이 어렵고, 최적 성능으로의 수렴특성이 나쁘다. Imasaki[15]는 퍼지규칙이 전반부 네트워크, 추론 네트워크, 후반부 네트워크로 구조화된 신경회로망을 제안하였으며, 이

* 準 會 員 : 圓光大 電氣電子工學部 博士課程
** 正 會 員 : 圓光大 電氣電子工學部 副教授 · 工博
*** 正 會 員 : 水原大 電氣電子情報通信工學部 副教授 · 工博
接受日字 : 2000年 3月 8日
最終完了 : 2000年 5月 29日

는 추론 네트워크의 추가로 인한 오랜 학습 시간과 동조의 난점을 가지고 있다. 또한 입출력 증가에 따른 네트워크 구성시 경험적인 방법을 많이 요구한다. Nomura[8]는 델타규칙에 의한 퍼지 추론의 자동 동조 방법을 제안하였다. 그러나 이것은 학습에 의한 동조로 각 규칙마다 멤버쉽함수가 변형됨에 따라 퍼지규칙 수가 증가하며, 입출력의 증가는 더 많은 퍼지규칙 수를 요구한다. Oh[25]는 다항식 뉴럴네트워크로서 고차 다변수 입력에 의한 다양한 구조의 자기 구성 네트워크를 설계하였으나 입출력 변수에 의한 공간분할의 효과적 이용과 제한된 입력변수의 선택에 따른 많은 어려움이 따른다.

따라서 복잡한 비선형계 모델링을 위하여 본 논문에서는 퍼지 뉴럴네트워크(Fuzzy Neural Networks:FNN) 및 다항식 뉴럴네트워크(Polynomial Neural Networks:PNN) 구조의 상호 결합으로부터 두 종류의 퍼지 다항식 뉴럴네트워크(Fuzzy Polynomial Neural Networks:FPNN) 구조-BFPNN 및 MFPNN-의 최적 설계 방법을 제안한다. FPNN의 전반부 구조로 이용된 FNN은 두 가지-상호 분리된 입력변수 공간에 의한 기본 FNN(Case 1)과 결합된 입력변수 공간에 의한 수정된 FNN(Case 2)-의 경우로 나누어 설계된다. Case 1은 Yamakawa가 제안한 구조[9,21]로, 다른 비선형 모델에 비해서 학습속도가 빠르고 수렴특성이 우수하다는 장점을 가졌으나, 입력변수들의 상호 관계를 고려하지 않고 각 입력변수에 대해 분리된 퍼지공간을 사용한 단점을 가졌다. Case 2는 Case 1의 단점을 보완하기 위한 변형된 구조이며, 입력변수들의 상호 관계를 고려하여 퍼지 공간을 분할한 것으로 공정의 비선형 정도와 상호 입력변수의 상관성을 고려하여 공정 시스템에 선택적응이 가능함으로써 그 적응능력을 향상시킨다.

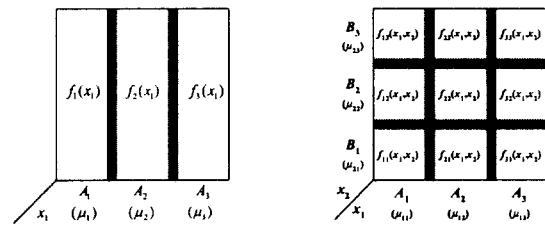
FPNN의 후반부를 구성하는 PNN은 GMDH 알고리즘[1,2,4]을 기반으로 하며, 기존의 신경회로망 구조와 같이 고정된 구조가 아닌 새로운 구조를 생성하는 네트워크 구조를 가진다. A. G. Ivakhnenko가 제안한 GMDH 알고리즘[1]은 입력변수의 선택과 입출력 데이터의 분할, 2변수 2차식의 부분표현식에 의해 시스템 방정식을 예측하였다. 그러나 2변수의 입력변수는 입력변수들 사이의 상호의존적인 면을 약하게 하였으며, 간단히 표현될 수 있는 경우 간략화된 구조로 나타내는 것이 효과적이지만 2차식으로 모든 경우를 표현함으로써 해당 공정모델의 적응력이 떨어진다. 따라서 다항식의 차수를 1차(linear), 2차(quadratic), 3차(cubic) 등으로 확장하였다. 이것은 공정 모델 특성과 계층구조의 간략성 및 상호 연계성 고려로부터 고차 회귀다항식 사용이 용이하고 간단한 구조로 설계됨을 알 수 있다. 특히, PNN 입력층의 각 노드는 FNN 구조의 중간 혹은 출력층의 각 노드에 연결된다. 이 두 형태의 연결 방법을 고려함으로써, 다입력 변수 및 고계 다항식을 가진 PNN 구조를 보다 효과적으로 이용할 수 있다. 이로부터 공정의 비선형 특성을 고려할 수 있고, 더 향상된 출력 특성을 얻는 것이 가능하다.

모델의 성능평가를 위해 비선형 함수[3,6,19]와 교통 경로 선택 데이터[18,20,21]를 이용한다. 또한 기존의 학습용 데이터뿐만 아니라 평가용 데이터를 이용한 모델의 성능, 즉 예측능력과 일반화 능력 모두를 고려하여 실 공정 적용의 유용성을 검토한다.

2. 퍼지 다항식 뉴럴네트워크(FPNN)의 구조와 알고리즘

2.1 퍼지 뉴럴네트워크(FNN)

본 논문에서는 제안된 FPNN의 전반부 구조로 FNN을 이용한다. FNN은 퍼지추론의 최소-최대 연산과 신경회로망의 곱셈-덧셈 연산, 멤버쉽함수와 시그모이드 함수에 의해 시스템의 비선형성을 표현한다는 유사성과 퍼지추론이 논리구조를 취급하는 반면에, 신경회로망은 학습기능을 가진다는 상호 보완관계를 이용한 것이다.



(a) Case 1 : 분리된 입력변수 (b) Case 2 : 결합된 입력변수
그림 1 입력 퍼지 공간 분할

Fig. 1 Division of input fuzzy space

FNN은 분리된 입력변수 퍼지 공간에 의한 기본 FNN(Case 1)[9,21]과 결합된 입력변수 퍼지 공간에 의한 수정된 FNN(Case 2)으로 나누어 설계된다. Case 1은 그림 1(a)와 같은 퍼지공간 분할을 사용한 것으로 입력변수간의 상호관계를 고려하지 않은 각 변수의 멤버쉽함수 적합도를 구함으로써 모델을 구축하는 방법이고, 그림 1(b)와 같은 형태의 Case 2는 각 입력변수간의 상호관계를 고려한 경우로 멤버쉽 값을 구하고 규준화된 적합도를 계산하여 각 연결가중치와 곱함으로써 입력관계를 최종 출력에 반영한 구조이다.

2.1.1 분리된 입력변수 퍼지공간 분할에 의한 FNN(Case 1)

분리된 입력변수 퍼지공간 분할에 의한 기본 FNN 구조는 그림 2와 같다.

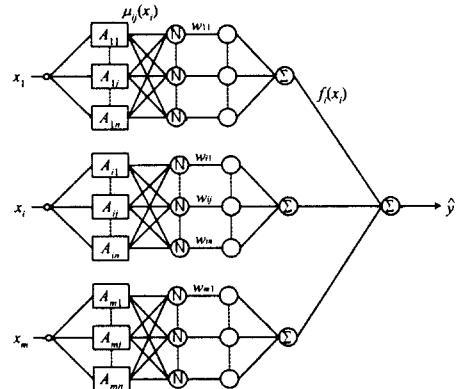


그림 2 분리된 입력변수 퍼지공간 분할에 의한 FNN 구조
Fig. 2 FNN Structure by means of the fuzzy space division of the separated input variables

그림 2에서 x_i 는 입력으로 퍼지-뉴런의 특성을 나타내는 비선형 함수 f_i 에 의해 출력 $f_i(x_i)$ 를 생성한다. 퍼지-뉴런에서 출력된 값들은 식 (1)과 같이 모두 대수적으로 합해져 \hat{y} 를 최종적으로 출력하게 된다.

$$\hat{y} = f_1(x_1) + f_2(x_2) + \dots + f_m(x_m) = \sum_{i=1}^m f_i(x_i) \quad (1)$$

여기서, m 은 퍼지-뉴런의 수이다.

$$R^j: \text{If } x_i \text{ is } A_{ij} \text{ Then } y_{ij} = w_{ij} \quad (2)$$

각각의 퍼지-뉴런(f_i)의 내부구조는 퍼지규칙을 이루고 있으며 그 규칙은 식 (2)의 형태이다. 여기서 R_j 는 j 번째 퍼지규칙을 나타내며, A_{ij} 는 전반부의 퍼지변수이다. 후반부의 w_{ij} 는 상수이며, 그림 2의 연결 가중치를 나타낸다. 퍼지추론의 전반부 멤버십함수는 보수적(complementary)인 관계를 가진 삼각형 형태이다. 식 (2)의 최종 추론결과는 식 (3)과 같다. 이 식은 무게 중심법(center of gravity)에 의한 비퍼지화(defuzzification) 과정까지 포함한다.

$$f_i(x_i) = \frac{\sum_{j=1}^n \mu_{ij}(x_i) \cdot w_{ij}}{\sum_{j=1}^n \mu_{ij}(x_i)} \quad (3)$$

FNN의 학습은 그림 2의 멤버십함수 뒷부분에 위치한 연결 가중치를 모델에 가장 적절한 값으로 조정해 나가면서 이루어진다. 학습방법은 신경회로망의 학습에 가장 일반적으로 사용되는 BP(Back-Propagation) 알고리즘이 사용되는데, 이는 최종출력과 원 데이터간의 차, 즉 오차를 바탕으로 이루어진다[9,21].

연결 가중치의 조절은 식 (4)와 같이 기존의 값에 오차로부터 구해진 조절량 만큼의 변화를 더해주면 된다.

$$w(\text{new}) = w(\text{old}) + \Delta w \quad (4)$$

연결 가중치의 변화량 Δw 는 식 (5)와 같다. 여기서 η 는 학습율이다.

$$\Delta w_{ij} = 2 \cdot \eta \cdot (y_p - \hat{y}_p) \cdot \mu_{ij}(x_i) \quad (5)$$

여기에 모멘텀을 더해 학습시간을 좀 더 단축시키고, 학습능력 역시 향상시킨다. 모멘텀은 관성이란 뜻으로, 연결 가중치의 변화량이 클 경우에는 좀 더 변화량을 키워주고, 변화량이 작아지면 또 여기에 맞춰서 변화량을 줄여주는 역할을 한다. 모멘텀은 식 (6)과 같이 현재의 연결 가중치와 그 이전 연결가중치의 차이에 의해 계산된다.

$$m(t) = w(t) - w(t-1) \quad (6)$$

이렇게 계산된 모멘텀은 모멘텀계수라 불리는 α 와 곱해져 연결 가중치 조절식에 추가된다. 모멘텀까지 포함하는 연결 가중치 변화량은 식 (7)과 같다.

$$\Delta w_{ij} = 2 \cdot \eta \cdot (y_p - \hat{y}_p) \cdot \mu_{ij}(x_i) + \alpha(w_{ij}(t) - w_{ij}(t-1)) \quad (7)$$

2.1.2 결합된 입력변수 퍼지공간 분할에 의한 FNN(Case 2)

결합된 입력변수 퍼지공간 분할에 의한 FNN은 비선형적인 데이터들간의 상호 연관성을 찾아 그림 1(b)와 같은 입력 공간의 분할로부터 데이터를 구성하고 연결가중치의 집합에 의해 표현되는 FNN 구조이다. 이는 입력 데이터들간의 상호 연관성을 찾아 입력 공간을 분할하여 각 입력변수들의 상호관계를 나타내는 퍼지규칙의 적합도에 연결가중치

를 곱함으로써 FNN의 각 규칙을 생성한다. 2입력 1출력을 가지고 각 입력변수에 대하여 3개의 멤버십함수를 사용하는 FNN 구조를 그림 3에 보였다.

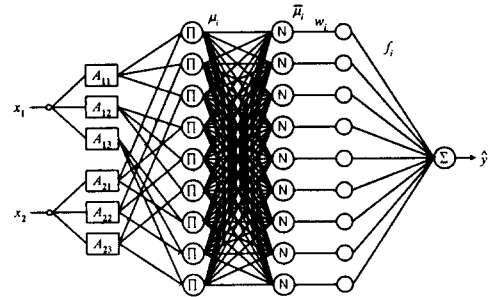


그림 3 결합된 입력변수 퍼지공간 분할에 의한 FNN 구조
Fig. 3 FNN structure by means of the fuzzy space division of the combined input variables

$$R^j: \text{If } x_1 \text{ is } A_{1j} \text{ and } \dots x_k \text{ is } A_{kj} \text{ Then } y = w_j \quad (8)$$

FNN의 R_j 는 j 번째 퍼지규칙으로 식 (8)과 같으며 Case 1의 FNN과 유사하면서도 다른 구조를 가진다. Case 1은 그림 2처럼 퍼지-뉴런(f_i)의 내부구조가 퍼지규칙으로 구성되지만 이 구조는 네트워크의 전체 구조가 퍼지규칙을 이루고 있다. FNN의 추론과정은 각 규칙의 적합도를 구하고 모든 규칙의 적합도 합을 비율을 계산한다. 즉 표준화된 $\bar{\mu}_i$ 는 하중계수 w_i 와 곱하며, 출력층의 입력이 된다. 각 노드의 출력 f_i 는 식 (9)와 같이 모두 대수적으로 합해져 \hat{y} 를 최종적으로 출력하게 된다.

$$\hat{y} = \sum_{i=1}^n f_i = \sum_{i=1}^n \bar{\mu}_i \cdot w_i = \sum_{i=1}^n \frac{\mu_i \cdot w_i}{\sum_{i=1}^n \mu_i} \quad (9)$$

FNN의 학습은 Case 1과 동일한 BP 알고리즘이 사용되며, 본 논문에서는 식 (10)과 같은 형태의 출력오차를 사용한다.

$$E_p = |y_p - \hat{y}_p| \quad (10)$$

여기서 E_p 는 각 데이터에 대한 오차이고, y_p 는 주어진 출력 데이터, \hat{y}_p 는 모델의 출력을 나타낸다. m 개의 입력층 데이터 쌍이 주어진 경우, FNN의 최종 출력 오차는 본 논문에서 다음의 식 (11), (12) 두 가지 형태를 사용한다.

$$E = \frac{1}{N} \sum_{p=1}^N \frac{|y_p - \hat{y}_p|}{y_p} \times 100(\%) \quad (11)$$

$$E = \sum_{p=1}^N |y_p - \hat{y}_p| \quad (12)$$

연결 가중치의 조절은 식 (4)와 같으며 오차 E_p 를 최소화하기 위한 연결 가중치의 변화량 Δw 는 식 (13)에서 보이는 것처럼 경사강하법(gradiant descent method)을 이용하여 구한다.

$$\Delta w_i = \eta \cdot \left(-\frac{\partial E_p}{\partial w_i} \right) \quad (13)$$

식 (13)에서 우변의 편미분은 사슬법칙(chain rule)에 의해 식 (14)과 같이 전개할 수 있다.

$$-\frac{\partial E_p}{\partial w_i} = -\frac{\partial E_p}{\partial y_p} \cdot \frac{\partial y_p}{\partial f_i} \cdot \frac{\partial f_i}{\partial w_i} \quad (14)$$

식 (14)에서 우측 식의 각 항은 다음과 같이 유도된다.

i) $y_p \geq \hat{y}_p$ 인 경우

$$-\frac{\partial E_p}{\partial y_p} = -\frac{\partial}{\partial y_p}(y_p - \hat{y}_p) = -(-1) = 1,$$

$$\frac{\partial \hat{y}_p}{\partial f_i} = 1, \quad \frac{\partial f_i}{\partial w_i} = \bar{\mu}_i \quad (15)$$

ii) $y_p < \hat{y}_p$ 인 경우

$$-\frac{\partial E_p}{\partial y_p} = -\frac{\partial}{\partial y_p}(\hat{y}_p - y_p) = -(1) = -1,$$

$$\frac{\partial \hat{y}_p}{\partial f_i} = 1, \quad \frac{\partial f_i}{\partial w_i} = \bar{\mu}_i \quad (16)$$

식 (15), (16)으로부터 연결 가중치의 변화량 Δw_{ij} 는 식 (17)과 같이 된다.

$$\begin{cases} \Delta w_{ij} = \eta \cdot \bar{\mu}_i & \text{for } y_p \geq \hat{y}_p \\ \Delta w_{ij} = -\eta \cdot \bar{\mu}_i & \text{for } y_p < \hat{y}_p \end{cases} \quad (17)$$

모멘텀을 포함한 연결 가중치 변화량은 식 (18)과 같다.

$$\begin{cases} \Delta w_{ij} = \eta \cdot \bar{\mu}_i + \alpha(\omega_{ij}(t) - \omega_{ij}(t-1)) & \text{for } y_p \geq \hat{y}_p \\ \Delta w_{ij} = -\eta \cdot \bar{\mu}_i + \alpha(\omega_{ij}(t) - \omega_{ij}(t-1)) & \text{for } y_p < \hat{y}_p \end{cases} \quad (18)$$

2.1.3 최적화를 위한 유전자 알고리즘

유전자 알고리즘[16,17]은 생물학적 진화, 자연선택, 유전자 재조합의 생물학 원리에 바탕을 둔 확률적 탐색 방법으로, 잠재적으로 좋은 결과를 가져다주는 과정을 추적하고, 반면에 지역 극소에 빠질 가능성을 제거하기 위해 임의의 지점을 검사함으로써, 해 공간 전체를 탐색할 수 있는 능력을 가진다[16]. 유전자 알고리즘의 특징은 주어진 값을 2진 코딩하여 코딩된 문자열을 하나의 개체로 그리고 개체의 비트를 유전자처럼 취급하며, 하나의 지점부터 최적지점으로가 아닌 동시에 여러 지점에서 최적지점을 찾아 나간다. 그리고 미분과 같은 수학적 연산이 아닌 결과의 적합도를 기준(목적함수)으로 수행되며, 결정적인 방법이 아닌 확률적인 방법이다.

유전자 알고리즘은 3개의 기본 연산자-생산, 교배, 돌연변이-를 가지고 있다. 생산(reproduction)은 다음 세대를 위해 교배 유전자가 선택되어지는 과정으로, 적합도에 따라 다음 세대를 생산할 수 있는 기회를 차등적으로 부여한다. 각각의 문자열들은 그들의 적합도를 가지고 다음세대에 전달된다. 교배(crossover)는 두 단계의 과정을 가지는데 첫 번째 단계는 교배 유전자로부터 염색체들은 짝짓는다. 두 번째 단계로 짝지어진 문자열 쌍은 문자열의 k 위치가 $[1, l-1]$ 사이에서 확률적으로 균일하게 선택되어진다. 여기서 l 은 문자열의 길이이다. 선택된 k 와 l 사이의 모든 특성들이 교차함으로써 두 개의 새로운 문자열이 생성되는 과정으로 교배가 이루어진다. 돌연변이(mutation)는 문자열 위치의 값을 임의로 변경한다. 2진 코딩에서 돌연변이는 0을 1로 또는 1을 0으로 바꾸는 것을 의미한다. 적절하게 정의된 적합도 함수와 함께 조합된 연산자는 유전자 알고리즘의 주요 부분이 되며 만들어진 문자열의 군집을 가지고 3단계-평가(evaluation), 선택(selection), 재생산(reproduction)-를 통해 반복 수행된다[21].

본 논문에서는 FNN의 최적화를 위해 유전자 알고리즘을 이용하여 학습을 및 모멘텀계수 그리고 멤버쉽함수의 파라

미터를 동정한다. 모델의 최적화를 위하여 사용된 유전자 알고리즘은 2진(binary)형의 직렬방식을 사용하며, 선택(selection) 연산으로는 roulette-wheel방식, 교배(crossover) 연산은 one-point crossover 연산을 사용한다. 그리고 돌연변이 연산은 선택된 비트를 반전시키는 invert방식을 사용한다. 세대는 100세대, 군집은 60개체까지 진행시켰으며, 각 문자열의 길이는 10 bit를, 교배율은 0.6, 돌연변이 발생율은 0.35를 사용한다.

2.2 다항식 뉴럴네트워크(PNN)

PNN은 본 논문에서 제안한 FPNN의 후반부를 구성한다. GMDH 알고리즘에 기초한 PNN 알고리즘은 차수 및 입력 변수의 선택과 알고리즘의 최종 종료판정 기준에 의하여, 각 층에서 파라미터 동정에 의한 모델을 선정하게 되고, 계층적 모델 추정에 의해 최종 모델을 얻는다. 추정 모델을 얻기 위해, 먼저 부분표현식을 구성한다. 부분표현식으로 회귀다항식을 사용한다[25]. 다음에, 입출력 데이터를 이용하여 선형회귀분석으로 부분표현식의 계수를 구하고 중간변수를 구한다. 중간변수에 의한 새로운 입출력 데이터를 구성하여 부분표현식의 계수와 중간변수를 구한다. 다음 계층에서도 이와 같은 형태의 연산을 되풀이하며, 알고리즘의 종료 판정식을 만족할 때까지 계속 반복된다. 알고리즘을 종료한 후 얻어지는 부분표현식은 계층적으로 조합하고, 최종 추정 모델을 얻는다.

그림 4는 PNN 구조의 예를 보여주며, FPNN 후반부 구조로써 PNN에 사용된 노드의 입력변수 수와 차수는 2, 3, 4 입력 수에 따라 각각 세 Type의 다항식 차수를 이용한다. 즉, Type 1과 Type 2는 1차, 2차 다항식, Type 3는 변형된 2차 다항식이다.

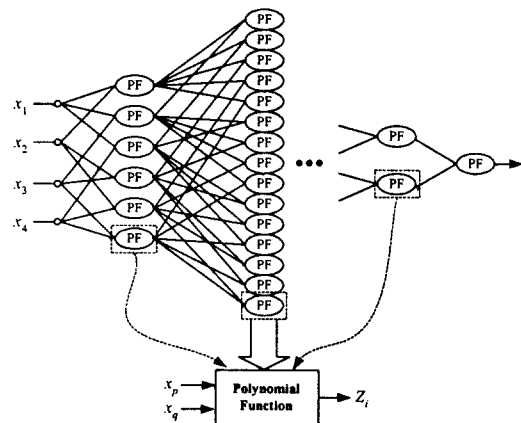


그림 4 PNN 구조
Fig. 4 PNN structure

2.3 퍼지 다항식 뉴럴네트워크(FPNN)

FPNN 구조는 전반부 FNN과 후반부 PNN의 상호 결합으로 구성된다. 전반부 구조인 두 형태의 FNN(기본 FNN과 수정된 FNN)에 따라 두 가지 구조, 즉 BFPNN 및 MFPNN으로 나누어 설계된다. 각 구조의 전반부와 후반부 각 층에 대한 상세 설계과정은 다음과 같다.

Case 1. 분리된 입력변수 공간에 의한 FNN(기본 FNN)과 PNN으로 결합된 BFPNN

분리된 입력변수 공간에 의한 FNN 즉 기본 FNN과 PNN의 결합은 결합 위치에 따라 그림 6과 같은 두 가지 형태의 BFPNN 구조를 가진다. PNN 입력층의 각 노드는 기본 FNN의 중간 혹은 출력층의 노드와 연결된다. 특히 이 기본 FNN과 PNN 구조의 두가지 형태의 연결 방법을 고려함으로써, 다입력 변수 및 고계다항식을 가진 PNN 구조를 효과적으로 이용할 수 있다. 이는 PNN 입력층의 입력변수들을 확장 및 표준화 시키는 역할을 함으로써 PNN 입력변수들의 상호관계를 돕고 모델 출력을 개선시킨다. 뿐만 아니라 적절한 FNN 출력노드를 PNN 알고리즘에 의해 최적 선택할 수 있어 보다 출력 성능을 향상시킬 수 있다. 그림 5는 PNN과 결합되는 FNN의 노드를 보여준다.

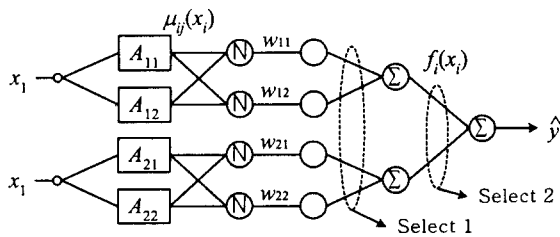
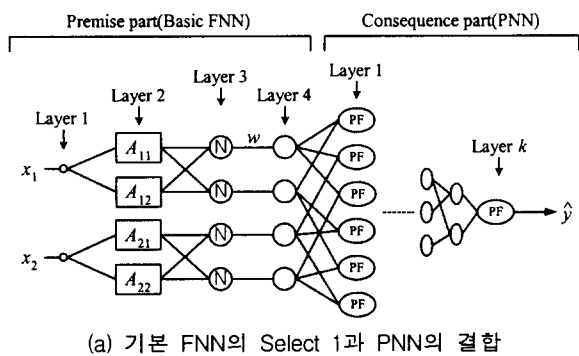
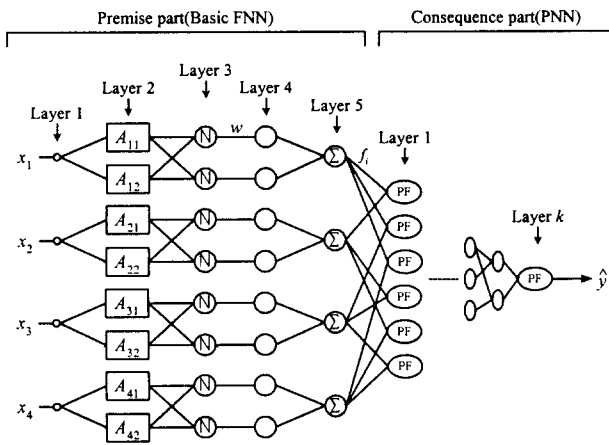


그림 5 PNN과 결합될 기본 FNN의 노드들
Fig. 5 The nodes of basic FNN to be combined with PNN



(a) 기본 FNN의 Select 1과 PNN의 결합



(b) 기본 FNN의 Select 2와 PNN의 결합

그림 6 기본 FNN과 PNN으로 결합된 BFPNN
Fig. 6 BFPNN combined the basic FNN with PNN

BFPNN의 전반부

[Layer 1] 입력층

각 입력 신호는 상호 분리된 입력변수에 대해 퍼지공간으로부터 분할된 멤버쉽함수로 출력된다.

[Layer 2] 적합도 계산

멤버쉽함수는 그림 7과 같은 삼각형 형태이다. 각 입력 신호 x_i 에 의해 멤버쉽함수 $\mu_{ij}(x_i)$ 로부터 멤버쉽 값 즉 적합도를 구한다.

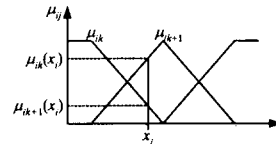


그림 7 보수적 관계를 갖는 멤버쉽함수
Fig. 7 Complementary membership function

[Layer 3] 표준화된 적합도 계산

각 규칙의 적합도가 구해지면 표준화된 적합도 $\bar{\mu}_{ij}$ 를 계산한다.

$$\bar{\mu}_{ij} = \frac{\mu_{ij}}{\sum_{j=1}^n \mu_{ij}} \quad (19)$$

여기서 n 은 각 입력에 대한 멤버쉽함수의 수이다. 입력변수 x_i 는 그림 7과 같이 동시에 두개의 멤버쉽함수 $k, k+1$ 에만 영향을 받기 때문에 식 (19)는 다음과 같이 다시 쓸 수 있다.

$$\bar{\mu}_{ik} = \frac{\mu_{ik}}{\mu_{ik} + \mu_{ik+1}} = \mu_{ik} \quad (20)$$

보수적 관계를 가지는 두개의 이웃하는 멤버쉽함수의 입력변수에 대한 적합도 합은 언제나 1이 된다. 즉, $\mu_{ik}(x_i) + \mu_{ik+1}(x_i) = 1$ 따라서, $\bar{\mu}_{ij} = \mu_{ij}$ 이다. 이와 같은 특징 때문에 추론속도가 증가하는 효과를 얻을 수 있다.

[Layer 4] 연결 가중치와의 곱

구해진 적합도는 연결 가중치와 곱하여진 후 다음 층으로 입력된다. 연결구조 Select 1을 선택한 경우는 그림 6(a)와 같이 후반부 구조인 PNN 입력층의 입력이 되고, Select 2를 선택한 경우는 그림 6(b)와 같이 기본 FNN 출력층의 입력이 된다.

$$a_{ij} = \mu_{ij}(x_i) \times w_{ij} \quad (21)$$

[Layer 5] 각 입력변수의 퍼지추론 값

그림 6(b)의 전반부 5층 출력은 무게 중심법에 의해 퍼지 추론하며 식(22)와 같다.

$$f_i(x_i) = \frac{\sum_{j=1}^n a_{ij}}{\sum_{j=1}^n \mu_{ij}(x_i)} = \frac{\sum_{j=1}^n \mu_{ij}(x_i) \cdot w_{ij}}{\sum_{j=1}^n \mu_{ij}(x_i)} \quad (22)$$

식 (22)를 정리하면,

$$f_i(x_i) = \mu_{ik}(x_i) \cdot w_{ik} + \mu_{ik+1}(x_i) \cdot w_{ik+1} \quad (23)$$

이 f_i 는 연결구조 Select 2를 선택한 경우의 후반부 PNN 입력변수가 된다.

[BFNN의 출력층] BFNN의 최종 출력 계산

5층에서 출력된 값들은 식 (24)과 같이 모두 대수적으로 합해져 BFNN의 출력 \hat{y} 를 최종적으로 계산하게 된다.

$$\hat{y} = f_1(x_1) + f_2(x_2) + \dots + f_m(x_m) = \sum_{i=1}^m f_i(x_i) \quad (24)$$

BFPNN의 후반부

[단계 1] 입력변수 설정

그림 6(a)는 그림 5의 Select 1이 선택된 경우로 전반부 2층 출력인 a_{ij} 를 후반부 구조인 PNN의 입력변수로 설정한다. 즉 $x_1=a_{11}, x_2=a_{12}, \dots, x_n=a_{ij} (n=i \times j)$ 가 되며, 그림 6(b)는 그림 5의 Select 2가 선택된 경우로 전반부 4층 출력인 f_i 를 후반부의 입력변수로 설정한다. 즉 $x_1=f_1, x_2=f_2, \dots, x_n=f_m (n=m: \text{전반부 입력변수의 수})$ 가 된다.

[단계 2] PNN 구조 선택

n 개의 입력변수 x_1, x_2, \dots, x_n 중 노드의 입력변수 수를 선택하고 입력에 대한 노드의 다항식 차수를 선택한다. r 개의 변수 선택시 노드의 개수 k 는 $\frac{n!}{(n-r)!r!}$ 가 된다. 여기서 n 은 전체 입력변수의 수이고, r 은 선택된 입력변수의 수 즉, 노드의 입력변수 수이다. 예를 들어 PNN 각 노드의 부분 표현식을 2입력 2차식이라 하자. n 개의 입력변수 중에서 두 개의 입력변수 x_p, x_q 를 선택하면, 다음과 같은 부분표현식 k 개를 얻을 수 있다.

$$z_k = c_0 + c_1x_p + c_2x_q + c_3x_p^2 + c_4x_q^2 + c_5x_px_q \quad (25)$$

$$k = 1, 2, \dots, \frac{n!}{(n-2)!2!}$$

여기서, c_0, c_1, \dots, c_5 는 계수이며, z_k 는 중간변수이다.

노드의 차수 및 입력변수 선택은 시스템의 특성, 모델 설계방법, 비선형성 및 예측성능 등을 고려함으로써 제기된 문제에 대해 적절한 구조를 결정하는데 도움을 준다.

[단계 3] 매개변수 추정

선형 회귀분석에 의해 학습용 데이터를 사용하여, 식 (26)를 최소화하는 계수 c_0, c_1, \dots, c_5 를 구한다

$$E = \frac{1}{2} \sum_{i=1}^{N_i} \epsilon_i^2 = \frac{1}{2} \sum_{i=1}^{N_i} (y_i - z_{ki})^2 \quad (26)$$

여기서 N_i 는 학습데이터의 수이다. PNN의 파라미터는 계수 c_i 로써, 입출력 데이터가 주어졌을 때 최소자승법에 의해 구해진다. 최소자승법에 의한 계수 동정은 식 (27)와 같다.

$$\hat{C} = (X^T X)^{-1} X^T Y \quad (27)$$

[단계 4] 평가 후 노드의 선택

계수 c_0, c_1, \dots, c_5 를 추정한 식 (25)의 부분표현식에 테스트 데이터를 대입하고, 오차 E 를 계산한다. 얻어진 $E_1, E_2, \dots, E_{n!/(n-r)!r!}$ 에 대해서 값이 작은 노드를 다음 층의 입력변수로 선택한다. 이때 노드 선택은 $E_1, E_2, \dots, E_{n!/(n-r)!r!}$ 에 대하여, 값이 작은 순으로 나란히 정렬하여 이들 중 값이 작은 순으로 다시 W 개를 선택하고, W 개를 제외한 나머지 값들은 버린다. 즉 일정한 노드의 수를 유지해 나간다. 방법 1의 경우 실제적으로 노드 수의 증가를 두드러지게 감소시키지 못하며, 이에 따른 계산량 또한 증가한다. 그러므로 선택

될 노드의 수(W)를 고정시키고 다음 층의 노드 수를 정함으로써 모델링 작업에서 오는 많은 계산량과 과대한 연산시간을 방지한다. 즉, $\frac{n!}{(n-r)!r!} < W$ 인 경우 노드의 수는 $\frac{n!}{(n-r)!r!}$ 이 되고, $\frac{n!}{(n-r)!r!} \geq W$ 인 경우 다음 층 입력변수의 수는 W 가 된다.

[단계 5] 종료판정

알고리즘을 종료하는 방법에는 다음의 2가지 방법이 사용된다.

· 방법 1

[단계 4]에서 얻어진 E_{min} 이 다음의 부등식을 만족할 경우에 알고리즘을 종료한다.

$$E_{min} \geq E_{min*} \quad (E_{min*} \text{는 전 층 오차 } E_{min} \text{이다}) \quad (28)$$

· 방법 2

정해진 층 수에 의해 알고리즘을 종료한다. 방법 1처럼 현재의 오차가 전 층의 오차보다 크다고 하여 종료할 경우, 종료된 이후 층에서 좋은 결과를 얻을 수 있는 기회를 무시할 수 있기 때문이다. 실제로 현재의 오차가 크더라도 다음 층에서 더 좋은 결과가 나타난다. 또한 현재의 오차가 계속해서 전 층의 오차보다 작을 경우 결과의 큰 차이를 보지 못하며, 많은 층의 수행 과정을 거쳐야 하므로 많은 실행 시간과 계산량을 요구하게 된다. 따라서 본 논문에서는 방법 2를 택하여 5층까지 알고리즘을 실행한 후 종료하는 방법을 사용한다.

[단계 6] 새로운 입력변수 선택

종료 판정을 만족하지 않을 경우, $x_{1i}=z_{1i}, x_{2i}=z_{2i}, \dots, x_{wi}=z_{wi}$ 가 되어 새로운 입출력 데이터를 구성하고, 단계 3으로 돌아간다. 이후, 단계 3부터 단계 6까지를 반복한다. 알고리즘이 종료할 경우, E_{min} 을 얻은 부분표현식에 전 층에서의 중간변수를 대입하고, 동일한 조작을 되풀이하여 최종 추정 모델 \hat{y} 를 얻는다.

Case 2. 결합된 입력변수 공간에 의한 FNN(수정된 FNN)과 PNN으로 결합된 MFPNN

그림 8은 결합된 입력변수 공간에 의한 FNN, 즉 수정된 FNN과 PNN이 결합한 MFPNN의 구조를 보여준다.

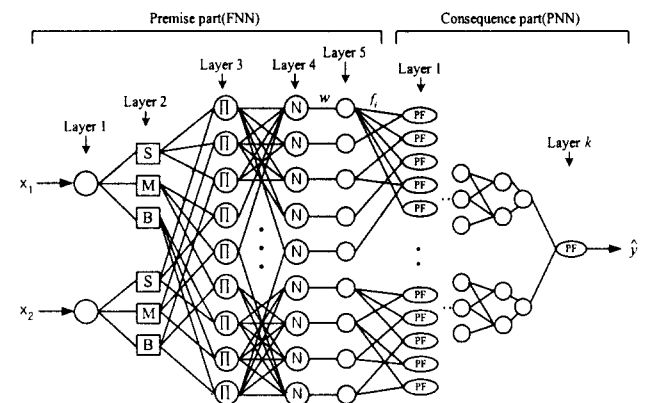


그림 8 수정된 FNN과 PNN으로 결합된 MFPNN
Fig. 8 MFPNN combined the Modified FNN with PNN

MFPNN 전반부

[Layer 1] 입력층

입력신호는 각 입력변수에 대해 퍼지공간으로부터 분할된 멤버쉽함수로 출력된다.

[Layer 2] 입력에 대한 멤버쉽함수 값 계산

각 입력 신호 x_i 에 대한 멤버쉽함수의 값을 계산한다. Case 1 구조의 2층과 같다.

[Layer 3] 각 규칙의 적합도 계산

각 노드는 입력되는 신호들을 곱하여 상호 결합된 퍼지 부분공간을 형성하며, 각 규칙의 적합도를 계산한다.

$$\mu_i = \mu_A(x_1) \times \mu_B(x_2) \quad A, B=S, M, B \quad (29)$$

[Layer 4] 표준화된 적합도 계산

각 규칙의 적합도가 구해지면 모든 규칙의 적합도 합을 비를 식 (30)과 같이 계산한다. 즉, 각 노드의 출력은 표준화된 적합도 $\bar{\mu}_i$ 가 된다.

$$\bar{\mu}_i = \frac{\mu_i}{\sum_{i=1}^n \mu_i} \quad (30)$$

[Layer 5] 연결 가중치와의 곱

4층에서 구해진 적합도는 연결 가중치와 곱해져 후반부 구조인 PNN의 입력이 된다.

$$f_i = \bar{\mu}_i \cdot \omega_i = \frac{\mu_i \cdot \omega_i}{\sum_{i=1}^n \mu_i} \quad (31)$$

[MFNN의 출력층] MFNN의 최종 출력 계산

5층의 출력 f_i 는 식 (32)과 같이 모두 대수적으로 합해져 \hat{y} 를 최종적으로 출력하게 된다.

$$\hat{y} = \sum_{i=1}^n f_i = \sum_{i=1}^n \bar{\mu}_i \cdot \omega_i = \sum_{i=1}^n \frac{\mu_i \cdot \omega_i}{\sum_{i=1}^n \mu_i} \quad (32)$$

MFPNN 후반부 : Case 1과 같은 동일한 구조를 가진다.

그림 9는 FPNN 알고리즘의 순서도이다.

3. 시뮬레이션 및 결과 고찰

본 논문에서 연구되어진 FPNN의 평가를 위해 두 종류의 데이터를 이용한다. 첫 번째는 간단한 수치 예제인 비선형 함수[3,6,19]를, 두 번째는 교통로 선택 데이터[18,20,21]를 이용한다. 모델의 평가 기준인 성능지수(Performance Index: PI)는 수치 데이터인 비선형 함수에 대해 식 (11)을 사용하며, 교통 선택 데이터에 대해서는 식 (12)를 이용한다.

3.1 비선형 함수

본 논문에서 제안한 모델의 적용 및 유용성을 비교 평가하기 위하여 식 (33)로 표현되는 비선형 함수를 이용한다.

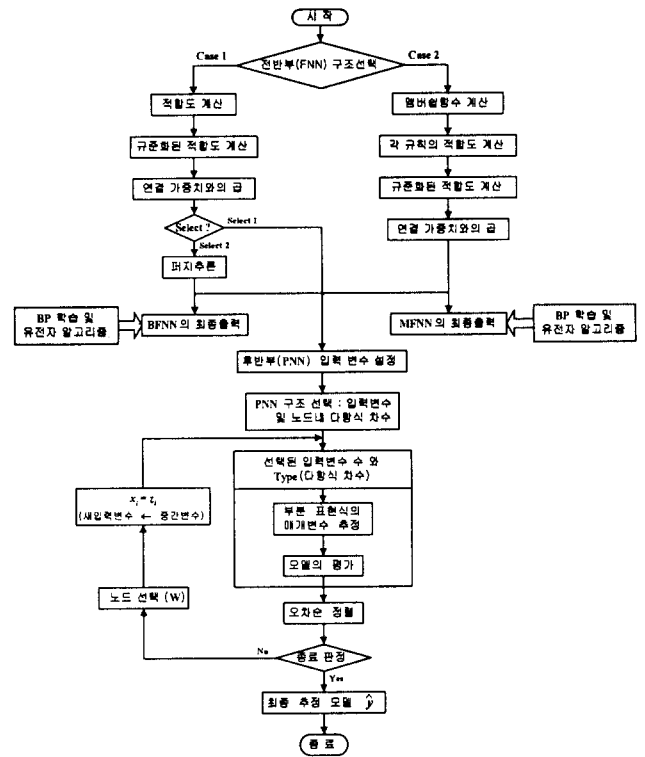


그림 9 FPNN 알고리즘의 전체 순서도
Fig. 9 Overall flowchart of FPNN algorithm

$$y = (1 + x_1^{0.5} + x_2^{-1} + x_3^{-1.5})^2 \quad (33)$$

이 함수는 Sugeno[3]가 제안한 퍼지 추론 및 파라미터 동정 기법에 적용된 함수로 Sugeno[19]가 사용한 40개의 입력 출력 데이터를 본 연구에 적용함으로써 제안된 모델의 타당성 및 정확도를 기존 모델[2,3,6]과 비교 검토한다. 모델의 성능지수는 식 (11)을 이용하며, 40개의 데이터 중 20개는 학습 데이터로, 나머지 20개는 테스트 데이터로 사용한다.

FPNN의 후반부에 사용된 PNN은 주어진 비선형 함수와 같이 입력변수가 적은 모델에 대하여 적용에 한계가 있다. 비선형 함수는 전체 입력 수가 3입력으로 PNN 노드 입력의 선택을 2입력으로 제한하기 때문이다. 이 때문에 PNN 구조는 기존의 GMDH[1,2]와 같고, 적은 입력변수의 선택으로 좋은 결과를 얻지 못한다. 또한 1층 노드의 입력으로 2입력을 선택할 경우, 1층 출력 노드의 수는 $3!/((3-2)!2!)=3$ 개를 얻게 되고 다음 층 역시 같은 조합 수를 얻게된다. 따라서 노드 입력변수의 다양한 변화를 줄 수 없다. 이러한 문제를 해결하고 좋은 성능의 모델 구조를 얻기 위해 PNN의 전반부로 FNN을 적용한 FPNN을 이용한다.

표 1은 FNN과 4입력 PNN의 결합인 FPNN 구조의 성능지수를 보여준다. 여기서 PI는 학습 성능지수를, E_PI는 테스트 성능지수이며 각 층에서 최소가 되는 노드의 성능지수를 보여주고 그때의 입력 노드 번호를 '노드'로 나타내었다. 즉 4입력 Type 2에서 5층 출력 결과는 4층의 2, 12, 14, 29번 노드가 입력으로 선택되었을 때 최소의 오차를 가지게 된다. 표 1의 FPNN은 그림 6(a)의 연결 구조를 가진다. 그림 6(b)의 연결 형태를 가진 구조는 PNN 구조와 마찬가지로 입력변수의 제한을 받기 때문에 좋은 결과를 얻지 못한다

다. 비선형 함수에 대한 Case 1 구조의 FNN은 유전자 알고리즘에 의해 학습을 및 모멘텀 계수 그리고 멤버쉽함수의 파라미터를 동조한 것으로 PI=5.219, E_PI=5.144의 성능지수를 갖는다. 결과에서 보여지는 것과 같이 FPNN은 FNN보다 월등한 정확도를 가지고 있음을 알 수 있다.

표 1 FPNN Case 1 구조의 성능지수

Table 1 Performance index of FPNN Case 1

| 후반부 층 | Type2 | | | Type3 | | |
|-------|------------|--------|--------|-------------|--------|--------|
| | 노드 | PI | E_PI | 노드 | PI | E_PI |
| 1 | 1 3 5 6 | 1.3604 | 2.5223 | 2 3 5 6 | 4.5946 | 6.313 |
| 2 | 3 9 10 13 | 0.9436 | 1.3275 | 2 7 9 11 | 1.4411 | 1.9452 |
| 3 | 2 13 17 20 | 0.2993 | 0.7189 | 2 7 27 29 | 0.5199 | 1.062 |
| 4 | 8 9 13 30 | 0.2993 | 0.6152 | 3 9 19 22 | 0.352 | 0.7139 |
| 5 | 2 12 14 29 | 0.2993 | 0.5554 | 12 22 25 30 | 0.3332 | 0.4997 |

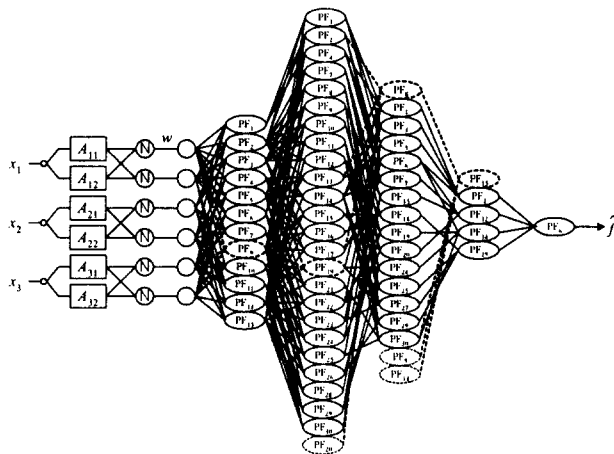


그림 10 4입력 Type 2 BFPNN 구조

Fig. 10 BFPNN structure with 4 input-Type 2

그림 10은 FNN과 4입력 Type2의 PNN이 결합한 FPNN의 Case 1 구조를 보여주며 이때 PI=0.2993, E_PI=0.5554이다. 여기서 PF는 회귀다항식을 말하며 번호는 선택된 노드의 번호이다. 점선으로 표시된 노드 중 진하게 표시된 노드는 최적의 PNN의 네트워크를 생성하는 과정에서 제외될 수 있지만 각 층에서 최소값의 테스트 성능지수를 가진 노드이며 일반 점선 노드는 전체 네트워크 형성에서 사용되지 않은 노드를 나타낸다. PNN은 FNN의 출력층을 형성하며 모델의 성능을 향상시키고 적절한 FNN 출력노드를 선택한다. FNN은 PNN의 입력층으로, 입력변수들을 확장함으로써 입력변수가 적은 모델에 대한 PNN의 단점을 보완하였고 다양한 입력선택에 따른 FPNN을 설계할 수 있다. 또한 입력변수들을 표준화 시킴으로써 PNN 입력변수들의 관계를 돕고 모델 출력을 개선시킨다.

FPNN Case 2 구조의 성능지수를 표 3에 보였다. 여기서 FNN은 각 입력에 대해 2개의 멤버쉽함수를 사용하였고, 8개의 규칙을 가진다. FNN은 유전자 알고리즘을 이용하여 동조하였으며, 이때 PI=3.997, E_PI=3.269이다. PNN은 3입력 Type 2, 4입력 Type 3의 구조를 가진다. 본 논문에서는 최적의 FPNN 구조로 4입력 Type 3의 구조를 선택하였으며 5층까지 전개하였을 때 PI=0.1164, E_PI=0.3601의 결과를 얻었고, 그림 11과 같은 구조를 갖는다.

표 2 FPNN Case 2 구조의 성능지수

Table 2 Performance index of FPNN Case 2

| 후반부 층 | 3입력 Type2 | | | 4입력 Type3 | | |
|-------|-----------|---------|---------|------------|---------|---------|
| | 노드 | PI | E_PI | 노드 | PI | E_PI |
| 1 | 3 4 5 | 8.3343 | 7.8198 | 4 5 6 7 | 3.2335 | 8.0021 |
| 2 | 12 13 27 | 1.1951 | 4.2742 | 1 7 16 22 | 0.80591 | 3.1963 |
| 3 | 14 15 21 | 0.59261 | 1.8477 | 2 10 15 23 | 0.55928 | 1.3745 |
| 4 | 1 9 25 | 0.40008 | 1.0157 | 9 16 23 27 | 0.25358 | 0.53708 |
| 5 | 3 9 15 | 0.2046 | 0.94716 | 3 4 16 18 | 0.11644 | 0.36015 |

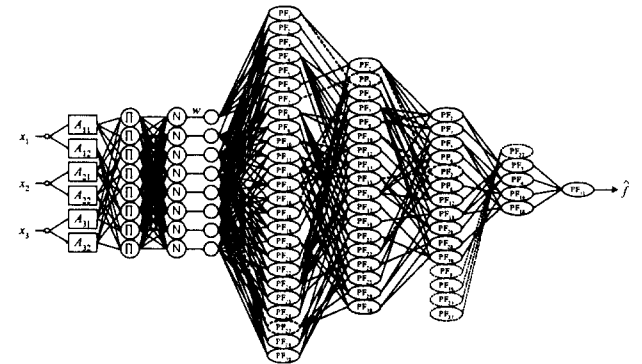


그림 11 4입력 Type 3 MFPNN 구조

Fig. 11 MFPNN structure with 4 input-Type 3

PNN의 전반부에 FNN을 이용함으로써 변수 확장과 같은 효과를 얻었으며, 비선형적인 입력 데이터들간의 상호 연관성을 찾아 그림 1(b)와 같이 입력 공간을 분할하여 각 입력 변수들의 관계를 나타내는 퍼지규칙의 적합도를 생성한 FNN을 적용함으로써 FPNN의 Case 1 구조보다 향상된 결과를 얻을 수 있었다.

본 논문에서 제안된 모델은 표 3에서 기존에 제안된 모델과 성능지수를 비교하였다. Sugeno의 퍼지모델 I, II는 다차원 추론방식에 기초한 퍼지모델이고 Shin-ichi의 모델은 신경회로망의 학습기능을 이용하여 퍼지규칙을 동정한 방법이다. 표에서 보여지는 것처럼 제안된 모델이 가장 작은 동정오차를 가지며, 또한 FPNN Case 1과 Case 2 구조의 비교로부터 입력에 대해 독립적인 공간분할 보다는 입력변수들의 상호관계를 고려한 공간분할이 보다 효과적임을 알 수 있다.

표 3 기존 모델방법과의 동정오차 비교

Table 3 Comparison of identification error with previous modeling methods

| 모델 | | PI | E_PI |
|------------------|------------|--------|--------|
| 선형모델[19] | | 12.7 | 11.1 |
| GMDH[2,19] | | 4.7 | 5.7 |
| Sugeno의 모델[3,19] | Fuzzy모델 I | 1.5 | 2.1 |
| | Fuzzy모델 II | 1.1 | 3.6 |
| Shin-ichi의 모델[6] | FNN Type 1 | 0.84 | 1.22 |
| | FNN Type 2 | 0.73 | 1.28 |
| | FNN Type 3 | 0.63 | 1.25 |
| 제안된 모델 (FPNN) | Case 1 | 0.2993 | 0.5554 |
| | Case 2 | 0.1164 | 0.3601 |

3.2 교통 경로 선택

도로망에 대한 교통량배분은 종래부터 교통계획 및 공학에 있어서 실무자 및 연구자의 관심을 불러일으키고 있는 분야중의 하나이다. 이에 대한 연구는 1950년대부터 산업경제의 발전에 따라 미국, 유럽, 일본 등 선진국을 중심으로 사람 및 물동량 수송을 위해 수많은 고속도로가 계획 설계 건설되게 되어, 경쟁관계에 있는 도로간 즉 2개 경로 또는 다수 경로간에 어떻게 적절하게 교통량배분하여 원활한 수송시스템을 구축할 것인가가 교통계획에 있어 큰 문제로 대두되어 일반도로에서 고속도로로, 또는 고속도로에서 일반도로로의 교통량전환 모델에 대한 연구가 진행되게 되었다. 이것에 대한 기본개념은 all or nothing법이 이용되고 있는데 이것은 임의의 기종점(OD) 교통량은 가장 짧은 경로로 흐른다는 것을 배경으로 하고 있다.

교통량배분 이론으로써 최초로 제안한 것은 J. G. Wardrop[11]의 교통균형원칙이다. 이 교통균형원칙이 오늘날의 교통배분이론의 기초가 되었다. 그후, 교통배분에 관한 연구는 佐佐木[12]의 수송계획법에 의한 접근을 시작으로 星野[13]의 연립방정식에 의한 추계법이 제안되었고, 등시간 원칙[13]을 도입한 정식화 방법도 제안되었다. 1960년대에 이르러서는 전자계산기의 발달에 따라 실제의 도로망을 대상으로한 simulation 방법에 의해 도로망의 교통류를 재현하는 수법이 제안되기도 하였다. 그후 분할배분법[14]에 의해 최단경로만이 아닌 다경로 교통배분법이 개발되었다. 이러한 여러 방법은 Wardrop의 등시간 원칙에 바탕한 Network의 균형 흐름을 나타내는 것이다.

Wardrop의 등시성원칙에서는 교통이용자의 선택행동에 대해서 다음과 같은 전제조건을 설정하고 있다. 1)모든 이용자는 동일한 기준을 가지고 독립적으로 행동한다. 2)교통이용자는 완전한 정보를 파악하고 있다. 3)모든 이용자는 항상 소요시간을 최소화하는 경로를 선택한다. 그러나 기존의 교통시스템에 있어서는 교통이용자가 완전한 교통정보를 입수하는 경우나 항상 기종점(OD)간의 확정적인 최단경로를 선택하는 경우가 현실의 현상에 따르는 지에 대한 의문이 많이 제기되고 있다.

이러한 배경으로부터 본 연구에서는 현실의 교통시스템에 있어서는 교통이용자의 경로선택행동시 존재하는 선택행동 지식과 그것에 존재하는 퍼지성에 근거하여 교통흐름 상에서의 경로선택 행동모델화를 시도하였다. 본 연구의 중요한 관점은 불완전정보하에서의 도로 이용자의 경로선택행동을 도로 이용자의 경로선택 의사결정 규칙과 의사결정 요인 속에 포함된 퍼지성을 고려하여 모델화를 시도하는 것이다. 즉, 도로 이용자는 경호의 실제의 소요시간 및 도로에 대한 각종 정보 등을 정확히 알지 못하고 과거의 주행경험 또는 습관 등을 바탕으로 경로선택행동을 한다는 것을 가정으로 한다. 이러한 접근은 기존의 통계적인 데이터처리를 기본으로 하는 함수 모델과는 구조적 또는 기능적으로 다른 관점에서의 시도이다. 이를 위해 본 연구에서는 FPNN 모델을 구축하고 모델에 대한 평가를 시도하였다. 그림 12는 본 논문에서 실행 대상으로 한 경쟁관계에 있는 2개의 경로를 나타낸 것이다. 경로선택의 모델화는 교통수단 및 교통로의 상이, 그리고 경쟁 경로 수 등에 의해 다양하게 전개되어지

지만 논문에서는 목적함수의 주요 요소로 시간비용과 교통비용만을 생각한다. 모델의 성능지수는 식 (12)에서처럼 정의한다. BL 모델에 대한 샘플 데이터 [7]을 이용하고 두 입력 T1과 T2는 경로 1과 경로 2에 대한 교통통행 비용이다.

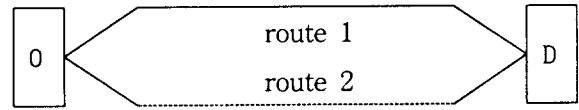


그림 12 경로선택의 간단한 예

Fig. 12 Simple Example of Route Choice

교통 경로 선택 모델에 대해 FPNN의 Case 2 구조만을 고려한다. Case 1 구조는 비선형 함수 경우와는 달리 교통 경로 선택 모델의 적은 입력변수 수에서 좋은 결과를 얻지 못한다. 따라서 입력변수 확장이 용이하고, 입력변수 간의 상호 관계를 고려한 Case 2 구조에서 뚜렷한 성능 개선을 나타낸다.

FPNN의 전반부에 이용된 FNN은 유전자 알고리즘에 의해 학습을 및 모멘텀 계수 그리고 멤버쉽함수의 파라미터를 동조한 것으로, 멤버쉽함수가 각 입력에 대해 3, 2일 때 즉 규칙이 6개일 때 PI=0.905, E_PI=0.893이며 3, 3으로 규칙이 9개일 때 PI=0.086, E_PI=0.109의 성능지수를 가진다. 표 4는 6개 규칙을 가지는 FNN과 PNN이 결합한 FPNN의 성능지수를 보여주고 있으며 표 5는 9개의 규칙을 가진 FPNN의 성능지수이다.

표 4 6개의 규칙을 가진 FPNN의 성능지수

Table 4 Performance index of FPNN with 6 rules

| 후반부 층 | 2입력 Type 2 | | | | 3입력 Type 3 | | | | |
|----------|------------|-----------|--------|---------|------------|-----------|----|----|------|
| | 노드 | PI | E_PI | 노드 | PI | E_PI | 노드 | PI | E_PI |
| 1 | 1 2 | 1.5047 | 2.2759 | 2 3 4 | 8.743e-14 | 0.6648 | | | |
| 2 | 1 11 | 1.369e-05 | 0.5681 | 2 4 7 | 1.321e-13 | 0.5166 | | | |
| 3 | 5 7 | 2.799e-09 | 0.7388 | 3 15 19 | 2.437e-12 | 0.0126 | | | |
| 4 | 13 22 | 3.881e-09 | 0.0960 | 2 12 14 | 3.122e-11 | 0.0016 | | | |
| 5 | 1 26 | 4.503e-08 | 0.0385 | 4 15 19 | 5.175e-11 | 5.964e-05 | | | |

표 5 9개의 규칙을 가진 FPNN의 성능지수

Table 5 Performance index of FPNN with 9 rules

| 후반부 층 | 2입력 Type 2 | | | | 3입력 Type 1 | | | | |
|----------|------------|-----------|-----------|----------|------------|--------|----|----|------|
| | 노드 | PI | E_PI | 노드 | PI | E_PI | 노드 | PI | E_PI |
| 1 | 4 7 | 4.2523 | 4.1833 | 3 6 8 | 0.8455 | 0.8526 | | | |
| 2 | 2 8 | 1.8047 | 2.0548 | 15 25 29 | 0.0746 | 0.0678 | | | |
| 3 | 13 28 | 0.0230 | 0.0139 | 19 21 28 | 0.0152 | 0.0412 | | | |
| 4 | 3 25 | 5.964e-05 | 9.538e-05 | 23 27 30 | 0.0027 | 0.0254 | | | |
| 5 | 9 15 | 2.210e-09 | 2.837e-09 | 3 10 12 | 0.0013 | 0.0228 | | | |

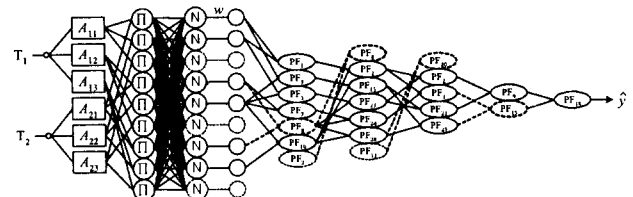


그림 13 9개 규칙을 가진 2입력 Type 2의 FPNN 구조

Fig. 13 FPNN structure with 2 input-Type 2 in 9 rules

그림 13은 9개의 규칙을 가진 FNN과 2입력 Type 2의 PNN이 결합한 FPNN 구조를 선택하였을 때 최적으로 구축된 모델의 구조이다. 표 4, 5와 그림 13에서 알 수 있듯이 제안된 모델은 PNN의 전반부에 FNN을 이용함으로써 변수 확장과 같은 효과를 얻었으며, 비선형적인 입력 데이터들간의 상호 연관성을 찾아 그림 1(b)와 같이 입력 공간을 분할하여 각 입력변수들의 관계를 나타내는 퍼지규칙의 적합도를 생성한 FNN을 적용함으로써 보다 좋은 학습 및 테스트 성능지수를 얻었다. 표 6은 기존의 모델링과 본 논문에서 제안한 FPNN과의 동정오차를 비교하여 제안된 방법이 우수한 특성을 나타냄을 보였다.

표 6 기존 모델방법과의 동정오차 비교

Table 6 Comparison of identification error with previous modeling methods

| 모델 | PI | PI | E_PI |
|------------------------------|------------|-----------|-----------|
| BL(Binary Logit) 모델[7] | 5.452 | | |
| PS(Production System) 모델[15] | 2.0 | | |
| NN(Neural Networks) 모델[15] | 0.497 | | |
| FNN 모델[15] | 1.178 | | |
| FNN 모델[20] | 0.000493 | | |
| Fuzzy 모델[18] | Simplified | 2.227 | 2.147 |
| | Linear | 0.827 | 3.268 |
| Complex+Fuzzy 모델[18] | Simplified | 0.01 | 1.397 |
| | Linear | 0.009 | 0.001 |
| GA+FNN 모델[21] | | 1.001 | 1.002 |
| HCM+GA+FNN 모델[21] | | 0.858 | 1.409 |
| 제안된 모델(FPNN) | | 2.210e-09 | 2.837e-09 |

4. 결 론

본 논문에서는 복잡한 비선형 시스템의 최적 모델동정을 위하여 FNN 및 PNN의 상호결합에 의한 FPNN 구조의 최적 설계절차와 관련된 새로운 방법론을 제안하였다. 제안된 FPNN 전반부 구조로써 FNN은 입력변수들의 상호 관계를 고려하지 않고 각 입력에 대해 분리된 퍼지공간을 사용한 FNN과 입력 데이터들 간의 상호 연관성을 고려하여 입력공간을 분할한 FNN의 두 구조를 이용하였다. 전자는 다른 비선형 모델에 비해서 학습속도가 빠르고 수렴특성이 우수하다는 장점을 가졌으나, 입력변수들의 상호 관계를 고려하지 못하였고, 후자는 이를 개선한 형태로 공정의 비선형 정도와 상호 입력변수의 상관성을 고려하여 모델의 성능을 개선할 뿐만 아니라 PNN 구조와 결합시 전자의 구조에 비해 적은 입력변수를 보다 효과적으로 사용할 수 있다. 따라서 이들 전반부 두 구조로부터 제기된 문제에 대해 가장 좋은 선택 구조를 결정하도록 전체 FPNN 구조 최적 설계에 대한 적용 능력을 보다 향상시켰다. PNN은 입력변수의 다양한 변화를 위해 4입력 이상을 요구하기 때문에 적은 입력을 가진 모델에 적용하기 힘들다. 그러나 PNN의 입력층으로 이용된 FNN은 입력변수가 적은 모델에 대해 입력변수를 확장함으로써 여러형태의 PNN 구조를 적용할 수 있도록 도와주며, PNN 모델의 성능을 개선한다. 여기서 입력변수의 확장은 멤버함수 수를 조절함으로써 가능하다. 후반부 구조로 이용된 PNN 구조의 특징은 복잡한 비선형계의 근사 동정과

예측이 가능하고, 입력변수의 선택방법, 다항식 차수와 입력 변수 수에 의한 부분표현식의 표현방법, 각 네트워크 구조의 선택방법 등에 따른 최적 구조 설계를 통하여 주어진 시스템 특성에 맞게 적응능력을 향상시킨 구조를 생성할 수 있다는 것이다. 이 구조는 기존의 뉴럴네트워크 구조와는 달리 네트워크 각 층의 노드가 고정되어 있지 않고 모델 특성에 따라 노드의 생성 및 소멸과정을 거쳐 최적의 네트워크를 형성한다. PNN 구조의 각 노드는 1차, 2차, 3차 다항식과 고계다항식을 사용하고, 노드의 입력 수는 다항식에서 사용되는 입력변수 수에 의존하므로 고계다변수 PNN 구조를 만들 수 있다. 이 PNN 입력층의 각 노드는 FNN 구조의 중간 혹은 출력층의 각 노드에 연결된다. 이 두 형태의 연결 방법을 고려함으로써, 다입력 변수 및 고계 다항식을 가진 PNN 구조를 보다 효과적으로 이용할 수 있다. 이러한 다양한 특성의 FPNN 구조 설계에 의한 모델의 표현과 적응능력으로부터 복잡한 다변수 비선형계의 시스템에 대해 우수한 동정 및 예측성능이 가능하였다.

감사의 글

이 논문은 1998년도 한국학술진흥재단의 연구비에 의하여 지원되었음(KRF-98-001-01048).

참 고 문 헌

- [1] A. G. Ivahnenko, "The group method of data handling : a rival of method of stochastic approximation," Soviet Automatic Control, Vol. 13, No. 3, pp. 43-55, 1968.
- [2] T. Kondo, "Revised GMDH algorithm estimating degree of the complete polyomial," Tran. of the Society of Instrument and Control Engineers, Vol. 22, No. 9, pp. 928-934, 1986.
- [3] G. Kang and M. Sugeno, "Fuzzy modeling," Tran. of the Society of Instrument and Control Engineers, Vol. 23, No. 6, pp. 106-108, 1987.
- [4] I. Hayashi and H. Tanaka, "The Fuzzy GMDH algorithm by possibility models and its application", Fuzzy Sets and Systems, 36, pp.245-258, 1990.
- [5] H. Takagi and I. Hayashi, "NN-driven Fuzzy Reasoning," Int. J. of Approximate Reasoning, Vol. 5, No. 3, pp. 191-212, 1991.
- [6] Shin-ichi Horikawa, Takeshi Furuhashi, and Yoshiki Uchigawa, "On Fuzzy Modeling Using Fuzzy Neural Networks with the Back Propagation Algorithm," IEEE trans. Neural Networks, Vol. 3, No. 5, pp. 801-806, 1992.
- [7] 森彬壽旁, "非集計モデルの推定と検定", 日本土木計画法講習會テキスト15, PP. JSCE, 1984.
- [8] H. Nomura and Wakami, "A Self-Tuning Method of Fuzzy Control by Descent Methods," 4th IFSA World Conference, pp. 155-159, 1991.

[9] Takeshi Yamakawa, "A New Effectie Learning Algorithm for a Neo Fuzzy Neuron Model," 5th IFSA World Conference, pp. 1017-1020, 1993.

[10] S. S. Kim, "A Neuro-fuzzy Approach to Integration and Control of Industrial Processes : Part I," Journal of Fuzzy Logic and Intelligent Systems, Vol. 8, No. 6, pp. 58-69, 1998.

[11] Wardrop, J, G. Some Theoretical Aspect of Road Traffic Research, Proc. of the Institution of Civil Engineers, Part I, Vol. 1, pp. 325~378, 1952.

[12] 佐左木 綱 : 道路網における輸送計劃について, 第4回日本道路會議論文集, pp.43~46, 1957.

[13] 星野 哲三 : 道路網における交通量配分の理論, 제5回日本道路會議論文集 pp.68 sim 70, 1959.

[14] Martin, B. V. and Manheim, M. L., A Research Program for Comparison of Traffic Assignment Techniques, Highway Research Record 88, pp. 69-84, 1965.

[15] 今崎直樹, 木治潤一, "ファジィルール構造化ニューラネットワーク", 日本ファジィ學會誌 Vol.4, No.5, pp. 985-995, 1992.

[16] David E. Goldberg, Genetic Algorithms in search, Optimization & Machine Learning, Addison-wesley, 1989.

[17] Zbigniw Michalewicz, Genetic Algorithms+Data Structure=Evolution Programs, Springer-Verlag, 1992.

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

[19] 菅野道夫(譯:박민용, 최항식), "퍼지제어 시스템," pp. 143-158, 大英社, 1990.

[20] 오성권, 노석범, 남궁문, "퍼지-뉴럴 네트워크 구조에 의한 비선형 공정시스템의 지능형 모델링," 한국퍼지및 지능시스템학회논문지 제5권 제4호, pp. 41-55, 1995.

[21] 오성권, 윤기찬, "유전자 알고리즘과 합성 성능지수에 의한 최적 퍼지-뉴럴 네트워크 구조의 설계," 제어자동화시스템공학논문지, 6권, 3호, pp. 273-283, 2000.

[22] 박병준, "다항식 뉴럴네트워크 및 퍼지추론에 기초한 퍼지 다항식 뉴럴네트워크 구조에 관한 연구," 원광대학교 대학원, 석사학위논문, 1999.

[23] Sung Kwun Oh, Tae Chon Ahn, Witold Pedrycz, "Fuzzy Polynomial Neural Networks-Based Structure and Its Application to Nonlinear Process Systems," 7th IFSA World Conference, Vol. 2, pp.495-499, 1997.

[24] H. S. Park, S. K. Oh, T. C. Ahn, and W. Pedrycz, "A Study on Multi-layer Fuzzy Polynomial Inference System based on an Extended GMDH algorithm," 1999 IEEE International Fuzzy Systems Conference Proceedings, Vol. 1, pp. 354-359, 1999.

[25] 오성권, 김동원, 박병준, "다항식 뉴럴네트워크 구조의 최적 설계에 관한 연구," 대한전기학회논문지, 49권, 3호, pp. 145-156, 2000.

[26] 오성권, "퍼지모델 및 제어이론과 프로그램," 기다리 출판사, 1999년 3월.

저 자 소 개



박 병 준 (朴 炳 俊)

1972년 12월 1일생. 1998년 원광대 제어계측공학과 졸업, 2000년 동 대학원 제어계측공학과 졸업(석사). 2000년~현재 동 대학원 제어계측공학과 박사과정. 주관심분야는 시스템자동화, 퍼지이론 및 신경회로

망 응용, 지능형 모델링 및 제어 등

Tel : 0653-850-6342

E-mail : lcap@autosys.wonkwang.ac.kr



오 성 권 (吳 聖 權)

1957년 2월 13일생. 1981년 연세대 전기공학과 졸업. 1983-89년 금성산전연구소(선임연구원). 1993년 연세대 대학원 전기공학과 졸업(공학박). 1996-97 캐나다 Manitoba 대학 전기 및 컴퓨터공학과 Post-Doc.

1993-현재 원광대 제어계측공학과 부교수. 주관심분야는 시스템자동화, 퍼지이론 및 신경회로망 응용, 지능형 모델링 및 제어 등

Tel : 0653-850-6342

E-mail : ohsk@wonms.wonkwang.ac.kr



김 현 기 (金 賢 起)

1955년 2월 26일생. 1977년 연세대 전기공학과 졸업. 1985년 동 대학원 전기공학과 졸업(석사). 1991년 동 대학원 졸업(공학박). 1997-현재 경기지역 산학연 기술지원단 단장. 1998-현재 산업기술정책연구소 심의

위원. 2000-현재 사단법인 산학연 컨소시엄 전국협의회 회장. 1989-현재 수원대 전기전자정보통신공학부 교수. 주관심분야는 시스템자동화 및 감시제어, 지능형모델링 및 제어 등

Tel : 0331-220-2300

E-mail : hkkim@mail.suwon.ac.kr