퍼지 및 지능시스템학회 논문지 2003, Vol. 13, No. 1, pp. 6-11

모듈라 신경망의 최적구조 설계

Optimal Structure Design of Modular Neural Network

김성주 · 전홍태
Seong-Joo Kim and Hong-Tae Jeon

중앙대학교 일반대학원 전자전기공학부

요 약

신경회로망은 최근 복잡하고 정형화가 어려운 문제의 해결을 위해 지능적인 기법이 제안되고 있다. 모듈라 네트워크는 복잡한 문제를 부문제로 나누어 해결한다.
퍼지 시스템은 복잡한 문제를 작은 부문제로 나누어 해결한다는 점에서 모듈라 네트워크와 유사하다. 하지만 퍼지 시스템의 경우에도 입력공간을 전문가가 직접 분할하고, 룰을 만들어야 한다는 단점이 있다. 이러한 단점을 해결하기 위해 퍼지 뉴럴 네트워크가 제안되었다. 하지만 입력의 증가에 따른 규칙수가 지수적으로 증가하는 문제점을 안고 있다.
본 논문에서는 이런 문제점을 계층적 구조를 설계함으로서 해결하였다. 또한, 제안된 구조는 앞선 모듈의 출력이 후건부에만 기여하는 구조를 사용하였다. 제안된 구조를 사용함으로써 입출력관계가 더욱 이해하기 쉬워졌으며, 설계의 어려움을 해결했다. 또한 룰이 수가 작다는 장점이 있다.

Abstract

Recently, the modular network was proposed in a way to keep the size of the neural network small. The modular network solves the problem by splitting it into sub-problems. In this aspect, fuzzy systems act in a similar way. However, in a fuzzy system, there must be an expert rule which separates the input space. To overcome this, fuzzy-neural network has been used. However, the number of fuzzy rules grows exponentially as the number of input variables grow.
In this paper, we would like to solve the size problem of neural networks using modular network with the hierarchic structure. In the hierarchic structure, the output of precedent module affects only the THEN part of the rule. Finally, the rules become shorter being compared to the rule of fuzzy-neural system. Also, the relations between input and output could be understood more easily in the proposed modular network and that makes design easier.

Key words : Scaling Function, Wavelet Function, Neural Network, Wavelet Neural Network, Genetic Algorithm

1. 서 론

일반적으로 지능 제어에는 불확실성의 취급 및 학습기능을 가진 신경망이 많이 사용된다. 신경회로망을 사용할 경우 지역적 최소치에 수렴할 가능성이 있으며 다차원 확장의 경우에 전체 회로망의 구조가 복잡해짐에 따라 학습 과정에서 많은 계산이 이루어져야 하기 때문에 수렴 속도가 느린 단점을 지니고 있다[1].

이에 각각의 지역 · 전역적인 매핑방법의 장점을 수용하고 하나의 복잡한 문제를 단순한 여러 개의 부분적인 문제로 나누어 해결하는 방식을 수용한 모듈화 설계방법이 관심의 대상이 되고 있다.

모듈화된 시스템은 전체를 부분적으로 나누어 해결하기 때문에 학습 속도가 향상되고, 사용되는 파라미터가 감소하며 표현 능력을 증가시킬 뿐만 아니라, 구조의 단순화로 하드웨어 구현이 용이하다. 또한, 모듈화는 전체 시스템에서 필요한 계산을 지역적으로 행함으로써 전체 시스템의 동작을 통해서 부시스템의 역할을 정의할 수 있다.

한편, 퍼지 로직 시스템에서 퍼지 규칙의 개수가 입력 변수의 개수에 따라서 지수 함수적으로 증가하는 문제를 해결하기 위하여 계층 퍼지 시스템이 제안되었다[2][3]. 계층 퍼지 시스템은 퍼지 규칙의 개수를 줄이는 데는 성공하였지만 퍼지 규칙의 전건부에서 이전 계층의 출력을 새로운 입력 변수로서 이용하기 때문에 퍼지 규칙의 의미를 이해하기 어려운 단점을 가진다[2][3]. 이를 해결하기 위해 나온 것이 이전 계층의 퍼지 로직 시스템의 출력을 입력변수로 사용하지 않고 후건부에서 필요한 만큼 수용하는 형태로만 사용하는 구조이다. 이렇게 함으로써 모든 계층의 퍼지 로직 시스템이 퍼지 규칙을 구성할 때 시스템 변수만을 이용하여 전건부를 구성하게 되므로 보다 이해하기 쉬운 계층 퍼지 시스템을 구성할 수 있다.

본 논문에서는 모듈라 신경망의 전문가모듈을 계층적 퍼지신경망을 사용하여 구성하는 최적화된 모듈라 신경망을 설계하고자 한다. 제안한 계층적 구조로 구성된 모듈라 신경망(MNN) 구조의 경우, 입력공간을 적당하게 분할하여 전체 문제를 작은 부문제로 분할하는 구조로 간략히 할 수 있으

접수일자 : 2002년 1월 10일
완료일자 : 2003년 2월 22일
이 논문은 2002년도 중앙대학교 학술연구비 지원에 의한 것임.

며, 이를 계층적 구조로 연결함으로써 퍼지 규칙의 지수 함수적 증가를 억제하는 효과가 있다. 또한, 이전 계층의 출력을 전건부에 사용하지 않는다는 제약을 가함으로써 기존의 계층적 구조보다 더욱 강력하게 규칙의 수를 줄이며 설계시 인간의 사고 방식에 기초하여 더욱 쉽게 설계할 수 있다는 장점도 가지고 있다.

2. 계층적 모델

2.1 계층적 모듈라 신경회로망

Jacobs와 Jordan이 제안한 모듈라 네트워크는 작업 재구성과 학습, 재결합의 관점에서 탁월할 성능을 발휘하고 있다[4]. MNN은 모듈을 구성하는 엑스퍼트 망(Expert Network; EN)과 게이팅 망(Gating Network; GN)으로 구성된다. EN은 모듈화식 설계방식을 도입하여 복잡한 문제를 단순한 부문제로 나눌 경우 부문제를 학습하게 된다. 또한 GN은 복잡한 문제를 단순한 부문제로 나누는 작업 재구성과 부문제를 학습한 EN의 출력을 조합하여 최종 출력을 생성하는 재결합을 담당한다.

그림 1은 단일 신경망을 사용하여 EN와 GN을 구성한 예를 보인 것이다.

그림 1에서 $\boldsymbol{x} = [x_1\ x_2 \cdots x_P]^T$는 P차원의 입력벡터, $\boldsymbol{y}=[y^1\ y^2 \cdots y^Q]^T$는 최종출력 벡터, $\boldsymbol{y}_k=[y_k^1\ y_k^2 \cdots y_k^Q]^T$ 는 k번째 EN의 $Q\times 1$ 출력벡터, $\boldsymbol{d}=[d^1\ d^2 \cdots d^Q]^T$는 원하는 응답벡터이다. MNN의 최종 출력은 다음 식과 같다.

$$\boldsymbol{y} = \sum_{k=1}^{K} g_k \boldsymbol{y}_k \tag{1}$$

여기서, K는 모듈의 개수이고, g_k는 k번째 GN의 출력이며, $\boldsymbol{y}_k$는 k번째 EN의 출력 벡터이다.

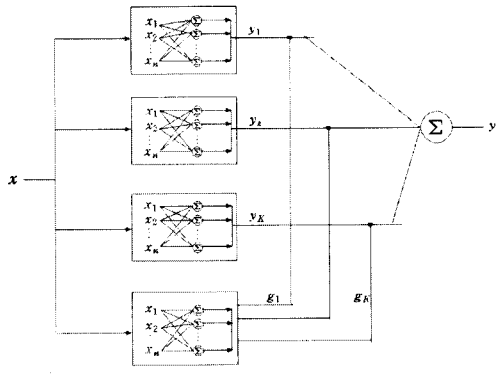


그림 1. 모듈라 신경망 구조
Fig. 1. Structure of modular neural network

그림 1의 일반적인 모듈라 신경망의 구조는 계층적 퍼지 신경회로망(HFNN)과 유사한 구조라고 할 수 있다. 모듈라 신경망 중에서 그림 2와 같은 계층적 구조를 지닌 신경망의 경우에 HFNN을 이용하면 전체 망 크기를 효율적으로 감소하고 학습에 있어서도 효과적인 최적화된 구조를 얻을 수 있다.

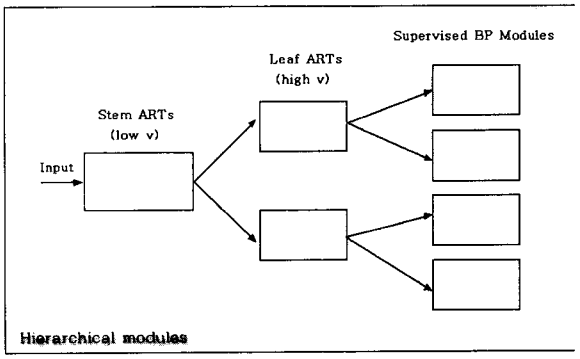


그림 2. 계층적 모듈라 신경망 구조
Fig. 2. Structure of hierarchical MNN

2.2 계층적 퍼지-신경회로망(Hierarchical Fuzzy Neural Network; HFNN)

룰의 수를 줄이거나, 구조의 간략화를 위해 계층적인 구조가 많이 연구되고, 많은 계층구조가 존재한다[2].

입력의 개수를 n이라 하고 PB, PM, ..., NB등과 같은 퍼지 변수 A_{ij}는 m개의 레이블로 기술되었다고 할 때, 퍼지룰 베이스는

$$R_i:\ If\ x_1\ is\ A_{i1}\ and\ x_2\ is\ A_{i2} \cdots and\ x_n\ is\ A_{im}\ then\ u\ is\ B_i \tag{2}$$

와 같이 표시할 수 있다.

이 때, 가능한 퍼지 규칙의 수(N)은 $N=m^n$이 됨을 알 수 있다. 이런 현상을 퍼지룰의 조합급증 현상이라고 부르고, 이를 해결하는 방법 중 하나가 적은 입력변수를 갖는 제어 장치를 계층적 구조를 이용하여 연결하는 방법이다.

이와 같은 계층적 구조에는 여러 가지가 존재하는 데, 가장 기본적인 계층적 구조는 그림 3과 같다.

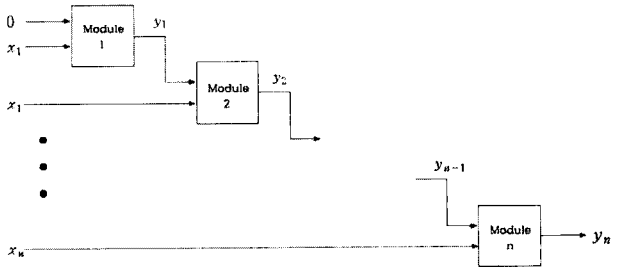


그림 3. 계층적 모델
Fig. 3. Hierarchical Model

입력의 개수를 n이라고 하고 PB, PM, ..., NB 등과 같은 퍼지 변수 A_{ij}는 m개의 레이블로 기술되었다고 할 때, FLU의 개수는 n-1개가 존재한다.

이러한 구조의 경우, 첫 번째와 k 번째 모듈의 i 번째 룰은 각각 아래와 같다.

$$If\ x_1\ is\ A_a^1\ and\ x_2\ is\ A_b^1\ \ THEN\ y_1\ is\ r_k \tag{3}$$

$$If\ x_{k+1}\ is\ A^i\ and\ y_{k-1}\ is\ B^i\ \ THEN\ y_k\ is\ r_k \tag{4}$$

이 때, 가능한 퍼지 규칙의 수(N)은 $N=(n-1)m^2$이 됨

을 알 수 있다. 이러한 구조에서 이전 단의 출력이 다음의 후건부에만 기여하는 구조가 제안되었다[5]. 이 때, 가능한 규칙의 수(N)는 $N=nm$이 된다.

한편, 다입력 시스템에서 발생하는 규모의 비대화를 줄이기 위해 여러 방면이 모색되어 왔는데 그 중 하나가 모듈라 네트워크이다. 이 모듈라 네트워크는 퍼지 시스템과 유사한 구조로 되어 있다. 룰의 전건부에 해당하는 부분은 엑스퍼트 네트워크(Expert Network)이고, 후건부에 해당하는 부분을 게이팅 네트워크(Gating Network) 부분이다. 모듈라 네트워크의 경우 이렇게 복잡한 문제를 부문제로 나누어 EM(Expectation Maximization) 알고리즘을 이용하여 가중치를 부여하는 반면에, 퍼지의 경우에는 퍼지 변수를 정의하면서 이미 부문제로 나누어 놓고 소프트 클러스트링 시키는 방식이다. 다시 말해, 모듈라 네트워크의 경우 입력공간을 적당하게 분할하여 EN이 해당 입력공간에서 학습이 수행되도록 조정해 주며, EN의 출력을 조합하여 최종출력을 생성하도록 스위칭을 수행하는 반면에 퍼지의 경우는 설계자체에서 이미 입력공간을 분할한다. 즉, 모듈라 네트워크의 경우에는 엑스퍼트 네트워크의 개수를 정의해야 하며, 각 엑스퍼트 네트워크가 하는 역할을 학습이 끝나기 전에 알 수 없다. 이에 반해, 퍼지 뉴럴을 이용한 구조의 경우 각 부분의 역할을 이해하기 쉽다.

제안된 구조에서 모듈라의 각 모듈은 퍼지 뉴럴 시스템으로 구성되어 있다. 따라서 바꿔주어야 할 변수들은 입력 멤버쉽 함수의 폭과 위치, 후건부 계수값이다. 멤버쉽 함수와 관련된 변수의 경우, 모듈의 수와는 무관하게 입력과 입력변수의 정의에 의해서만 결정된다. n개 모듈이 존재하고 각 모듈당 c개의 입력과 각 입력당 m개의 퍼지변수를 사용하는 경우에 총 입력의 수는 cn개다. 따라서, 멤버쉽 함수와 관련된 변수는 입력의 수×m×2 = $2cnm$로 표시할 수 있다. 일반적인 구조에서 룰의 수는 m^{cn}개이고, 한 개의 룰에 바꿔야 할 파라미터의 수는 입력+1개이다. 즉, 후건부에 관련된 파라미터의 수는 $m^{cn}\times(cn+1)$개다. 총 파라미터의 수는 $2cnm + m^{cn}(cn+1)$개다.

후건부에만 기여하는 계층적 구조의 경우 룰의 수는 nm^c이 되고 바꾸어 주어야 할 파라미터의 수는 $nm^c(c+2)-m^c$이 된다. 따라서, 총 파라미터의 수는 $2cnm+nm^c(c+2)-m^c$이다.

2.3 학습 알고리즘

제안된 HFNS의 파라미터 조정은 오차 역전파 학습 알고리듬을 이용한다. 비용함수, E를 아래와 같이 정의할 수 있다.

$$E(t) = \frac{1}{2}(y_d(t) - y^*(t))^2 \tag{5}$$

이를 이용하여 최급강하법에 의해 각 변수들을 조정한다.

소속함수의 형태는 다음과 같은 가우시안 함수를 사용하였다.

$$A_{ij}(x_i) = \exp\left(-\frac{1}{2}\left(\frac{x_i - c_{ij}}{w_{ij}}\right)^2\right) \tag{6}$$

여기서, c_{ij} 소속함수의 위치, w_{ij}는 소속함수의 넓이로 조정해야 할 변수이다.

퍼지규칙의 전건부 진리값은 Product-operation 규칙에 따라 다음과 같이 계산된다.

$$\mu_j = \prod_{i=1}^{n} A_{ij}(x_i) = \prod_{i=1}^{n} \exp\left[-\frac{1}{2}\left(\frac{x_i - c_{ij}}{w_{ij}}\right)^2\right] \tag{7}$$

$$\widehat{\mu_j} = \frac{\mu_j}{\sum_{j=1}^{m}\mu_j}$$

여기서, μ_j는 j번째 규칙의 진리값을 의미하며, $\widehat{\mu_j}$는 μ_j의 정규화된 값이다. 그리고 m은 규칙의 수이다. 퍼지 뉴럴 제어기의 후건부는 실제 제어량 발생을 위한 최종 추론값을 계산한다. 각 퍼지 규칙의 추론값은

$$g_j = q_j y_{n-1} + \sum_{i=1}^{n} p_{ij} x_i + r_j \tag{8}$$

와 같이 계산된다. 그리고 최종 추론값은 전건부의 진리값을 고려하여 다음과 같이 계산된다.

$$y = \frac{\sum_{j=1}^{m}\mu_j \cdot g_j}{\sum_{j=1}^{m}\mu_j} \tag{9}$$

n번째 모듈에서 출력 y_n에서 비용함수 E와의 미분값을 $\delta(n)$로 정의하면,

$$\delta(n) = \delta(n+1)\cdot\sum_{j=1}^{J} q_j(n+1) \tag{10}$$

와 같다.

만약, FLU가 마지막 단이라면

$$\delta(n) = \frac{\partial E}{\partial y_n} = -(y_d(t) - y^*(t)) \tag{11}$$

와 같이 계산된다.

최급강하법과 체인 룰을 적용하면 i번째 입력, j번째 룰에 해당하는 퍼지변수에 위치(Center) 갱신 규칙은 아래와 같다.

$$c_{ij}(t+1) = c_{ij}(t) + \Delta c_{ij} \tag{12}$$

$$\begin{aligned}\Delta c_{ij} &= -\eta\cdot\frac{\partial E}{\partial c_{ij}} \\ &= -\eta\cdot\frac{\partial E}{\partial y}\cdot\frac{\partial y}{\partial \mu_j}\cdot\frac{\partial \mu_j}{\partial c_{ij}}\end{aligned} \tag{13}$$

따라서, 갱신될 위치의 변화량은 식 (14)와 같다.

$$\begin{aligned}\Delta c_{ij} &= -\eta\cdot\delta(n)\cdot\frac{1}{\sum_{j=1}^{m}\mu_j}(g_j - y)\cdot\mu_j\cdot\left(\frac{x_i - c_{ij}}{w_{ij}^2}\right) \\ &= -\eta\cdot\delta(n)\cdot(g_j - y)\cdot\widehat{\mu_j}\cdot\left(\frac{x_i - c_{ij}}{w_{ij}^2}\right)\end{aligned} \tag{14}$$

i번째 입력, j번째 룰에 해당하는 퍼지변수에 폭(Width) 갱신 규칙은 아래와 같다.

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij} \tag{15}$$

$$\begin{aligned}\Delta w_{ij} &= -\eta\cdot\frac{\partial E}{\partial w_{ij}} \\ &= -\eta\cdot\frac{\partial E}{\partial y}\cdot\frac{\partial y}{\partial \mu_j}\cdot\frac{\partial \mu_j}{\partial w_{ij}}\end{aligned} \tag{16}$$

따라서, 갱신될 폭의 변화량은 식 (17)과 같다.

$$\Delta w_{ij} = -\eta \cdot \delta(n) \cdot \frac{1}{\sum_{j=1}^{m} \mu_j} (g_j - y) \cdot \mu_j \cdot \frac{(x_i - c_{ij})^2}{w_{ij}^3} \quad (17)$$

나머지 변수에 대해서도 동일한 방법으로 갱신 규칙을 구한다. 갱신될 후건부 파라메터 p_{ij}, q_j, r_j의 변화량은 각각 식 (18-20)과 같다.

$$\Delta p_{ij} = -\eta \cdot \delta(n) \cdot \hat{\mu}_j \cdot x_i \quad (18)$$

$$\Delta q_j = -\eta \cdot \delta(n) \cdot \hat{\mu}_j \cdot y_{n-1} \quad (19)$$

$$\Delta r_j = -\eta \cdot \delta(n) \cdot \hat{\mu}_j \quad (20)$$

위와 같은 방법으로 계층적 구조를 지닌 모듈라 신경망의 구조에 HFNN의 구조 및 학습 알고리즘을, 적용하여 최적화된 HFNN에 의한 모듈라 신경망의 구조를 마련할 수 있다. 본 논문에서는 시스템의 제어 문제에 제안한 구조를 적용하여 모의 실험하였다.

3. Ball and Beam 제어 모의 실험

3.1 Ball and Beam 시스템

Ball and Beam 시스템은 그림 4 와 같다.

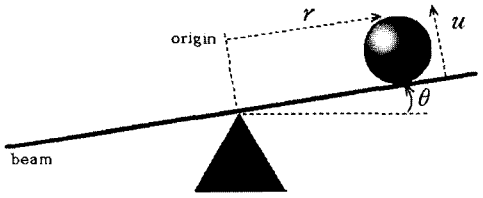


그림 4. The Ball and Beam 시스템
Fig. 4. The Ball and Beam system

여기서, r은 중점으로부터 볼과의 거리, θ는 수평선과 beam과의 기울어진 각이다. θ에 의해 볼의 방향이 결정된다. 따라서 제어 입력 u에 의해 볼의 위치를 조정할 수 있으며, 적절한 u의 조정으로 일정한 볼의 위치를 유지할 수 있다.

$\boldsymbol{x} = (x_1, x_2, x_3, x_4)^T = (r, \dot{r}, \theta, \dot{\theta})^T$라고 정의하면, 출력 y는 r로 정의할 수 있다. 이 시스템을 상태 공간모델로 나타내면 식 (21)과 같다.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ B(x_1 x_4^2 - G \sin x_3) \\ x_4 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u \quad (21)$$

$$y = x_1$$

식(22)와 같이 제어 규칙을 구하였다[6].

$$u^* = \frac{4BGx_4 \cos x_3 + 6BG \sin x_3 - 4x_2 - x_1 - BGx_4^2 \sin x_3}{-BG \cos x_3} \quad (22)$$

3.2 Ball and Beam 제어 모의 실험

본 논문에서는 HFNN 구조의 성능 검증을 위해 Ball and Beam 시스템을 이용 볼의 위치제어를 하였다. 성능 검증을 위한 조건과 사전 처리는 아래와 같다.

[1] 초기화(Initialize)

멤버쉽 함수의 위치는 멤버쉽의 수를 고려하여 등간격으로 설정하고, 폭은 0.5로 설정하며 후건부의 파라메터는 -1에서 1사이의 값을 임의로 발생시켜 정한다.

[2] 범위(Range)의 정의

x_1 : [−5, 5], x_2 : [−2, 2], x_3 : [$-\pi/4$, $\pi/4$], x_4 : [−0.8, 0.8]의 범위로 정의하고 이 값을 최대, 최소로하여 정규화(Normalize)를 시켜서 입력한다.

[3] 학습 데이터 쌍(Training Data)

3000개의 입출력 쌍으로 구성하고 입출력 쌍은 30개의 샘플을 기준하여 임의로 발생시킨다.

[4] Iteration : 500회

[5] 에러의 처리(Rms) $J_u = \frac{1}{3000}\sqrt{\sum_{k=1}^{3000}(u_k^* - u_k)^2}$

[6] 학습 결과

표 1. 학습 결과
Table 1. Result of leaning

	Test 1	Test 2
모듈의 개수	2	4
멤버쉽의 개수	2	2
총 퍼지룰의 수	8	8
η	0.001	0.001
최종 j_u(RMS)	0.00116	0.00342

3.3 실험결과(Test 1)

두 개의 모듈에 각 두 개의 입력, 각 입력당 2개의 멤버쉽 함수를 사용하였다.

따라서, 가능한 룰의 수는 $2^2+2^2=8$이다. 일반적인 계층적 구조의 경우에는 $2^2+2^3=12$의 룰이 필요하고, 일반적인 퍼지로직 시스템의 경우 $2^4=16$개의 룰이 필요하다.

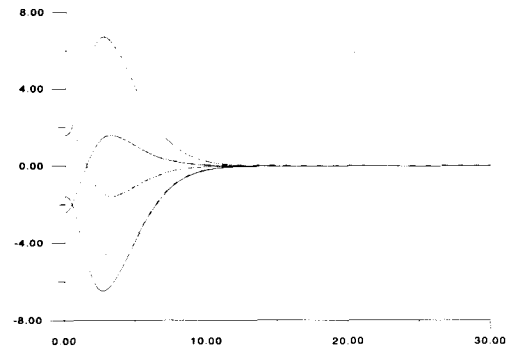
그림 5(a)에서 실선은 선형화 이론에 의거한 결과를 점선은 학습된 제안된 계층적 구조를 이용하여 제어한 경우이다. 또한 그림 5(b)에서 실선은 학습을 위한 데이터이며, 점선은 학습된 결과이다. 그림 5(c)에서는 학습하는 동안의 RMS값이다.

3.4 실험결과(Test 2)

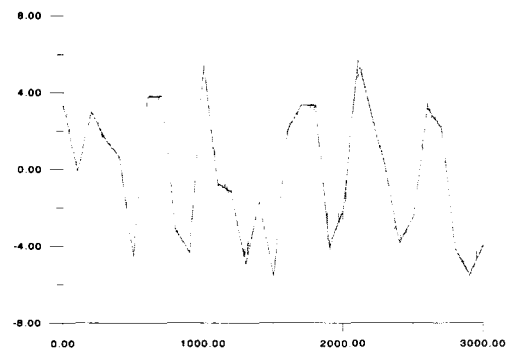
4 개의 모듈에 각 한 개의 입력, 각 입력당 2개의 멤버쉽 함수를 사용하였다.

따라서, 가능한 룰의 수는 $2+2+2+2=8$이다. 일반적인 계층적 구조의 경우에는 $2^2+2^2+2^2=12$의 룰이 필요하고, 일반적인 퍼지로직 시스템의 경우 $2^4=16$개의 룰이 필요하다.

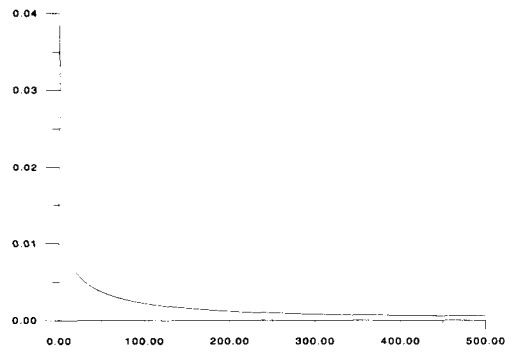
그림 6(a)에서 실선은 선형화 이론에 의거한 결과를 점선은 학습된 제안된 계층적 구조를 이용하여 제어한 경우이다. 또한, 그림 6(b)에서 실선은 학습을 위한 데이터이며 점선은 학습된 결과이다. 그림 6(c)에서는 학습하는 동안의 RMS값이다.



(a) r(t) of the closed loop ball and beam system

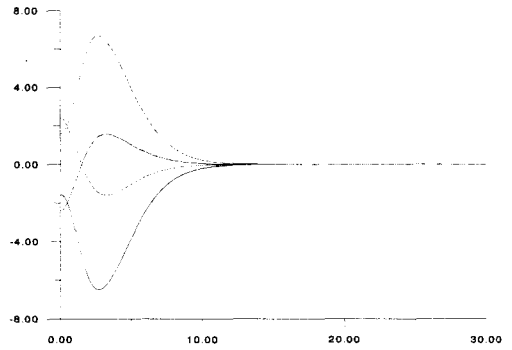


(b) Learning result

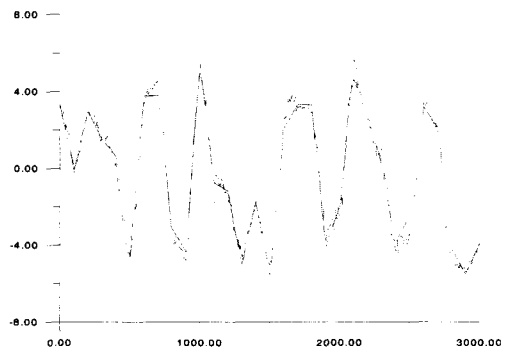


(c) Learning plot

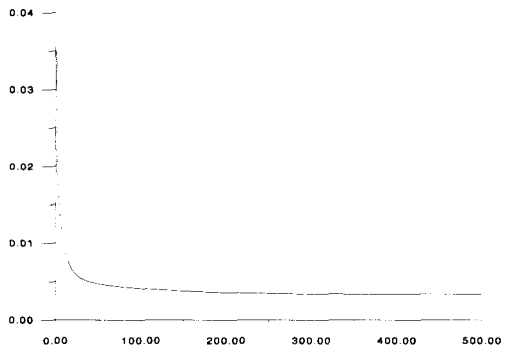
그림 5. 제안된 HFNS을 이용한 결과(2개의 모듈 사용)
Fig. 5. Result with the proposed HFNS and learning plot using two modules



(a) r(t) of the closed loop ball and beam system



(b) Learning result



(c) Learning plot

그림 6. 제안된 HFNS을 이용한 결과(4개의 모듈 사용)
Fig. 6. Result with the proposed HFNS and learning plot using four modules

4. 결 론

입력차원의 증가에 따른 신경회로망 구조의 거대화 문제에 대한 해결책으로 복잡한 문제를 작은 부문제로 나누어 처리하는 모듈라 네트워크가 제안되었다. 모듈라 네트워크는 입력공간을 분할하고 출력에 가중치를 주어 입력공간에 해당하는 출력을 더욱 활성화시키는 구조이며 일반적으로 퍼지 이론과 유사하다. 하지만 퍼지의 경우 입력공간을 전문가가

직접 분할하고, 가중치를 부여해야 한다는 단점이 있고 이러한 단점을 극복하기 위해 퍼지 뉴럴 네트워크가 제안되었다. 하지만 이 경우에도 입력의 증가에 따른 규칙수가 지수적으로 증가하는 문제점이 지적되고 있다. 이러한 문제는 계층적 구조를 설계함으로서 해결할 수 있지만 입출력 관계가 불명확하며 설계하기에 어려운 점이 존재한다. 이에 본 논문에서는 모듈라 구조의 최적 설계를 위해 계층적 구조의 특징을 살리고 단점을 해결할 수 있는 구조로서 앞선 모듈의 출력이 후건부에만 기여하는 구조를 사용하였다. 제안된 구조의 특징은 룰의 수가 적다는 점과 새로운 문제에 적용하는 경우 설계가 용이하다는 장점을 가진다.

이와 같은 구조를 Ball and beam 제어에 적용하여 적은 수의 룰과 효과적인 모듈라 구조를 마련할 수 있음을 확인하였다.

참 고 문 헌

[1] Simon Haykin, *Neural Networks - A Comprehensive Foundation*, Macmillian College Publishing Company Inc., 1994.

[2] CHIN-TENG LIN and C. S. GEORGE LEE, *Neural Fuzzy Systems - A Neuro Fuzzy Synergism to intelligent Systems*, Prentice Hall PTR, 1996.

[3] Campello, R.J.G.B. and Amaral, W.C., "Optimization of hierarchical neural fuzzy models," *Proc. of the IEEE-INNS-ENNS Int. Conf.* on Neural Networks, Vol 5 , pp. 8-13. 2000.

[4] Yongkiang Pan; Haiying Zhmg; Xiaoian Wang, "A new fuzzy neural network for nonlinear approaching," *Proceedings of the 4th World Congress on Intelligent Control and Automation*, vol. 3, pp. 1985 -1989, 2002.

[5] Joo, M.G., "Hierarchical fuzzy system with constraints on the fuzzy rule and its approximation property," *POSTECH, Ph.D. Dissertation*, 2001.

[6] Hauser, J.; Sastry, S.; Kokotovic, P. , "Nonlinear control via approximate input-output linearization: the ball and beam example," *IEEE Trans. on Automatic Control*, Vol. 37 Issue: 3, pp. 392 -398, 1992.

저 자 소 개

김성주(Seong-Joo Kim)
2002년 제12권 4호 참조

전홍태(Hong-Tae Jeon)
현재 중앙대학교 전자전기공학부 교수