Optimal Model Design of Software Process Using Genetically Fuzzy Polynomial Neuray Network

In-Tae Lee, Sung-Kwon Oh, Hyun-Ki Kim
Department of Electrical Engineering, University of Suwon

Abstract - The optimal structure of the conventional Fuzzy Polynomial Neural Networks (FPNN)[3] depends on experience of designer. For the conventional Fuzzy Polynomial Neural Networks, input variable number, number of input variable, number of Membership Functions(MFs) and consequence structures are selected through the experience of a model designer iteratively. In this paper, we propose the new design methodology to find the optimal structure of Fuzzy Polynomial Neural Network using Genetic Algorithms(GAs)[4, 5]. In the sequel, it is shown that the proposed Advanced Genetic Algorithms based Fuzzy Polynomial Neural Network(Advanced GAs-based FPNN) is more useful and effective than the existing models for nonlinear process. We used Medical Imaging System(MIS)[6] data to evaluate the performance of the proposed model.

1. 서론

많은 분야에서 행해지고 있는 시스템 모델링에 관한 연구의 궁극적 목표는 자연현상을 이해하고, 분석함으로써 원하는 정보를 이용하는데 있다. 그러나 대부분의 현상들은 비선형으로 나타내고 있기 때문에 선형성을 바탕으로 개발된 기존의 방법들을 이용해 이들 해석하기에는 많은 어려움에 부딪히게 된다. 현재까지 개발된 모델들은 입력변수와 증가하면서, 이러한 현상이 복잡하고 복잡한 시스템으로 표현하는데 한계가 있다. 이러한 문제를 해결하기 위해, A. G. Ivakhnenko는 Group Method of Data Handling(GMDH) [1] 알고리즘을 사용하였다. 그러나 GMDH는 비교적 간단한 시스템에 대해서 매우 복잡한 다항식을 생성하는 경우가 있으므로, 입력 2차 회귀다항식의 형태로 복잡한 비선형 시스템을 표현하는데는 어려움도 한계가 있다. 이를 극복하기도, Oh 등에 의해 자기조성 다항식 뉴럴네트워크(Self-Organizing Polynomial Neural Networks : SOPNN) [2]와 자기조성 회귀다항식 뉴럴네트워크(Self-Organizing Fuzzy Neural Networks : SOFPNN)[3]가 제안되었다. SOFPNN(또는 FPNN)의 구조는 일반적인 다층 네트워크 구조와 달리 미리 정해져 있지 않고, 이에 의한 학습이 이루어지는 동안 입력 신호로 노드의 선택과 제거를 통해 최적의 퍼포머 구조를 생성할 수 있다. 그러나 FPNN 구조는 선택자에 의존 또는 반복적인 학습을 통해 노드의 입력변수 수와 평균 후보 다항식 차수 입력변수 그리고 회귀의 계수를 선택하였기에 입력의 네트워크 구조를 결정하는데 어려움이 내재되어 있었다. 따라서 본 논문에서는 FPNN을 최적화 시키기 위해, 탐색방법론을 사용하여 출력값을 얻으면서, 그에 따른 입력변수 그리고 규칙 후보 다항식의 차수와도 관련형식isation 함수의 중국을 탐색한다. 가로 포트에 의해 자동화된 Process 모델의 정확성 및 예측 성능들을 기존 지능 모델들과 비교하여 그 우수성을 보인다.

2. Advanced GAs-based FPNN

2.1 유전자 알고리즘에 의한 구조 정의

본 논문에서 이용한 유전자 알고리즘은 한 개의 탐색 변수와 또한 변량 함수의 값을 담은 적절 방식을 사용하여 선택된 상수는 Roulette-Wheel 방식을, 교차현상을 유도하는 One-crossover방식을 사용하였고, 또한 돌연변이 현상으로는 Invert 방식을 사용하여 최적화의 FPNN 구조를 결정하였다. 그 설계과정은 다음과 같다.

[단계 1] GAs-based FPNN 구조를 구축하는데 있어서 기본적인 정보를 결정한다.
[단계 2] GAs-based FPNN 모델을 구축하는데 기본 유전자 Fuzzy Polynomial Neuron(FPNN)의 유전자를 알고리즘을 통하여 최적 구조를 선택한다.
[단계 2-1] 입력변수의 수 선택
[단계 2-2] 자기조성 후보 다항식 차수 선택
[단계 2-3] 입력변수 선택
[단계 2-4] 맵 Beckham 함수의 수 결정

[단계 2-1]에서 단계 2-3까지 수행하여 산출된 입력변수의 수와 자기조성 후보 다항식 차수, 입력변수의 수에 따른 입력변수 그리고 맵 Beckham 함수의 수를 이용하여 퍼포머 구조를 선택한다.

<p>| 표 1. 자기조성 후보 다항식의 서로 다른 형태 |</p>
<table>
<thead>
<tr>
<th>(Type 1)</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Type 2)</td>
<td>Linear</td>
<td>Constant</td>
<td>Polynomial</td>
</tr>
<tr>
<td>Quadratic</td>
<td>Bi-quadratic</td>
<td>Modified</td>
<td>Trilinear</td>
</tr>
<tr>
<td>Bi-quadratic</td>
<td>Modified</td>
<td>Bi-quadratic</td>
<td>Trilinear</td>
</tr>
</tbody>
</table>

i) 간략 추론

후보가 단일의 상수함만을 가지는 것으로 이와 같은 추론을 간단 추론법이라 한다.

\[ R \subseteq \{ x_1 = A_{11}, \ldots , x_n = A_{n} \} \]

(1)

\[ f(x) = \sum_{i=1}^{n} w_i(x) \]

(2)

\[ w_i = A_{i}(x_i) \wedge \ldots \wedge A_{n}(x_n) \]

\[ y_i = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_j} = \frac{\sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} w_
ii) 회귀항성 추론
후반부가 1, 2, 변형된 2차 항항식 형태로 표현된 것으로 회귀항성 추론이이라 한다. 이 경우의 피치 모델은 식(3)과 같은 형태로 구성된다.

\[ R^2 : \text{if } x_i = A_x \text{ and } \ldots \text{ and } x_n = A_n \text{ then } y = f(x_1, \ldots, x_n) \]  
(3)

2.2 유전자 알고리즘을 이용한 FPNN의 설계
[단계 3] 모델 구축과 테스트 그리고 노드의 선택
Step 1) GAs-based FPNN 모델 구축을 위한 유전자의 세대, Population 개수, 돌연변이율, 교배률, 그리고 엽색체의 길이와 같은 유전자의 정보를 설정한다.
Step 2) 첫 번째 세대에서 설정한 population 개수만큼 엽색체에 의한 노드(FPNN)을 구축한다.
Step 3) 테스트 데이터를 이용해 구해진 성능지를 구구자 알고리즘의 적합도 함수로 계산하기 위해서 다음식(4)과 같이 변환하여 유전자 알고리즘의 적합도 함수로 사용한다.

\[ F(\text{Fitness Function}) = 1/(1+EPF) \]  
(4)

Step 4) 다음 세대의 생성을 위해서 구해진 개체별 정 보와 적합도 값을 이용하여 선택연산, 교배, 돌연변이를 실행한다.
Step 5) 구해진 적합도 값(F_n, F_m, ..., F_p)을 선호하는 적합도 값 순(내림차순)으로 정렬한다.
Step 6) 정렬된 적합도 값을 중에서 최저인 적합도 값들은 제거하고 나머지 적합도 값을 사용한다.
Step 7) 중복된 적합도 값을 하나로 처리한 후 설계 자가 FPNN 구조로 설계하기 전 결정된 노드의 재헌한 개수 W에 만족할 때까지 선택을 반복한다.
Step 8) Step 7에서 선택된 노드들을 다시 초기 population 번호 순서에 따른 내림차순으로 재정렬한다.
Step 9) 다음 세대에 엘리트 선택을 위해 선택된 노드 들 중에서 최저의 적합도 값을 가지는 노드를 선택한다.
Step 10) Step 4에서 얻은 정보를 가지고 다음 세대의 population을 생성 후 Step 2에서 Step 9까지 반복한다.
Step 11) 이전 세대에서 구해진 적합도 값을과 현재 세 대에서 구해진 적합도 값을 결합하여 다시 Step 5) 부터 Step 8)까지 반복한다.
Step 12) 설계한 세대까지 Step 10)부터 Step 11)을 반복 실행한다.
[단계 4] [단계 4]의 현재 층층에서 얻어진 최적 노드의 최적 적합도 값 F_0이 다음 부동선식을 만족하는 경우에 알고리즘을 종료한다.

\[ F_0 \leq F \]  
(5)

다. F는 이전 층의 최적 노드인 최적 적합도 값이다.
본 논문에서의 모델의 성능저지수로는 다음 식을 이용하 였다.

\[ E = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 \]  
(6)

[단계 5] 현재 층층에서 보존된 노드의 출력들(z_{ih}, z_{ih}, ..., z_{im})에서 다음 층의 새로운 입력들(x_{1h}, x_{2h}, ..., x_{nh})로, x_{1h}=z_{ih}, x_{2h}=z_{ih}, ..., x_{nh}=z_{ih}로 구성하고, [단계 2]로 간다(여기서 i=1,2). 이 후, [단계 2]부터 [단계 5]까지 반복한다. 알고리즘을 종료할 경우, 마지막 층의 최적 적합도 값 F=0을 얻은 노드의 다항식에 입력이 되는 점이 없는 출력을 대입하고 동일한 조작을 5층까지 반복하여 최종 추정 모델 y를 얻는다.

3. 실험 데이터를 통한 결과 고찰
3.1 GAs과 FPNN의 기본정보
Advanced GAs-based FPNN의 성능을 테스트하기 위해 Medical Imaging System (MIS) 데이터를 이용하였다. 이 데이터는 11개의 시스템 입력값과 1개의 출력값으로 구성된 데이터로 본 논문에서 제한된 모델은 11개의 입력변수 중 최대 4개까지 선택하도록 설계하였다.
표 2는 각각의 FPNN마다 GAs 정보와 FPNN 모델 구축을 위한 정보를 나타낸 것이다.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>1층-5층</th>
<th>1층-5층</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum gen</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>Total population size</td>
<td>150</td>
<td>150</td>
</tr>
<tr>
<td>Selected population size</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>Crossover rate</td>
<td>0.63</td>
<td>0.63</td>
</tr>
<tr>
<td>Mutation rate</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>String length</td>
<td>3+3+30+5</td>
<td>3+3+30+5</td>
</tr>
<tr>
<td>polynomial(1) type</td>
<td>T (1)</td>
<td>T (1)</td>
</tr>
<tr>
<td>Membership Function(MF) type</td>
<td>Gaussian</td>
<td>Gaussian</td>
</tr>
<tr>
<td>No. of MFs per each input</td>
<td>2 or 3</td>
<td>2 or 3</td>
</tr>
</tbody>
</table>

3.2 Advanced GAs-based FPNN의 성능평가
그림 1과 그림 2는 MFs의 형태와 층의 증가에 따른 성능 저지를 나타내고 있다. 그림에서 A(4, 7, 1, 3, 2) 의 의미는 먼저, A는 Max값을 나타내고 4와 7은 Max 값에 따른 선택된 node 번호를 나타내고 레이어별로(:) 뒤 첫 번째 숫자 1은 후반부 다항식 형태를 나타내고 그 뒤 3과 2는 각 입력에 대한 MFs의 개수를 나타내고 있다.

<Training data>  <Testing data>
그림 1. Triangular MFs 임계 성능지수

<Training data>  <Testing data>
그림 2. Gaussian-like MFs 임계 성능지수

그림 1과 2를 살펴보면 층의 증가에 따라 MFs의 숫자와 후반부 탑업이 고정되지 않고 유연하게 선택될 수 있다.

<Training data>  <Testing data>
그림 3. Advanced GAs-based FPNN의 오차곡선

<Training data>  <Testing data>
그림 3. Advanced GAs-based FPNN의 오차곡선

- 2874 -
또한 입력변수수 중의 중개에 따라 고정되어 선택되는 게 아니라 가변적으로 선택될 수 있다. 그림 3은 실제 출력값과 모델 출력값의 오차곡선이다. 이 그림은 $max=3$이고 MFs의 형태가 가우시안형태의 그림으로 데이터 변호에 따른 오차값이다.

그림 4는 실제 출력값과 모델에서 구한 출력값을 비교해 주고 있다.

<Training data>  
<Testing data>

그림 4. 실제 출력값과 모델 출력값 비교

그림 5는 $max=3$이며 가우시안형태의 3층까지의 최적 네트워크 구조를 보여주고 있다. 그림에서 보듯이 각층별로 입력변수 선택에 있어서 매우 유연하며 각 입력에 대한 멤버십 개수도 유연하게 선택될 수 있다.

그림 5. Advanced GAs-based FPNN의 최적 구조

기존의 모델과 성능비교를 표 3에서 보여주고 있다. 표에서 보듯이 가우시안형태의 경우에는 $PI\%$과 $EPI\%$값이 25.804, 12.637로 성능이 우수함을 보여준다.

<table>
<thead>
<tr>
<th>Model</th>
<th>Structure</th>
<th>PI</th>
<th>EPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>SONPN</td>
<td>Simplified Generic Type Basis architecture</td>
<td>40.755</td>
<td>17.898</td>
</tr>
<tr>
<td></td>
<td>Linear Generic Type Basis architecture</td>
<td>35.748</td>
<td>17.807</td>
</tr>
<tr>
<td></td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td></td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td></td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>Our model</td>
<td>$G$</td>
<td>$G$</td>
<td>$G$</td>
</tr>
<tr>
<td></td>
<td>$G$</td>
<td>$G$</td>
<td>$G$</td>
</tr>
<tr>
<td></td>
<td>$G$</td>
<td>$G$</td>
<td>$G$</td>
</tr>
<tr>
<td></td>
<td>$G$</td>
<td>$G$</td>
<td>$G$</td>
</tr>
</tbody>
</table>

여기서, $max$는 최대로 선택될 수 있는 입력변수를 나타내고 있으며, $M_{type}$은 MFs의 형태를 $T$은 후반부 단양시 Type을 나타내고 있다.

4. 결론

본 논문에서 제안한 Advanced GAs-based FPNN 모델은 기존의 FPNN 모델의 최적 구조에 영향을 미치는 네트워크 중, FPNN의 입력변수수의 수와 후반부 차수 그리고 고정된 입력변수 선택을 설계자의 경험이 아닌 유전자 알고리즘을 사용하여 입력변수의 수와 그에 따른 입력변수, 후반부 차수 그리고 편바이스 함수들의 수를 선택하여 동조한다. 따라서 설계자의 경험이 아닌 객관적이고 유연한 네트워크 구조를 설계할 수 있다. 결과적으로 기존의 모델보다 최적의 네트워크를 찾아내는 시간이 단축되었고 모델의 성능 또한 중의 중개에 따른 유연한 네트워크 구조를 선택 동조하여 향상되었다.

감사의 글

본 연구는 산업자원부 지원에 의하여 기초전력연구원(1-2004-0-074-0-00) 주관으로 수행된 과제임.

[참고 문헌]