

論文2002-39CI-1-3

진화전략으로 학습되는 뉴로퍼지 시스템의 비선형 시스템 동정에의 응용

(Application of a Neuro-Fuzzy System Trained by Evolution Strategy to Nonlinear System Identification)

鄭 成 薰 *

(Sung Hoon Jung)

요 약

본 논문에서는 진화전략을 이용하여 빠르게 학습하는 새로운 구조의 뉴로퍼지 시스템을 제안하고 제안한 시스템의 효용성을 입증하기 위하여 비선형 시스템 동정에 응용한 결과를 설명한다. 뉴로퍼지 시스템의 학습 방법으로는 지금까지 주로 변형된 오류역전파 알고리즘과 최적화 기법인 유전자 알고리즘이 많이 사용되어왔으나, 오류역전파 알고리즘은 학습시간이 많이 걸리며 유전자 알고리즘은 해를 유전형 형태로 표현함으로써 미세한 탐색이 힘든 단점이 있었다. 본 논문에서 사용한 진화전략은 해를 표현형의 개체로 나타내어 실수형태로 진화하기 때문에 미세한 탐색이 가능하며 오류역전파 알고리즘에 비해 지역해에 빠질 가능성이 작고 속도가 빠른 장점이 있다. 제안한 뉴로퍼지 시스템을 비선형 시스템 동정에 적용한 결과 학습속도가 빠르며 학습결과도 우수함을 보였다.

Abstract

This paper proposes a new neuro-fuzzy system that is fast trained by evolution strategy and describes application results of the proposed system to nonlinear system identification to show its usefulness. As training methods of neuro-fuzzy systems, modified error back-propagation algorithms and genetic algorithms have been used so far. However, the former has some drawbacks such as long training time, falling to local optimum, and experimental selecting of learning rates and the latter has difficulty in precise searching solutions because genetic algorithms represents solutions as genotype individuals. The evolution strategy we used can do precise search because its individuals are represented as phenotype real values, it seldom falls into a local optimum, and its training speed is faster than error back-propagation algorithms. We apply our neuro-fuzzy systems to nonlinear system identification. It was found from experiments that training speed is fast and the training results were considerably good.

I. 서 론

뉴로퍼지 시스템은 구조적으로 신경망 구조를 가지나

* 正會員, 漢城大學校 情報電算學部

(School of Information and Computer Engineering, Hansung Univ.)

※ 본 연구는 2001년도 한성대학교 교내연구비 지원과제임
接受日字: 2001年5月8日, 수정완료일: 2001年10月30日

링크의 기중치와 노드의 동작이 퍼지화된 시스템으로서 패턴인식, 음성인식, 지능제어, 시계열 예측등 여러분야에 응용되어 왔다^[1~8]. 뉴로퍼지 시스템의 성능은 주로 뉴로 퍼지 시스템의 구조와 학습방법에 의존한다. 즉 뉴로퍼지 시스템의 중간노드의 수 (다른 말로 퍼지규칙의 수)가 적절하여야 하며 뉴로퍼지 링크의 퍼지 언어항(fuzzy linguistic term)과 퍼지 언어항의 소속함수(membership function)가 학습데이터에 따라 적절히 학습되어야만 한다. 학습데이터를 이용하여 적절한 중간노드의 수를 결정

하는 것을 구조학습이라 하며 구해진 구조를 이용하여 뉴로퍼지 링크의 퍼지 언어항과 퍼지 언어항의 소속함수를 조정하는 것을 파라미터 학습이라고 한다^[9,10]. 뉴로퍼지 시스템의 파라미터를 학습하기 위한 방법으로는 변형된 오류역전파 알고리즘을 사용하는 방법이 있으며^[9,10], 유전자 알고리즘을 사용하여 구조학습과 파라미터 학습을 병행하는 방법도 제시 되었다^[3,7,11]. 그러나, 변형된 오류역전파 알고리즘은 신경망 학습에서처럼 지역최적해에 빠질 수 있으며 학습시간이 많이 걸리고 적절한 학습율(learning rate)를 반복적인 실험에 의하여 선택해 주어야 하는 단점이 있다. 또한 유전자 알고리즘을 사용하는 방법은 해를 유전형(genotype)으로 표현해서 진화를 함으로서 파라미터 학습시 미세한 탐색이 어려운 점이 있다.

본 논문에서는 구조학습을 위하여 학습데이터로부터 필요한 중간노드의 수를 결정하는 새로운 알고리즘을 소개한다. 구조학습 알고리즘에서는 각 학습데이터 별로 해당 학습데이터를 위해 새로운 규칙이 필요한지 안한지를 결정하며 새로운 규칙이 필요할 경우 해당 데이터를 이용해 규칙을 만들게 된다. 해당 학습데이터를 위하여 새로운 규칙이 필요한지 안한지를 결정할 때 생성된 규칙이 없을 경우에는 해당 학습데이터를 위한 규칙을 생성하고 생성된 규칙이 있을 경우에는 해당 학습데이터와 기존 생성된 규칙을 고려해 새로운 규칙(다른 말로, 새로운 중간노드)이 필요한지 안한지를 결정하게 된다. 그러므로, 본 논문의 구조학습 알고리즘은 필요한 중간노드의 수 뿐만 아니라 학습데이터에 대한 퍼지 규칙도 구해진다. 구해진 규칙을 초기 규칙으로 사용하면 무작위적인 초기 규칙을 사용할 때 보다 학습시간이 단축되는 효과가 있다. 본 논문의 실험에서는 구조학습으로 얻어진 퍼지규칙을 초기 규칙으로 사용한다. 필요한 퍼지규칙의 수가 결정되면 해당 구조의 뉴로퍼지 시스템하에서 파라미터 학습을 수행하게 된다. 본 논문에서는 파라미터 학습 방법으로 최적화 기법으로서 학습속도가 빠르고 지역최적해에 빠질 확률이 적고 미세한 탐색이 가능한 진화전략(Evolution Strategy)을 이용한다. 진화전략은 유전자 알고리즘과 다르게 해를 유전형으로 표현하지 않고 실수형의 표현형(phenotype)을 사용함으로 미세한 탐색이 가능한 장점이 있다^[12].

본 논문에서 제안한 뉴로퍼지 시스템의 효율성을 보이기 위하여 제안한 뉴로퍼지 시스템을 비선형 시스템 동정에 응용하였다. 비선형 시스템의 동정은 제어나 시계열 예측분야에 많이 사용되는 방법으로서 근래에 뉴로퍼지를 이용

한 연구가 많이 진행되고 있는 분야이다. 실험에 사용한 비선형 시스템은 비선형성이 강한 시스템으로 기존의 논문^[5,8]에서 사용한 것을 이용하였다. 두개의 비선형성이 강한 시스템에 적용시켜 본 결과 본 논문에서 제안한 뉴로퍼지 시스템의 학습속도가 빠르면서도 학습성능이 우수함을 확인하였다.

본 논문은 다음과 같이 구성된다. 2절에서는 제안한 뉴로퍼지 시스템의 구조 및 학습방법에 대하여 다룬다. 비선형 시스템 동정에 응용한 결과는 3절에 있으며 4절의 결론으로 끝을 맺는다.

II. 제안한 뉴로퍼지 시스템

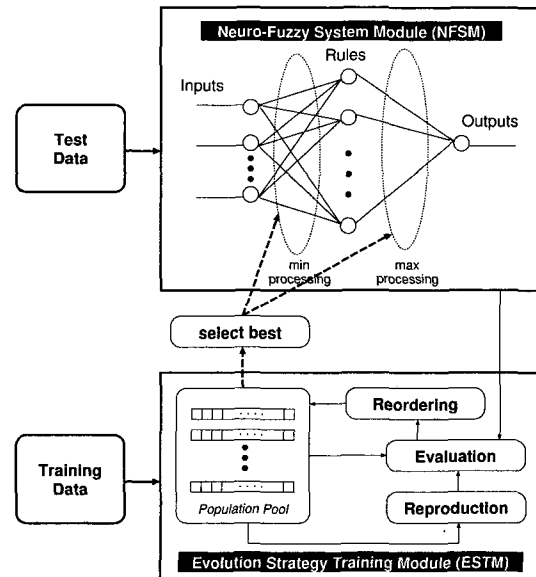


그림 1. 뉴로퍼지 시스템의 전체 구성
Fig. 1. Overall structure of neuro-fuzzy system.

본 논문에서 제안하는 뉴로퍼지 시스템의 전체적인 구성이 그림 1에 있다. 그림 1에서 보듯이 크게 뉴로퍼지 시스템 모듈(Neuro-fuzzy System Module: NFSM)과 뉴로퍼지 모듈의 학습을 담당하는 진화전략 학습 모듈(Evolution Strategy Training Module: ESTM)로 구성된다. 알고리즘 1은 뉴로퍼지 시스템의 전체적인 동작을 간략하게 기술한 것이다. NFSM은 뉴로퍼지 시스템의 핵심 모듈로서 학습후 테스트 데이터에 대한 출력을 발생한다.

ESTM은 NFSM을 학습시켜 주는 모듈로서 그 동안 신경망이나 뉴로퍼지 시스템 학습에 많이 사용된 오류 역전

Algorithm 1 뉴로퍼지 시스템 동작

- // PP : N 개의 부모개체와 N 개의 자식개체로 이루어진 개체군//
- // f : 어떤 세대에서 최대 적합도 //
- // f_t : 학습 종료 판단용 적합도 //
- 1 각 모듈을 초기화 하고 학습데이터를 설정
- 2 NFSM 을 초기화
- 3 ESTM 을 초기화
- 4 학습 데이터를 설정
- 5 ESTM을 이용하여 학습
- 6 **do**
- 7 N 개의 부모 개체를 평가
- 8 N 개의 자식 개체를 생성하기 위하여 N 개의 부모 개체를 돌연변이 시킴
- 9 N 개의 자식 개체를 평가하고 개체군 PP 에 삽입
- 10 개체군 PP 에서 적합도가 좋은 N 개를 선택하여 다음세대의 부모개체로 설정
- 11 **until** $f < f_t$
- 12 최대 적합도를 갖는 개체의 파라미터를 NFSM에 설정
- 13 테스트 데이터를 NFSM에 적용하여 출력을 발생
- 14 테스트 데이터를 NFSM에 적용
- 15 NFSM의 출력을 얻음
- 16 NFSM의 출력을 이용하여 응용에 따라 사용

파 알고리즘 (error back propagation algorithm)의 기능에 해당 하는 작업을 한다. 그러나 ESTM은 오류 역전파 알고리즘에 비하여 진화전략의 특성상 지역최적해 (local optimum)에 빠지는 경우가 거의 없고 학습이 빠른 장점을 갖는다. 이 절의 나머지 부분에서 각 모듈의 세부적인 동작에 대하여 자세히 기술한다.

그림 1에서 보듯이 NFSM 은 구조적으로 입력, 중간 및 출력노드로 구성된 신경망 구조를 갖는다. 입력노드와 중간 노드사이 그리고 중간노드와 출력노드 사이에는 완전히 연결된 링크를 갖는다. 본 논문에서 앞으로 입력노드와 중간노드 사이의 링크를 입력링크라고 부르며 중간노드와 출력노드 사이의 링크를 출력링크로 부른다. 본 논문에서는 i 번째 입력노드에서 j 번째 중간노드로 가는 입력링크의 가중치를 w_{ij} 로 표현하며 j 번째 중간노드에서 k 번째 출력노드로 가는 출력링크의 가중치를 v_{jk} 로 표현한다. 일반적인 신경망에서는 입출력 링크는 실수의 가중치를 갖게 되나 본 논문에서 제안한 NFSM 에서는 가중치가 하나의 퍼지 언어항으로 표현된다. 예를 들어, 3개의 퍼지 언어항 NB (Negative Big), ZO (Zero), PB (Positive

Big)으로 표현되는 시스템이라면, 가중치는 NB , ZO , PB 중 하나로 표현된다. 결국 중간노드로 입력되는 값은 각각의 입력이 각 입력링크에 주어진 퍼지 언어항에 얼마나 정합되는지를 나타내는 값들이 된다. 중간노드는 이 입력값에 t-norm 연산을 수행하여 출력링크로 보낸다. 뉴로퍼지 시스템이 l 개의 입력노드 $x_i, i \in \{1, \dots, l\}$, m 개의 중간노드 $h_j, j \in \{1, \dots, m\}$, n 개의 출력노드 $y_k, k \in \{1, \dots, n\}$ 로 구성되었다고 가정하고 t-norm 연산으로 가장 간단하면서도 많이 사용되는 min 연산을 사용한다면 결국 j 번째 중간노드의 출력 h_j^o 는 다음과 같다.

$$h_j^o = \min\{\mu_{w_{ij}}(x_i), i \in \{1, \dots, l\}, w_{ij} \in \{NB, ZO, PB\}\} \quad (1)$$

식 (1)에서 $\mu_{w_{ij}}(\cdot)$ 는 입력링크 w_{ij} 의 퍼지언어항에 대한 소속함수이다. 주의할 것은 각 입력노드에 들어오는 값의 최소, 최대 범위와 각 출력노드에서 출력되는 값의 최소, 최대 범위가 다를수 있기 때문에 같은 퍼지 언어항이라도 소속함수는 입출력 노드에 따라 다를 수 있다. 즉 w_{11} 와 w_{21} 이 동일한 퍼지 언어항 NB 를 가진다고 해

도 두 입력 x_1 과 x_2 의 최소, 최대 범위가 다르다면 두 입력 링크 퍼지 언어항의 소속함수 $\mu_{w_{11}}(\cdot)$ 과 $\mu_{w_{21}}(\cdot)$ 는 다르다. 그러므로, 엄밀한 의미에서 $\mu_{w_{11}}(\cdot)$ 은 $\mu_{w_{11}}^{x_1}(\cdot)$ 로 $\mu_{w_{21}}(\cdot)$ 는 $\mu_{w_{21}}^{x_2}(\cdot)$ 로 나타내어야 한다. 그러나, 소속함수의 입출력 링크 정보나 적용입출력값을 보면 어떤 입출력과 관계된 것인지 알 수 있으므로 앞으로 수식 (1)과 같이 간단하게 나타낸다.

중간노드의 출력값은 출력링크 퍼지 언어항의 α -cut으로 작용하고 출력노드에서는 α -cut된 퍼지 언어항들에 t-conorm 연산을 적용한 후 비퍼지화 과정을 거쳐서 출력값을 생성한다. 비퍼지화 방법은 무게중심법 (center of gravity) 등 다수의 방법이 제안되었는데 본 논문에서는 LGM 방법^[13]을 사용한다. LGM 방법은 퍼지 언어항 소속함수의 폭이 다를 경우에 유용한 방법으로서 NFSM에서는 학습후 각 입출력 링크 퍼지언어항의 소속함수가 폭이 다르게 되기 때문에 이 방법이 유용하다. LGM은 출력노드의 t-conorm 중 가장 간단한 max 동작과 비퍼지화 동작을 하나의 수식으로 구현한 것으로서 따로 max

동작이 필요하지 않다.

LGM 비퍼지화 방법을 사용했을 때 출력노드의 동작은 다음과 같이 수식으로 정리할 수 있다. 중간노드 j 에서 출력노드 k 로 가는 출력링크의 퍼지 언어항 소속함수를 $\mu_{v_k}(\cdot)$ 라 하고, 이 출력링크로 출력되는 중간노드의 값을 h_j^o 라고 하고, 출력 퍼지 언어항의 확실성 (measure of certainty)를 $m_c(\cdot)$ 라고 한다면 $m_c(\cdot)$ 는 다음과 같이 주어진다.^[13]

$$m_c(h_k^o) = \frac{h_j^o}{\mu_{v_k}^{-1}(h_k^o)_{\max} - \mu_{v_k}^{-1}(h_k^o)_{\min} + 1} \quad (2)$$

식 (2)에서 $\mu_{v_k}^{-1}(\cdot)_{\max}$ 와 $\mu_{v_k}^{-1}(\cdot)_{\min}$ 은 $\mu_{v_k}^{-1}(\cdot)$ 의 두 값중 큰 값과 작은 값을 의미한다. 결국 k 번째 출력노드의 출력 y_k^o 는 비퍼지화 과정을 거쳐서 다음과 같이 계산된다.

$$y_k^o = \frac{\sum_{j=1}^m \mu_{v_k}^{-1}(1) m_c(h_j^o)}{\sum_{j=1}^m m_c(h_j^o)}$$

Algorithm 2 구조학습 방법

```
//  $r_j$  :  $j$ 번째 규칙 //
//  $k$  : 생성된 규칙 갯수 //
//  $t_i$  :  $i$ 번째 학습 데이터 //
//  $N$  : 총 학습 데이터 //
//  $m_{ij}$  :  $t_i$ 와  $r_j$  사이의 정합 정도//
//  $m^d$  : 정합판단 값 //
//  $M(\cdot, \cdot)$  : 정합 함수 //
//  $R(t_i)$  : 학습데이터  $t_i$ 로부터 생성된 규칙//
```

```
1  $R(t_1) \triangleright$  첫번째 학습데이터로 부터 첫번째 규칙을 생성
2  $k \leftarrow 1$ 
3 for  $i=2$  to  $N \triangleright$  각 학습데이터를 테스트
4   for  $j=1$  to  $k \triangleright$  기 생성된 규칙들과 테스트
5      $m_{ij} = M(t_i, r_j)$ 
6   end for
7   If  $\max(m_{i1}, \dots, m_{ik}) < m^d$  then
8      $k \leftarrow k + 1$ 
9      $R(t_i) \triangleright k$ 번째로 생성된 규칙  $r_k$ 
10  end if
11 end for
```

$$= \frac{\sum_{j=1}^m \mu_{v_k}^{-1}(1) \frac{h_j^o}{\mu_{v_k}^{-1}(h_j^o)_{\max} - \mu_{v_k}^{-1}(h_j^o)_{\min} + 1}}{\sum_{j=1}^m \frac{h_j^o}{\mu_{v_k}^{-1}(h_j^o)_{\max} - \mu_{v_k}^{-1}(h_j^o)_{\min} + 1}} \quad (3)$$

식 (3)에서, $\mu_{v_k}^{-1}(1)$ 은 해당 소속함수의 중심을 의미한다. 즉 가우시안 함수 $e^{-((x-c)^2/\sigma^2)}$ 를 소속함수로 사용한다 $\mu_{v_k}^{-1}(1)$ 은 중심 c 를 의미한다.

퍼지 규칙을 형성하는 방법을 설명하면 다음과 같다. 예를 들어, 입력노드가 x_1, x_2 이고 중간 노드가 h_1, h_2 이고 출력노드가 y_1 인 간단한 시스템을 살펴보자. 퍼지언어항이 NB, ZO, PB 3개로 표현되고 중간노드 h_1 의 입력 링크 w_{11} 는 NB, w_{21} 는 ZO, 출력링크 v_{11} 는 NB로 중간노드 h_2 의 입력링크 w_{12} 는 ZO, w_{22} 는 PB, 출력링크 v_{21} 는 ZO로 주어졌다고 한다면 이것은 다음과 같은 퍼지규칙을 표현하는 것이 된다.

$$\begin{aligned} r_1 : & \text{if } x_1 \text{ is NB and } x_2 \text{ is ZO, then } y_1 \text{ is NB} \\ r_2 : & \text{if } x_1 \text{ is ZO and } x_2 \text{ is PB, then } y_1 \text{ is ZO} \end{aligned} \quad (4)$$

결국 중간노드의 수는 뉴로퍼지 시스템에 있어서 퍼지 규칙의 수가 된다. 퍼지 규칙의 수는 신경망에 있어서 중간노드의 수 만큼이나 중요한 요소이다. 신경망에 있어서 최적의 중간노드의 수를 결정하는 결정적인 알고리즘은 아직 개발된 것이 없을 정도로 이 문제는 난이한 문제이다. 중간노드의 수가 적으면 학습시간은 빠르나 입출력 매핑능력이 떨어질 수 있다. 반면 중간노드의 수가 많으면 매핑능력이 좋아지나 학습시간이 너무 오래 걸리는 단점이 있다. 본 논문에서는 퍼지규칙의 수를 자동 결정하기 위한 구조학습 알고리즘으로 알고리즘 2를 사용하였다. 알고리즘 2에서, 모든 입출력 퍼지 언어항의 소속함수는 해당 입출력 범위내에서 일정하게 분포하고 있다고 가정한다. 이러한 가정은 제안한 뉴로퍼지 시스템이 학습되기 전의 상태이다. 학습이 진행됨에 따라서 소속함수의 분포와 소속함수의 파라미터 (예로, 가우시안 함수를 소속함수로 사용했을때는 중심과 표준편차)는 학습데이터에 맞추어 변화되며, 최종적으로는 초기 분포와 다르게 된다. 첫번째 학습데이터를 적용시에는 기 만들어진 규칙이 없기 때문에 이 데이터를 이용하여 바로 다음과 같이 퍼지 규칙을 생성한다. 첫번째 학습데이터가 $t_1 = (x_{11}, x_{21}, y_{11})$ 라고 주어지고 3개의 퍼지언어항 NB, ZO, PB가 사용된다면 첫번째 규칙은 다음과 같이 결정된다.

$$\begin{aligned} w_{11} &= \arg \max \{ \mu_{NB}(x_{11}), \mu_{ZO}(x_{11}), \mu_{PB}(x_{11}) \} \\ w_{21} &= \arg \max \{ \mu_{NB}(x_{21}), \mu_{ZO}(x_{21}), \mu_{PB}(x_{21}) \} \\ v_{11} &= \arg \max \{ \mu_{NB}(y_{11}), \mu_{ZO}(y_{11}), \mu_{PB}(y_{11}) \} \end{aligned} \quad (5)$$

다음 학습데이터 부터는 이전에 만들어진 규칙과의 정합 정도를 따져서 새로운 규칙이 생성될수도 있고 생성되지 않을 수도 있다. 알고리즘 2에서, 정합 함수 $M(\cdot, \cdot)$ 는 학습데이터와 기존의 만들어진 규칙들과의 정합정도를 계산해낸다. 예를들어, 두번째 학습데이터가 $t_2 = (x_{12}, x_{22}, y_{12})$ 라고 주어졌다면 정합정도 m_{21} 는 다음 수식처럼 계산된다.

$$m_{21} = M(t_2, r_1) = \min \{ \mu_{w_{11}}(x_{12}), \mu_{w_{21}}(x_{22}), \mu_{v_{11}}(y_{12}) \} \quad (6)$$

만약 m_{21} 의 값이 정합판단 값 m^d 보다 작다면 두번째 학습데이터를 이용하여 새로운 규칙이 생성된다. 그렇지 않다면 두번째 규칙은 생성되지 않으며 세번째 학습데이터를 처리하게 된다. 만약 기 생성된 규칙의 수가 여러개 존재할 때는 모든 규칙에 대하여 정합 값을 계산하여 판단하여야만 한다. 즉 i 번째 학습데이터 처리시 k 개의 퍼지 규칙이 기 생성되었다면 새로운 규칙을 생성하는 조건은 $\max(m_{i1}, \dots, m_{ik}) < m^d$ 가 된다. 결국 m^d 값이 크면 클수록 많은 규칙이 생성된다. m^d 값이 너무 작을 경우에는 규칙의 수가 매우 작아 학습속도는 빠르게 되나 정합능력이 떨어지게 된다. 반면 m^d 값이 클 경우에는 규칙의 수가 많이 생성되어 학습속도가 느려지게 되나 정합능력은 향상된다. 그러나, 규칙의 수가 너무 많아지면 학습속도가 느려지는 것과 더불어 과도 정합 (over-fitting) 이 될수도 있다. 학습데이터 수가 충분하지 않을 경우 과도정합은 일부 테스트 데이터에 대하여 좋지 않은 결과를 생성할 수 있다. 뉴로퍼지 시스템에서 규칙의 갯수는 중간노드의 수로 신경망에서 중간노드의 수와 동일한 의미를 지닌다. 신경망에서 최적의 중간노드의 수를 찾는 것은 아직 해결되지 않은 문제로서 알고리즘이 존재하지 않는다. 신경망에서와 마찬가지로 본 논문의 뉴로퍼지 시스템에서 최적의 m^d 값을 찾아내는 알고리즘은 매우 어려운 문제이다. 본 논문에서는 실험적으로 선택한 m^d 값을 사용했으나, 차후 퍼지 언어항의 갯수 및 대상 시스템의 복잡도 (complexity)에 따라서 적절한 m^d 값을 찾는 알고리즘 개발이 필요하다 하겠다. 위 알고리즘을 사용하면 규칙의 갯수뿐만 아니라 규칙도 생성되어 진다. 이렇게 생성된 규칙을 초기규칙으로 사용하면 무작위적으로 생성된 초

기규칙을 사용할 때 보다 학습시간이 단축된다.

진화전략을 이용하여 NFSM을 학습시키는 과정은 다음과 같다. 그림 1에서 보듯이 ESTM 은 개체 집단을 생성후 이를 진화시킴으로서 NFSM 을 학습시킨다. 학습을 시키기 위하여 먼저 학습의 대상이 되는 파라미터를 결정해야한다. 본 논문에서 사용한 파라미터는 다음과 같다.

- 입출력링크의 퍼지 언어항 (다른 말로 퍼지 규칙)
- 퍼지 언어항의 중심과 표준편차

이러한 파라미터는 개체 표현에 포함되어 진화하게 된다. 그림 2는 개체군에서 각 개체의 표현형식을 보여준다. 그림 2에서 보면 하나의 개체는 퍼지언어항을 표시하는 부분과 퍼지 규칙을 표현하는 부분으로 나뉜다. 퍼지언어항을 표시하는 부분은 입출력 별로 퍼지 언어항 소속함수의 중심과 표준편차 (x, o)를 표현하며 퍼지규칙 부분은 입출력에 대응하는 퍼지규칙을 표현한다. 그림 2에서 첫번째 개체가 표현하는 퍼지규칙은 식 (4)의 퍼지규칙을 표현하고 있다. 결국 하나의 개체는 하나의 뉴로퍼지가 동작하는데 필요한 모든 파라미터를 표현한다. 이와같이 입출력별로 퍼지언어항의 소속함수를 정함으로서 보다 정교한 매핑이 가능해진다.

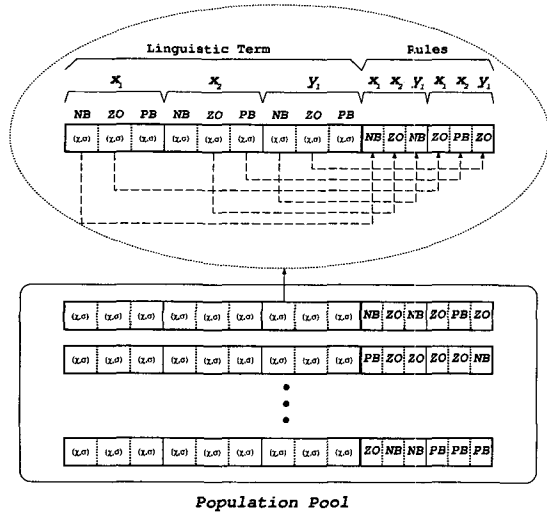


그림 2. 개체군에서 개체의 표현
Fig. 2. Individual representation in population pool.

초기에는 모든 개체 퍼지언어항의 중심은 각 입출력 범위에서 균등하게 생성된다. 즉 어떤 입력이 -1에서 1사이

1 우리는 퍼지 언어항을 표현하는데 가우시안 함수를 사용하였다.

의 값을 가지고 퍼지언어항 3개를 사용하여 표현된다면, NB, ZO, PB의 중심은 각각 -1, 0, 1을 갖게 된다. 표준편차는 사용자 파라미터로서 사용자로부터 입력을 받게 되며 초기에는 모든 개체의 모든 입출력에서 동일하게 주어진다. 학습하는 동안에 중심과 표준편차는 갱신된다. 마찬가지로 퍼지규칙도 갱신된다. 퍼지규칙은 퍼지언어항 중 하나로 표현됨으로 퍼지언어항의 갯수내에서 정수로 표현된다. 예를들어, 3개의 퍼지언어항으로 표현되는 시스템에서는 0서부터 2까지의 값을 갖게된다. 퍼지규칙은 초기에는 알고리즘 2에 의하여 자동설정되며 모든 개체에서 같은 값을 갖는다. 퍼지규칙도 퍼지언어항처럼 학습에 의하여 갱신된다.

Algorithm 3 개체 평가

```

// N : 총 학습데이터 수 //
// P : 총 개체 수 //
// O : NFSM의 출력 수 //
// NFSMi : i번째 개체를 이용하여 설정된 NFSM //
// ojk : j번째 학습데이터 적용시 NFSMi의 k번째 실제 출력 //
// tjk : j번째 학습데이터의 k번째 목표 출력 //
// ei : i번째 개체의 MSE (mean square error)//
// fi : i번째 개체의 적합도 //

1 for i=1 to P
2   NFSMi 설정
3   ei ← 0
4   for j=1 to N
5     j번째 학습데이터를 NFSMi에 적용
6     NFSMi의 실제출력 oj을 구함
7     에러 계산, ej = ∑k=1O (tjk - ojk)2
8   end for
9   i번째 개체의 MSE를 계산, ei = ∑j=1N ej/N
10  i번째 개체의 적합도를 설정, fi = 1 / (1 + ei)
11 end for
    
```

위에서 설명한 개체표현 방법에 따라 초기에 2N개의 개체가 생성된다.² 개체숫자의 선택은 사용자 파라미터로

실험전에 설정되어야 한다. 진화전략 알고리즘은 부모개체로부터 자식개체를 생성하고 자식개체를 평가하여 좋은 개체를 유지함으로 진화해 나간다. 알고리즘 3은 각 개체를 평가하는 방법이다. 각 개체의 적합도를 평가하기 위하여 학습데이터를 하나씩 NFSM 에 입력후 출력을 구해 원하는 출력사이의 자승에러를 구한다. 모든 학습데이터에서의 자승에러의 합을 구하고 이를 학습데이터 수로 나눈 평균자승에러 (Mean Square error: MSE) e_i 가 구해지면 이를 이용하여 해당 개체의 적합도를 계산한다. 적합도는 MSE 가 작을것이 좋은 것임으로 MSE 에 역비례하게 주어진다. 식 (7)은 본 논문에서 사용한 i 번째 개체의 적합도를 구하는 수식이다. 본 논문에서는 다음 세대를 선택하기 위해 적합도가 큰 순위로 정렬하고 그 중의 N 개를 선택해 다음세대의 부모 세대로 사용하기 때문에 적합도 순위만 중요할 뿐 적합도 크기는 중요하지 않다³

$$f_i = \frac{1}{1 + e_i} = \frac{1}{1 + \frac{1}{N} \sum_{j=1}^N \sum_{k=1}^a (t_j^k - o_j^k)^2} \quad (7)$$

식 (7)을 보면 최대적합도는 MSE 가 0 일경우인 1 이된다. 알고리즘 1에서 보듯이 각 개체의 평가가 끝나면 자식개체를 생성한다. 자식개체를 생성하는 것은 부모개체의 파라미터를 변경하여 더 좋은 개체를 탐색하기 위한 작업이다. 보통 진화전략에서는 이 과정을 돌연변이라고 하는데 부모개체의 파라미터에 가우시안 잡음을 더해주는 동작이다. 본 논문의 뉴로퍼지 시스템에서 부모 개체의 파라미터로는 이 절 앞에서 언급한 것처럼 퍼지언어항 부분과 퍼지규칙 부분이 있다. 퍼지 언어항 부분은 각 퍼지언어항 소속함수의 파라미터 (실수의 중심과 표준편차)로 이루어져 있으며 퍼지규칙 부분은 퍼지 언어항을 지정하는 정수로 이루어져 있음을 앞에서 설명하였다. 그러므로, 자식 개체 생성시에는 퍼지언어항과 퍼지규칙 부분을 구분하여 생성해야만 한다.

먼저 퍼지언어항 부분의 돌연변이를 살펴보면 다음과 같다. 자식개체는 부모개체의 퍼지언어항 소속함수의 중심과 표준편차에 평균이 0 이고 특정 표준편차를 갖는 가우시안 잡음을 더해준다. 본 논문에서는 가우시안 잡음의 표준편차 값을 일정한 값을 사용하지 않았다. 돌연변이시

표준편차를 일정한 값으로 할 경우에는 여러가지 문제를 야기할 수 있다. 첫째로 표준편차 값이 크면 해에 접근하는 속도는 빨라지나 해에 근접한 상태에서는 발진 (oscillation) 할 수 있다. 또한 표준편차 값이 작으면 해에 접근하는 속도가 느려질 수 있다. 물론 가우시안 잡음은 표준편차 값이 크더라도 적은 값을 생성할 수 있지만 확률적으로 표준편차 값의 크기에 의존하여 잡음을 생성함으로써 이러한 경향이 나타난다. 또한 다수의 개체가 해에 근접해 있는 정도는 다르기 때문에 (즉 적합도가 큰 것과 작은 것이 있기 때문에), 가능한 해에 근접해 있는 개체는 (즉 적합도가 큰 개체) 미세한 탐색이 필요하며 해에서 멀리 떨어져 있는 개체는 (즉 적합도가 작은 개체) 탐색 범위가 커야할 필요가 있다. 또한 표준편차의 값은 입출력 변수의 동작 범위에 비례해야한다. 즉 어떤 입력이나 출력이 다른 입력이나 출력에 비하여 동작범위가 10배정도 크다면 표준편차의 크기도 이정도대로 커야한다는 것이다. 이러한 생각을 바탕으로 본 논문에서는 다음과 같이 각 개체의 적합도와 동작 범위에 따라서 표준편차 값을 계산하는 방법을 사용하였으며 실제 실험에서도 이 방법이 일정한 표준편차 값을 줄때 보다 더 좋은 성능을 보임을 확인하였다. 본 논문에서는 일정한 표준편차를 고정 표준편차로 부르며 표준편차를 변동하는 것을 변동 표준편차라 부른다.⁴

돌연변이를 위한 i 번째 개체의 변동 표준편차는 다음의 식으로 계산된다.

$$\sigma_i = \left(1 - \frac{f_i}{f^{\max}} + \sigma_i^{\min} \right) \times \frac{(\max - \min)}{N_h} \quad (8)$$

식 (8)에서 σ_i 은 돌연변이를 위한 표준편차를, σ_i^{\min} 은 최소 표준편차를, f^{\max} 는 개체군에서 최대 적합도를, f_i 는 i 번째 개체의 적합도를 의미하며, max와 min은 각 입출력의 최대, 최소 동작범위를 그리고 N_h 는 퍼지언어항의 갯수를 의미한다. 즉, 자식개체는 현재 부모개체의 적합도가 낮으면 낮을 수록, 입출력 동작범위가 크면 클수록 그리고 퍼지 언어항의 수가 작으면 작을수록 부모개체에 더해지는 가우시안 잡음이 커질수 있다. 이를 의미적으로 해석해 보면, 부모 개체의 적합도가 낮다는 것은 해와 많이 떨어져 있을 가능성이 높음으로 부모 개체와는 많은 변화가 필요하다는 것을 의미한다. 또한 입출력 동작 범

2 N 개는 부모개체, N 개는 자식개체이다.
3 유전자 알고리즘에서 많이 사용하는 Roulette wheel 선택방법은 적합도 크기가 중요하다.

4 본 논문의 핵심은 변동 표준편차 방법이 아님으로 고정 표준편차와 변동 표준편차에서의 성능 비교는 하지 않는다.

위가 크면 그렇지 않은 것보다 탐색공간이 커지기 때문에 큰 돌연변이를 필요로 한다. 퍼지 언어항의 수가 많을 때는 보다 정교한 조정이 필요함으로 작은 돌연변이가 유리하다. 식 (8)은 입출력 퍼지언어항 모두에 적용된다.

퍼지규칙 부분의 돌연변이는 입출력과 상관없음으로 동작범위를 고려할 필요없으며 마찬가지로 퍼지 언어항의 수 또한 고려할 필요가 없다. 본 논문에서는 퍼지 규칙 부분의 돌연변이를 위하여 식 (9)을 사용한다.

$$\sigma_r = \left(1 - \frac{f_i}{f_{\max}} + \sigma_r^{\min}\right) \quad (9)$$

퍼지 규칙은 퍼지 언어항의 수 내에서 정수로 표현되는 값이다.

수식 (8)과 (9)에 의하여 하나의 자식개체를 생성하는데 필요한 가우시안 잡음의 표준편차가 구해지며 그 다음 단계는 구해진 표준편차를 이용하여 실제로 자식개체를 생성하는 일이다. 자식개체의 퍼지 언어항 파라미터는 다음의 방법으로 생성된다.

$$x_o = x_p + N(0, \sigma_i) \quad (10)$$

$$\sigma_o = \sigma_p + N(0, \sigma_i) \quad (11)$$

수식에서 아래에 기입한 o 는 자식개체를 p 는 부모개체를 의미한다.

$$r_o = r_p + N(0, \sigma_r) \quad (12)$$

그러나, 퍼지 규칙은 0부터 퍼지언어항 수-1 까지의 정수만을 가져야 함으로 추가적인 조작이 필요하다. 추가적인 조작은 다음과 같다.

$$r_o^* = \begin{cases} 0 & \text{if } r_o < 0 \\ N_u - 1 & \text{if } r_o > N_u - 1 \\ (\text{integer})r_o & \text{Otherwise} \end{cases} \quad (13)$$

식 (8)과 (9)에서 보듯이, 최대 적합도를 갖는 개체에서는 $\sigma_i = \sigma_i^{\min} \times \frac{(\max - \min)}{N_u}$ 이며, $\sigma_r = \sigma_r^{\min}$ 가 된다. 그러므로, 최대 적합도를 갖는 부모 개체에서 생성된 자식 개체도 진화하게 된다. 본 논문에서 최대 적합도를 갖는 부모개체도 자신을 유지하지 않고 자식개체로 진화하게 한 것은 부모와 자식 개체를 합친 개체군에서 다음 세대 부모를 선택하기 때문이다. 그러므로, 부모 세대의 최대 적합도보다 큰 적합도를 갖는 자식 개체가 생성되지 않아도 부모 세대에서 최대 적합도를 갖는 개체는 다음 세대에 살아남게 된다. 만약 생성된 자식 세대를 다음세대 부모

세대로 선택하는 방법을 사용할 경우에는 최대 적합도를 갖는 부모 개체의 σ_i^{\min} 와 σ_r^{\min} 을 0으로 하여 이 개체가 자식세대에서도 보존되게 하는 것이 좋다.

N 개의 부모 개체로부터 N 개의 자식개체가 생성된 후 부모개체와 자식개체는 적합도가 큰 순으로 재정렬되며 이중 적합도가 큰 N 개의 개체가 다음 세대의 부모 개체가 된다. 그러므로 매 세대에서 가장 적합도가 큰 개체는 개체군의 제일 앞에오며 이 개체의 적합도가 사용자가 설정한 최대 적합도보다 커지면 학습이 종료하게 된다. 식 (7)에서 보듯이 적합도는 MSE로 부터 계산됨으로 종료 조건은 MSE 값으로 줄 수 있다. 예를 들어, 종료조건 MSE 값이 ϵ 이라면, 종료적합도는 $f_i = \frac{1}{1+\epsilon}$ 가 된다. 학습이 종료하면 최대 적합도를 갖는 개체의 파라미터를 이용하여 NFSM 의 파라미터값을 설정한다.

III. 실험 결과

본 논문에서 제안한 뉴로퍼지 시스템을 두개의 대표적인 비선형 시스템을 이용하여 비선형 시스템 동정에 응용하였다. 사용한 비선형 시스템은 여러논문에서^[5,8] 사용된 것으로서 식 (14) (15)과 같다.

$$\text{system 1 : } y(k+1) = \frac{y(k)}{1+y^2(k)} + u^3(k) \quad (14)$$

$$\text{system 2 : } y(k+1) = \frac{y(k)y(k-1)y(k-2)u(k-1)(y(k-2)-1)u(k)}{1+y^2(k-1)+y^2(k-2)} \quad (15)$$

첫번째 시스템의 뉴로퍼지 입력은 $u(k)$ 와 $y(k)$ 이고 출력은 $y(k+1)$ 이며 두번째 시스템의 입력은 $y(k)$, $y(k-1)$, $y(k-2)$, $u(k)$, $u(k-1)$ 이고 출력은 $y(k+1)$ 이다. 학습데이터는 대상 시스템의 출력 $y(k+1)$ 을 제외하고 모든 입력값을 $[-1,1]$ 범위에서 일정하게 분포한 무작위 값으로 주고 시스템의 출력을 기록하여 작성하였다.

표 1은 두 시스템에 적용된 뉴로퍼지 파라미터 값을 보여준다. 학습데이터 수는 해당 system의 입출력 동작을 충분히 학습할 수 있도록 적절히 설정되어야 한다. 학습 데이터 수가 너무 작으면 시스템 동정시 특정 입력 영역에 대한 출력이 정의되지 않아 시스템 동정에 문제가 발생할 수 있다. 또한 불필요하게 너무 많으면 거의 유사한 입출력 동작을 학습하는 경우가 많아 지게 되어 학습시간이 많이 걸리게 된다. 본 논문에서는 여러번의 실험결과 200개 정도의 학습데이터가 적당함을 발견하여 이것으로

표 1. 뉴로퍼지 파라미터
Table 1. Parameters of neuro-fuzzy system.

파라미터	system 1	system 2
입/출력 수	2/1	5/1
학습데이터 수	200	
퍼지언어항	5	3
초기소속 함수의 겹침 정도	0.8	
정합 판단값 m^d	0.4	
학습종료 MSE	0.001	0.005
부도개체수, 자식개체수	40, 40	
σ_r^{\min}	0.00005	
σ_f^{\min}	0.005	

설정하였다. 차후 대상 시스템의 동작범위 및 동역학(system dynamics)을 고려하여 최적의 학습데이터 수를 설정하는 알고리즘 개발이 필요하다 하겠다. 퍼지 언어항의 개수는 system 1에서는 5개를, system 2에서는 3개를 사용하였다. 알고리즘 2에서 보듯이, 입력의 개수가 많아지면 생성된 규칙과의 정합될 가능성은 더욱 줄어든다. 결국 이는 필요한 규칙의 수를 증가시키게 된다. 여러 번의 실험 결과 system 2에서 5개의 퍼지 언어항을 사용했을 때 필요한 퍼지 규칙의 수가 크게 증가하고 학습시간이 매우 길어지는 것을 관찰하였다. 또한 3개의 퍼지 언어항을 사용했을 때 성능이 5개의 퍼지 언어항을 사용했을 때와 거의 유사함에도 불구하고 작은 수의 퍼지 규칙이 생성되어 빠르게 학습됨을 관찰하였다. 이러한 관찰을 기반으로 system 2에서는 3개의 퍼지언어항을 사용하였다. 절에서 언급한 것과 같이 m^d 값과 퍼지언어항의 개수 그리고 학습데이터 수, 퍼지 규칙의 수는 밀접한 연관관계를 갖고 있다. 학습데이터 수가 결정되면 이것을 기반으로 최적의 퍼지규칙의 수를 구하기 위한 최적의 m^d 값과 최적의 퍼지언어항의 수를 설정하는 결정적인 알고리즘에 대한 연구가 앞으로 진행되어야 할 것으로 판단된다.

초기 퍼지규칙 및 퍼지규칙의 수 (다른 말로 중간노드의 수)는 알고리즘 2에 의하여 자동설정된다. 알고리즘 2에서는 기존의 만들어진 규칙과 새로만들어질 규칙의 정합도를 따지게 됨으로, 퍼지언어항의 초기 소속함수의 겹침정도와 정합판단값 m^d 값이 필요한 규칙의 수를 결정하는데 중요한 파라미터가 된다. 겹침정도라는 것은 이웃하는 퍼지언어항의 소속함수가 만나는 곳의 값을 말한

다. 예로, 초기 소속함수의 겹침정도가 0.8로 설정되고 3개의 퍼지언어항이 사용된다고 가정하고 입력 x_1 의 최소, 최대 범위가 -1에서 1이라면 입력 x_1 의 학습전의 소속함수의 분포는 그림 3와 같다. 초기 소속함수의 겹침정도 값이 크면 클수록 자동결정되는 퍼지규칙의 수는 감소하며 정합판단값이 커지면 커질수록 퍼지규칙의 수는 증가한다. 퍼지 규칙의 수는 너무 크거나 너무작은 경우가 아니면 성능에 큰 영향을 미치지 않는다. 단 퍼지 규칙의 수는 중간노드의 수를 결정하기 때문에 크고 적음에 따라서 학습시간에는 많은 영향을 끼쳤다. 퍼지 규칙의 수가 너무 작게 설정이 되면 자유도가 낮아져서 동정이 세부적으로 되지 않으며, 퍼지 규칙의 수가 너무 많으면 동정은 잘 되었으나 학습시간이 많이 걸린다. 표 1에서 정한 값은 성능에 큰 문제가 없으면서 가능한 학습시간이 적게 걸리는 파라미터 값을 실험적으로 설정한 값이다.

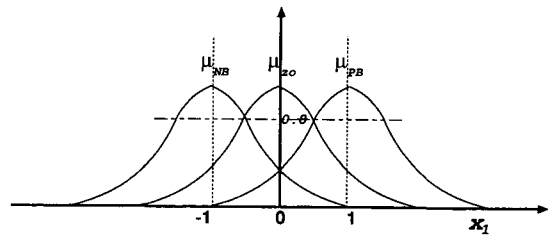


그림 3. 초기 소속함수의 분포 예
Fig. 3. Example of initial membership functions.

학습 초기에 각각의 퍼지 언어항에 대한 소속함수는 입출력별도 동작범위 안에서 일정하게 분포하게끔 생성된다. 이때 각 소속함수간의 겹치는 값에 따라서 소속함수의 폭이 결정된다. 여러 번의 실험결과 이 값이 0.5보다 작으면 학습이 빠르게 진행되나, 과도정합하는 경우가 발생할 가능성이 높아지게 됨을 발견하였다. 또한 학습데이터 중 특정한 학습데이터에 지나치게 의존하여 전체적인 성능은 떨어짐을 발견하였다. 즉, 이 값이 작으면 매칭되는 규칙의 수가 작아지고 출력 결과가 소수의 규칙에 의하여 결정되어 해당 규칙이 좋은지 아닌지를 판단하는데 부정확한 결과를 초래한다. 이것에 대한 질문이 올 경우 MSE가 똑같아도 특정데이터에서는 거의 0이고 다른 데이터에서는 MSE가 많아져서 전체적인 MSE는 동일함에도 특정 데이터 이외의 데이터에 대한 응답이 좋지 않음을 설명한다. 또한 너무 클 경우 특정 데이터 집합에 의존하는 경향이 사라지나 학습시간이 많이 걸리는 문제가 발생한다.

정합 판단값 m^d 는 실험을 통하여 0.4로 선택하였다. 성능에 큰 문제가 없으면서도 가능한한 적은 갯수의 퍼지 규칙의 수를 생성하게 하는 값을 선택하였다. 학습 종료 를 판단하는 MSE 값은 system 1에서는 0.001, system 2 에서는 0.005로 주었다. system 1인 경우 입력의 갯수가 작고 시스템 동역학이 비교적 간단하여 MSE 가 0.001에서 학습 포화상태⁵ 도달했으나, system 2인 경우 입력 갯 수가 많고 시스템 동역학이 복잡하여 학습 포화상태가 0.005 근방에서 일어났다. 이것을 근거로 하여 표에서와 같이 설정하였다.

부모 개체수와 자식개체수는 너무 작지 않는한 성능에 크게 영향을 미치지 않는았다. 그 이유는 한 세대내의 개체수가 작으면 한 세대에서의 계산량은 줄어드는 대신 여러세대에 걸쳐서 진화가 일어나기 때문에 진화정도대 시간으로 따지면 개체수가 많고 적고를 떠나 비슷한 시간에 비슷한 진화가 일어나기 때문이다. σ_i^{\min} 그리고 σ_r^{\min} 은 성능보다 상대적으로 학습율 및 학습 시간에 영향을 많이 주었다. σ_i^{\min} 과 σ_r^{\min} 이 클경우 자식 개체 생성시 부모 개체의 파라미터값에서 많이 변화를 함으로 초기에는 빠르게 진화하나, 진화가 많이 진행되어 해 근처에 다달았을 때는 오히려 많은 변화로 인해 좋은 자식 개체를 생성하는데 어려움이 있었다. 이 값이 너무 작을 경우에는 너무 세부적인 탐색을 하게 됨으로 꾸준한 진화를 보이나 해 근처에 다달았는데 많은 세대수가 필요했다. 대부분의 개체들이 해 근처에 도달했을 때에는 각 개체의 적합도가 해당 세대의 최대 적합도와 차이가 적기때문에 이 상황에서는 σ_i^{\min} 과 σ_r^{\min} 의 값에 의존하여 진화하게 된다.

학습데이터에서 입력은 각 입력별 동작범위 내에서 무작위적으로 선택된 값으로 주었으며, 이 값을 이용하여 각 시스템에 적용하여 출력을 구하고 해당 출력을 기록하여 만들었다. 그러므로, 각 학습데이터별로 어떠한 연관관계도 존재하지 않는다. 즉, 출력은 입력에만 의존하며 이전의 입력값에는 무관하게 구해진다. 구해진 학습데이터를 이용하여 ESTM 이 NFSM 을 학습시킨다. 학습이 끝나면 시험용 입력을 각 시스템과 학습된 뉴로퍼지 시스템에 가해서 출력이 얼마나 동정이 되었는지를 평가한다. 두 시스템의 시험용 입력은 각각 수식 (16)과 (17)에 주어졌다.

$$\text{system 1 : } u(k) = \sin(2\pi k/25), 0 \leq k < 100 \quad (16)$$

⁵ 학습이 진행되어도 MSE 가 더이상 떨어지지 않는 상태를 말한다.

system 2 :

$$u(k) = \begin{cases} 0.7 \sin(\pi k/25) & , 0 \leq k < 250 \\ 0.6 & , 250 \leq k < 500 \\ -0.4 & , 500 \leq k < 750 \\ 0.3 \sin(\pi k/25) + 0.1 \sin(\pi k/32) + 0.6 \sin(\pi k/10) & , 750 \leq k < 1000 \end{cases} \quad (17)$$

system 1 과 system 2에서 초기의 상태 저장값들, 예로 system 2에서 $y(k), y(k-1), y(k-2)$ 과 지연된 입력 $u(k-1)$ 는 0으로 간주된다. 시험용 입력이 가해 지면서 이러한 값들은 시험용 입력과 출력에 따라서 적절히 설정 되어 출력이 구해진다. system 1 및 system 2 에서의 실험결과가 표 2에 있으며 동정된 결과가 각각 그림 4 와 5 에 있다. 알고리즘 2 에 의하여 결정된 퍼지 규칙의 수는 system 1 과 system 2에서 동일하게 11개로 구해졌다. 학습 종료 조건을 만족하기 위한 학습 세대수는 system 2에서 크게 나타났다. 이는 시스템 2의 높은 복잡도에 기인한 것으로 판단된다. 또한 이러한 복잡도는 system 2에서 많은 실행시간이 필요하게 한 요인이 된 것으로 판단된다. 그림 4 와 5에서 보듯이, 뉴로퍼지 시스템은 두 시스템 동작에 상당히 동정이 되어있는 것을 볼 수 있다.

표 2. 실험 결과

Table 2. Experimental results.

측정 파라미터	system 1	system 2
구해진 규칙 수	11개	11개
총 학습 세대 수	2241	64110
학습 종료시 MSE	0.000996	0.004999
실행 시간	1510초 (25분 10초)	87666초 (24시간 21분 6초)

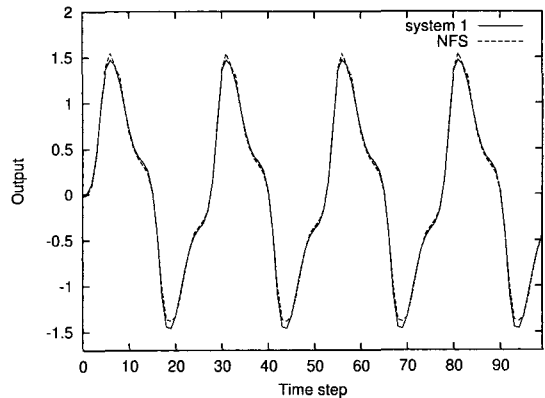


그림 4. 시스템 1에서의 실험 결과

Fig. 4. Experimental results of system 1.

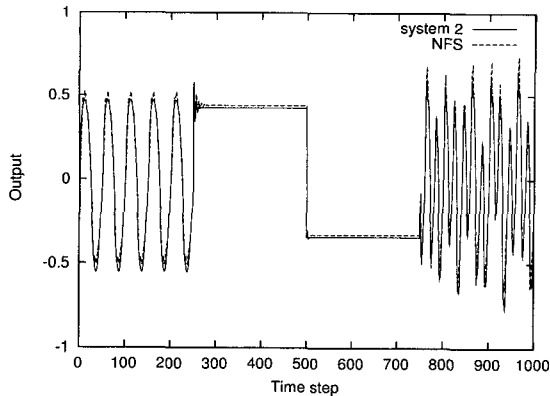


그림 5. 시스템 2에서의 실험 결과

Fig. 5. Experimental results of system 2.

IV. 결 론

본 논문에서 우리는 새로운 구조의 뉴로퍼지 시스템을 제안하고 제안한 뉴로퍼지 시스템을 두 개의 비선형 시스템 동정에 적용하여 그 유용성을 살펴보았다. 제안한 뉴로퍼지 시스템의 학습 알고리즘으로 본 논문에서는 표현형 형태의 실수로 개체를 표현할 수 있는 진화전략을 이용하여 학습하였다. 또한 학습데이터로부터 필요한 퍼지 규칙의 수를 자동 생성하는 알고리즘을 제안하여 반복적인 실험에 의하여 필요한 퍼지 규칙의 수를 찾는 것보다 유용함을 보였으며 이 알고리즘에 의하여 생성된 규칙을 초기 규칙으로 사용하여 보다 빠른 해에 접근이 가능함을 확인 하였다. 두개의 비선형성이 강한 시스템에 제안한 시스템을 적용해 본 결과 비교적 빠르고 성능이 우수한 동정을 수행함을 보였다. 본 연구 수행중 동정에 필요한 적절한 학습데이터 수를 찾는 것과 동작영역 전체를 포함하게하는 학습데이터를 추출하는 방법에 대한 연구가 더 필요함을 알게 되었다. 또한 대상 시스템의 동역학 특성에 따라 적절한 퍼지 언어항의 수 및 적절한 규칙의 수를 찾아내는 알고리즘이 필요함을 느꼈다. 본 논문에서 필요한 퍼지 규칙의 수를 찾는 데 사용한 알고리즘의 파라미터인 정합판단값 m^d 와 초기 소속함수의 겹침정도 그리고 퍼지 언어항의 수가 어떻게 퍼지 규칙의 수에 영향을 주는가를 면밀히 관찰하여 체계적인 m^d 값을 찾아내는 방법 또한 새로 연구되어야 할 필요가 있다.

참 고 문 헌

[1] D. Nauck, "Beyond neuro-fuzzy: Perspectives

and directions," in *Proc. of the Third European Congress on Intelligent Techniques and Soft Computing (EUFIT '95)*, pp. 1159-1163, 1995.

[2] c.-J. IIN, "SISO Nonlinear System Identification Using A Fuzzy-neural Hybrid System," *International Journal of Neural Systems*, Vol. 8, pp.325-337, June 1997.

[3] W. A. Farag, V. H. Quintana, and G. Lambert-Torres, "A Genetic-Based Neuro-Fuzzy Approach for Modeling and Control of Dynamical Systems," *IEEE Trans. on Neural Networks*, Vol. 9, pp. 756-767, Sept. 1998.

[4] S. K. Pal and S. Mitra, eds., *Neuro-Fuzzy Pattern Recognition-Methods in Soft Computing*. John Wiley & Sons, 1999.

[5] K. S. Narendra and K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Trans. on Neural Networks*, Vol. 1, pp. 4-27, Mar. 1990.

[6] J. Kim and N. Kasabov, "HyFIS: adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems," *Neural Networks*, Vol. 12, pp. 1301-1319, 1999.

[7] Fabrizio, "Evolutionary Neural Fuzzy Systems for Noise Cancellation in Image Data," *IEEE Trans. on Instrumentation and Measurement*, Vol. 48, pp. 915-920, Oct. 1999.

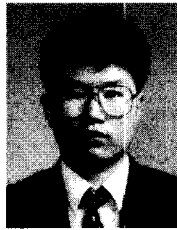
[8] C.-H. Lee and C.-C. Teng, "Identification and Control of Dynamic Systems Using Recurrent Fuzzy Neural Networks," *IEEE Transactions on Fuzzy Systems*, Vol. 8, pp. 349-366, Aug. 2000.

[9] D. Nauck and R. Kruse, "New learning strategies for nefclass," in *Proc. Seventh International Fuzzy Systems Association World Congress IFSA '97*, pp. 50-55, 1997.

[10] D. Nauck and R. Kruse, "A neuro-fuzzy approach to obtain interpretable fuzzy systems for function approximation," in *Proc. IEEE International Conference on Fuzzy Systems (FUZZ IEEE '98)*, pp. 1106-1111, 1998.

- [11] T. L. Seng, M. B. Khalid, and R. Yusof, "Tuning of a Neuro-Fuzzy Controller by Genetic Algorithm," *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 29, pp. 226-236, Apr. 1999.
- [12] D. B. Fogel, "An Introduction to Simulated Evolutionary Optimization," *IEEE Trans. on Neural Networks*, Vol. 5, pp. 3-14, Jan. 1994.
- [13] S. H. Jung, K. H. Cho, T. G. Kim, and K. H. Park, "Defuzzification Method for Multishaped Output Fuzzy Sets," *Electronics Letters*, Vol. 30, pp. 740-742, Apr. 1994.

 저 자 소 개



鄭 成 薰(正會員)

1966년생. 1988년 2월 한양대학교 전자공학과 학사. 1991년 2월 한국과학기술원 전기및전자공학과 공학석사. 1995년 2월 한국과학기술원 전기및전자공학과 공학박사. 1995년 3월~1996년 2월 한국과학기술원 전기및전자공학과 위촉연구원. 1996년 3월~1998년 2월 한성대학교 정보전산학부 전임강사. 1998년 3월~현재 한성대학교 정보전산학부 조교수. 주 관심분야는 지능시스템, 퍼지 및 신경망, 진화연산, 지능제어, 패턴인식, 시뮬레이션 등임