

자율분산 신경망을 이용한 비선형 동적 시스템 식별

論 文

45~4~17

Identification of Nonlinear Dynamical Systems Based on Self-Organized Distributed Networks

崔鍾洙*·金炯奭**·金聖中***·權五臣§·金棕晩§§

(Jong-Soo Choi · Hyong-Suk Kim · Sung-Joong Kim · Oh-Shin Kwon · Jong-Man Kim)

Abstract - The neural network approach has been shown to be a general scheme for nonlinear dynamical system identification. Unfortunately the error surface of a Multilayer Neural Networks(MNN) that widely used is often highly complex. This is a disadvantage and potential traps may exist in the identification procedure. The objective of this paper is to identify a nonlinear dynamical systems based on Self-Organized Distributed Networks (SODN). The learning with the SODN is fast and precise. Such properties are caused from the local learning mechanism. Each local network learns only data in a subregion. This paper also discusses neural network as identifier of nonlinear dynamical systems. The structure of nonlinear system identification employs series-parallel model. The identification procedure is based on a discrete-time formulation. Through extensive simulation, SODN is shown to be effective for identification of nonlinear dynamical systems.

Key Words : Nonlinear Dynamical Systems, System Identification, Self-Organized Distributed Networks(SODN), Multilayer Neural Networks(MNN).

1. 서 론

시스템 식별은 입출력 데이터로부터 동적 시스템의 모델을 결정하는 것으로서 모델에 대한 지식은 고성능 제어 시스템의 설계 및 구현에 필수적이다. 많은 실제 시스템은 복잡한 비선형 특성을 갖고 있고 기존의 선형이론을 사용하여 비선형 시스템을 다루는 데에는 한계가 있어 만족할만한 결과를 얻지 못하였다. 입출력 데이터로부터 복잡한 비선형 사상관계를 학습할 수 있는 능력을 갖고있는 신경망은 매우 복잡한 비선형성을 갖는 시스템을 모델링하고 식별하는데 이용하게 되었다[1~3]. 비선형 시스템의 식별 및 제어에 이용되는 신경망의 대부분을 차지하는 모델은 오차 역전달 학습법을 이용한 다층 신경망(Multilayer Neural Networks, 이하 MNN)이다. Narendra등[4]은 비선형 시스템 식별을 미지의 비선형 시스템 식별과 제어 문제에 일반화된 MNN과 선형 필터가 직렬 및 되먹임 구조로 구성되는 모델을 적용하였고, 식별 모델로서 직병렬 식별 및 병렬 식별 모델구조를 도입하였다. Qin등은 MNN과 순환신경망(recurrent neural networks)을 비선형 시스템 식별에 적용하였고 개별학습(pattern learning) 및 일괄학습(batch learning)의 특성을 해석하였다[5]. 또한 Sastry등은 메모리 뉴런 신경망(Memory Neuron Network)이라는 내부 메모리를 갖는 순환신경망 모델을 제안하여 비선형 시스템 식별 및 제어에 적용하였

다[6]. 이와 같이 MNN과 같은 완전연결 신경망(fully connected neural networks)은 전체적 일반화 능력이 우수하기 때문에 비선형 시스템 식별 및 제어에 가장 많이 사용된다. 그러나, 완전연결 신경망은 전체적 근사화 특성이 좋은 반면에 지역적인 변동을 갖는 비선형성이 매우 심한 시스템을 학습하는 것이 매우 어렵다. 이런 경우 만족할만한 일반화 특성을 얻기 위해서 충분히 많은 훈련 데이터를 필요로 하게 되고, 특성이 현저하게 다른 영역에 한정된 연결강도로 학습하기 때문에 일시적인 혼란(temporal crosstalk)을 일으키게 되어 특성이 서로 다른 영역 모두를 만족시키는 연결강도 값을 찾아야 하는 어려움에 봉착한다. 다행히도 훈련 영역이 서로의 경계를 넘나든다면 전체적인 일반화 특성을 얻을 수 있지만 학습시간이 길어지고 지역적 일반화 특성이 떨어지게 된다. 일시적 혼란 문제는 복잡한 비선형 시스템 식별과 제어 문제에서 두드러진다. 시계열 예측과 같은 시스템 식별 문제는 인접한 영역간의 비선형성이 매우 심해서 학습이 어렵다. 또한 제어 분야에서 비선형 동적 시스템은 상태공간을 통해서 상대적으로 천천히 이동하는 경향이 있기 때문에 훈련 데이터는 주로 구역화된 영역으로 표현되며, 플랜트의 동특성이 동작점에 따라서 변화한다면 일시적인 혼란현상은 피할 수가 없게 된다.

그러나, RBF(Radial Basis Function)신경망과 같은 지역연결 신경망(locally connected neural networks)은 훈련 데이터들을 특성에 따라 분류하고 특성별로 다른 연결강도를 사용하여 지역적 일반화 성능을 개선하는 구조의 신경망이다. 이 지역연결 신경망은 유사한 특성을 갖는 지역적 데이터간에만 공통 파라미터를 갖게 되므로 학습이 훨씬 쉽고 학습속도도 매우 빠르다는 장점이 있어서 부분적 비선형성이 심한 시스템 식별에 유용하다[7,8]. 그러나, 이러한 지역연결 신경망은 지역적 일반화 특성이 좋은 반면 전체적 일반화 특성이 떨어지는 단점이

* 正 會 員 : POSCON 技術研究所 先任研究員

** 正 會 員 : 全北大 工大 制御計測工學科 助教授 · 工博

*** 正 會 員 : 全北大 工大 制御計測工學科 教授 · 工博

§ 正 會 員 : 群山大 工大 制御計測工學科 副教授 · 工博

§§ 正 會 員 : 全北大 大學院 電氣工學科 博士課程 修了

接受日字 : 1995年 8月 2日

最終完了 : 1996年 3月 4日

있다. 한편, Jacobs등은 작업을 기능적으로 독립된 작업으로 분리하여 각각의 작업을 학습하기 위해 여러 개의 신경망을 사용하는 모듈화 신경망(Modular Neural Networks)을 제안하여 패턴 인식 및 로봇제어에 적용하였다[9].

저자들은 전체연결 신경망 및 지역연결 신경망의 장점을 결합한 자율분산 신경망(Self-Organized Distributed Networks, 이하 SODN)을 제안하였다[10]. SODN은 입력 공간을 학습 데이터의 분포와 학습의 어려운 정도에 따라 자율적으로 분할하고 각각 분할된 공간에 부분 신경망을 분산 배치하여 지역적 학습을 수행하게되므로 학습 속도가 빠르다. 또한 SODN은 학습 대상을 정확히 학습하는 지역연결 신경망의 구조를 갖고 있으며 일반화 능력이 우수한 완전연결 신경망의 장점을 갖춘 신경망으로서 MNN 및 RBF 신경망에 비해 비선형 함수 근사화 특성이 매우 우수하다[10~13]. 이 논문에서는 이러한 SODN의 특성을 이용하여 비선형 시스템의 식별에 응용한다. 여러 가지 복잡한 비선형성을 갖는 동적 시스템에 대한 시뮬레이션을 통해서 SODN이 비선형 시스템 식별기로서 유용함을 입증한다.

이 논문의 구성은 다음과 같다. 2절에서는 비선형 시스템의 식별에 대하여 간략히 언급하였으며, 3절에서는 SODN의 구조 및 학습방법을 소개하였다. 4절에서는 시뮬레이션을 통해서 SODN을 이용한 비선형 동적 시스템 식별이 MNN보다 학습속도가 매우 빠름을 보였고, 다입력 다출력 시스템(MIMO)의 식별에도 유용함을 입증하였다. 또한 비선형이 심한 시계열 시스템 식별에 적용하였다. 5절에서는 논문의 결과를 요약하였다.

2. 비선형 시스템 식별

설명의 편의를 위해서 SISO(single-input/single output)의 이산 시간 비선형 동적 시스템을 고려해 보자.

$$y(k+1) = f(y(k), y(k-1), \dots, y(k-m+1), u(k), u(k-1), \dots, u(k-n+1)) \quad (1)$$

여기에서 $y(k) \in \mathbf{R}^1$ 와 $u(k) \in \mathbf{R}^1$ 는 각각 시스템의 출력과 입력이고, $f: \mathbf{R}^{m+n} \rightarrow \mathbf{R}^1$ 는 미지의 비선형 함수이다. 양의 정수 m 과 n 은 시스템 출력과 입력의 최대 시간지연을 나타낸다. 플랜트는 입력의 존재하에서 BIBO(bounded-input bounded-output) 안정하다고 가정한다. 그러므로 플랜트의 모든 신호는 균등하게 유계(uniformly bounded)이다. 따라서 플랜트 입력은 일정 크기의 값으로 제한된다. 즉, 시간 k 에 대하여 $u_{\min} \leq u(k) \leq u_{\max}$ 이다. 이 논문에서는 그림 1과 같이 직병렬 모델(series-parallel model) 구조를 사용하여 비선형 시스템을 식별한다[4]. 이 경우 직병렬 구조에 의한 입출력 관계는 다음 식과 같이 표현할 수 있다.

$$\hat{y}(k+1) = F(y(k), y(k-1), \dots, y(k-m+1), u(k), u(k-1), \dots, u(k-n+1)) \quad (2)$$

여기에서 $\hat{y}(k+1)$ 은 시스템의 식별된 출력 즉, 신경망의 출력이고, F 는 신경망에 의해서 표현되는 비선형 사상이다. 위 식으로부터 알 수 있듯이 직병렬 식별 구조는 플랜트의 출력이 식별 모델로 되먹임되어 식별기의 입력이 된다. 그림 1에서 D

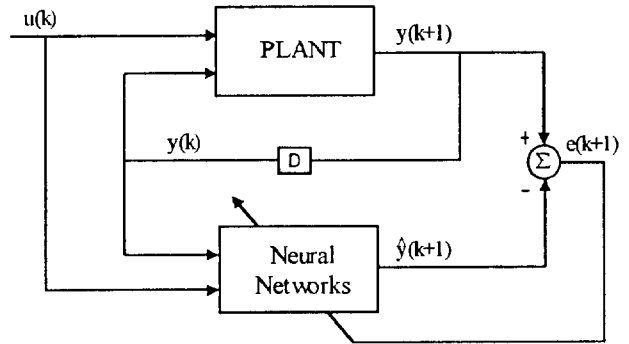


그림 1 비선형 시스템의 직병렬 식별 구조
Fig. 1 A Structure of series-parallel identification for nonlinear systems

는 플랜트 입출력 신호의 시간지연 요소를 나타낸다. 그러므로 플랜트 입력 및 출력 벡터의 시간지연값이 신경망의 입력 벡터를 이룬다. 비선형 시스템 식별의 목적은 시스템 출력과 식별기 출력 사이의 오차 $e(k) = y(k) - \hat{y}(k)$ 를 최소화하는 것이다. 신경망의 출력 벡터($\hat{y}(k), \hat{y}(k-1), \dots, \hat{y}(k-m+1)$)가 신경망의 입력으로 작용하는 병렬 식별구조에 비해 직병렬 식별 구조는 몇 가지 장점이 있다. 첫째, 플랜트가 BIBO 안정하다고 가정할 경우 식별 과정에서 이용되는 모든 입력신호는 유계이다. 둘째, 모델 내에서 되먹임 루프가 존재하지 않기 때문에 계산과정을 줄일 수 있는 정적 역전달 알고리즘이 파라미터를 조정하는데 이용될 수 있다. 셋째, 출력 오차 $e(k)$ 가 점근적으로 $\lim_{k \rightarrow \infty} e(k) = 0$ 이 된다고 가정하면 직병렬 모델은 병렬 모델에 의해서 대체될 수 있다. 위의 특징을 기초로 신경망을 이용한 시스템 식별 구조에서 직병렬 모델을 많이 사용하고 있다.

3. 자율분산 신경망(SODN)

3.1 SODN의 구조

SODN의 구조는 그림 2와 같으며 학습 신경망(Learning Networks)과 부분망의 분산 배치용인 분산 신경망(Distribution Networks)으로 구성된다. SODN은 학습 신경망 내의 MNN 한 세트와 분산 신경망내의 유니트 한 개는 짝이 되어 한 세트의 부분 신경망을 구성하며 분할된 공간의 함수 학습을 수행한다. 입력 벡터는 학습 신경망과 분산 신경망에 공통으로 연결되고 학습 신경망과 분산 신경망의 출력들은 상호 짝이 되는 신경망의 출력과 곱하여 연결된다. SODN에 입력 벡터가 인가되면 분산 신경망 내의 유니트 중 입력 벡터와 가장 유사한 연결강도 벡터를 갖는 유니트만 활성화되며, 이 유니트는 짝을 이루는 학습 신경망의 출력과 선형결합을 통해서 전체 신경망의 출력이 된다. 그림 2에서 굵은 실선은 주어진 입력 벡터에 의해서 활성화된 분산 신경망의 출력과 이것에 의해서 선택된 부분 학습 신경망의 출력이다. 점선으로 연결된 부분 신경망은 새로 생성된 부분 신경망으로 SODN만이 갖는 특징중의 하나이다. 비선형성이 강한 시스템에는 학습이 상대적으로 어려운 영역과 용이한 영역이 존재하므로 학습이 어려운 영역에는 추가적인 부분 신경망을 생성시켜서 그 영역의 학

습을 용이하게 하고 학습에 기여도가 낮은 부분 신경망은 소멸시켜 재활용함으로써 신경망을 효율적으로 이용할 수 있게 하는 구조를 갖고 있다.

학습 신경망은 두세 개의 은닉층 유닛을 갖는 소규모의 MNN과 제한망(LN)이 결합된 집합으로 구성되어 있다. 시간 k 에서 MNN의 $l(1 \leq l \leq L)$ 층에 있는 j 번째 유닛의 출력을 $x_j^l(k)$ 라 하면 출력은

$$\begin{aligned} x_j^l(k) &= \Psi\left(\sum_i w_{ji}^l(k) x_i^{l-1}(k) + b_j^l(k)\right) \\ &= \Psi(\text{net}_j^l(k)) \end{aligned} \quad (3)$$

로 계산된다. 여기서 $w_{ji}^l(k)$ 는 $l-1$ 층의 i 번째 유닛과 l 층의 j 번째 유닛 간의 연결강도이고, $b_j^l(k)$ 는 j 번째 유닛의 바이어스이다. 또한 $\text{net}_j^l(k)$ 는 입력 값의 총합이고, $\Psi(\cdot)$ 는 시그모이드 함수로 표현되는 유닛의 활성화 함수이다.

$$\Psi(\text{net}) = \frac{1}{1 + e^{-\text{net}}} \quad (4)$$

각 부분 학습 신경망의 출력은 MNN의 출력에 LN이 직렬로 연결되어 있어서 목표 값의 동작범위로 제한을 받는다. LN은 MNN 출력단에 연결되어 있고 환산 연결강도(scaling weight)와 바이어스를 갖는 선형 퍼셉트론으로 구성되어 있다. 환산 연결강도는 목표 값의 동작범위이고 퍼셉트론의 활성화 함수는 램프함수이며, 바이어스는 목표 값의 최소치이다. $m(1 \leq m \leq M)$, 여기에서 M 은 부분 신경망의 수)번째 학습 신경망에서 MNN의 $n(1 \leq n \leq N)$, 여기에서 N 은 출력 유닛의 수)번째 출력을 $x_n^l(k)$ 이라 하면, 학습 신경망의 출력 $y_{nm}(k)$ 는 다음과 같다.

$$\begin{aligned} y_{nm}(k) &= f(x_n^l(k) s_m(k) + b_m(k)) \\ &= x_n^l(k) s_m(k) + b_m(k) \end{aligned} \quad (5)$$

$s_m(k)$ 는 환산 연결강도으로써 동작범위의 최대 값과 최소 값의 차이이고, 바이어스 $b_m(k)$ 는 동작범위의 최소 값이므로 식 (5)는 다음과 같다.

$$y_{nm}(k) = x_n^l(k) (t_{\max}(k) - t_{\min}(k)) + t_{\min}(k) \quad (6)$$

여기에서, $x_n^l(k)$ 의 범위는 $0.0 \leq x_n^l(k) \leq 1.0$ 이다. 그러므로 $y_{nm}(k)$ 의 범위는 $t_{\min}(k) \leq y_{nm}(k) \leq t_{\max}(k)$ 로 제한된다. 환산 연결강도와 바이어스 값은 매 학습주기 마다 선택된 부분 신경망 영역에서 학습데이터를 통해서 갱신된다.

한편, 분산 신경망은 0 또는 1을 출력하는 경쟁망(Competition Network)으로서 각 유닛 내에 신경망의 생성/소멸에 관련된 에너지 저장소(Energy Storage)를 갖고 있다. 분산 신경망은 단층 신경망으로 구성되며 각 유닛은 입력단으로부터 연결강도를 통하여 연결된다. 즉, fan-out 유닛으로부터 입력신호를 받아들이는 M 개의 자율신경 유닛으로 구성된다. 입력

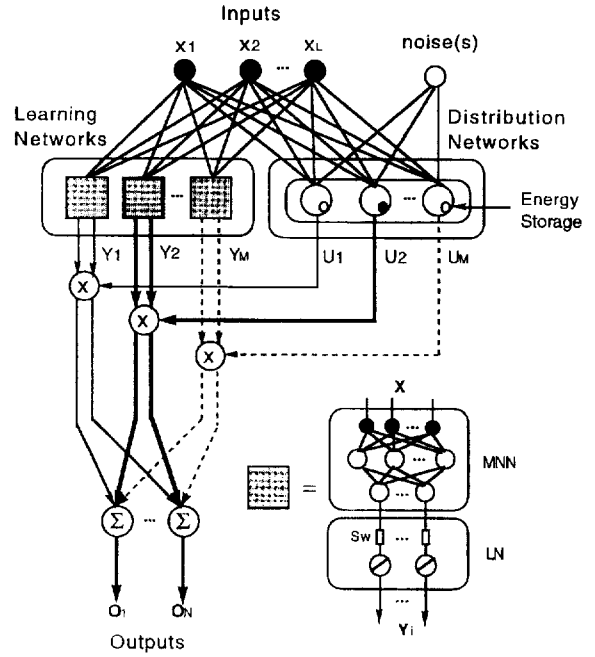


그림 2 SODN의 구조

Fig. 2 The structure of Self-Organized Distributed Networks

벡터에 대하여 m 번째 자율신경 유닛의 중심위치(연결강도)를 $c_m(k)$ 라 하면, 각 유닛은 다음 식과 같이 입력강도 $I_m(k)$ 를 계산한다.

$$I_m(k) = D(c_m(k), \mathbf{x}(k)) \quad (7)$$

여기에서 $D(\cdot)$ 는 유클리드 거리 함수이며, 잡음은 고려하지 않았다. 분산 신경망의 입력강도 $I_m(k)$ 가 계산되면 가장 작은 연결강도를 갖는 유닛을 찾기 위해 서로 경쟁을 한다. 다시 말하면, 입력벡터 \mathbf{x} 에 가장 가까운 연결강도 $c_m(k)$ 를 찾는 것이다. 경쟁 방법은 억제적 연결을 이용한다. 경쟁이 끝나고 winner 유닛이 결정되면 winner 유닛의 출력 $u_m(k)$ 만 1이 되어 활성화한다.

j 번째 출력 유닛에 대한 입력의 총합을 $\text{net}_j(k)$ 라 하면

$$\text{net}_j(k) = \sum_{m=1}^M y_{jm}(k) u_m(k) \quad (8)$$

이다. SODN에 대한 j 번째 출력 유닛의 최종 출력 $o_j(k)$ 는

$$\begin{aligned} o_j(k) &= \Psi(\text{net}_j(k)) \\ &= \Psi\left(\sum_{m=1}^M y_{jm}(k) u_m(k)\right) \end{aligned} \quad (9)$$

가 된다. 여기서 y 는 학습 신경망의 출력이고 Ψ 는 출력단의 활성화 함수이다. 그러므로, 출력단은 입력이 \mathbf{y} 이고 연결강도 벡터가 \mathbf{u} 이며 활성화 함수가 Ψ 인 신경망과 동가이다.

3.2 SODN의 학습

신경망 학습의 목표는 비용함수를 최소화하는 연결강도 벡터

를 찾는 것이다. 대부분의 학습법은 근본적으로 경사에 기초한 최적화법이다. 학습 방법은 개별학습 및 일괄학습으로 나뉜다 [5]. 이 논문에서는 일괄학습을 사용하여 SODN을 학습시킨다. SODN은 학습 신경망의 감독학습과 분산 신경망의 비감독학습의 복합학습 구조로 이루어져 있다.

주어진 동적 시스템의 훈련 데이터 $\{t(k), x(k)\}$, $k=1, 2, \dots, T$ 에 대하여 일괄학습을 적용할 경우 비용함수는 다음과 같은 자승 오차함수로 나타낼 수 있다.

$$E = \frac{1}{2} \sum_{k=1}^T \sum_{j=1}^J (t_j(k) - o_j(k))^2 \quad (10)$$

여기에서 $t_j(k)$ 는 동적 시스템의 목표출력이고 $o_j(k)$ 는 SODN의 출력이다.

학습 신경망은 감독학습으로써 오차 역전과 학습 알고리즘을 사용하여 학습하는데, 역전과 학습은 합자승 오차 함수의 최대 경사면 방향으로 연결강도를 점차 변화시켜가는 학습 원리이므로 입력 패턴에 대해 연결강도의 변화는 다음과 같이 계산된다.

$$\begin{aligned} w_{ji}(k+1) &= w_{ji}(k) - \eta \frac{\partial E(k)}{\partial w_{ji}(k)} \\ &= w_{ji}(k) + \eta \delta_{ji}(k) x_i(k) \end{aligned} \quad (11)$$

여기에서 η 는 학습률이다. 위 식에서 연쇄법칙을 사용하면

$$\frac{\partial E(k)}{\partial w_{ji}(k)} = \frac{\partial E(k)}{\partial net_j(k)} \frac{\partial net_j(k)}{\partial w_{ji}(k)} \quad (12)$$

가 된다. 출력단 또는 은닉층에서 j 번째 유니트로부터 역전파될 오차 벡터 $\delta_j(k)$ 는

$$\delta_j(k) = \begin{cases} (t_j(k) - o_j(k)) \Psi'(net_j(k)), & \text{출력단에 대하여} \\ \Psi'(net_j(k)) \sum_k \delta_k(k) w_{kj}(k), & \text{은닉층에 대하여} \end{cases} \quad (13)$$

이다. j 번째 출력 유니트로부터 역전파될 오차벡터 $\delta_j(k)$ 를 계산하기 위해서 식(9)와 (13)을 이용하면

$$\begin{aligned} \delta_j(k) &= error_j(k) o_j(k)'(net_j(k)) \\ &= (t_j(k) - o_j(k)) \Psi'(net_j(k)) \end{aligned} \quad (14)$$

이다. 또 모든 출력 유니트는 선형 결합기로서 그 활성화 함수 Ψ 는 램프함수이므로 미분값 Ψ' 는 1이다. 따라서 식 (14)는

$$\delta_j(k) = error_j(k) = t_j(k) - o_j(k) \quad (15)$$

가 된다. 또 각 학습 신경망에 역전파될 오차 성분은 등가적 연결강도 벡터 u 를 따라 전파시켜 얻을 수 있으며, 이 때 i 번째 부분 신경망에 역전파될 오차 성분은 식 (13)과 (15)에 의해서

$$\begin{aligned} \delta_{ji}(k) &= \Psi' \sum_{j=1}^J \delta_j(k) u_i(k) \\ &= u_i(k) \Psi' \sum_{j=1}^J (t_j(k) - o_j(k)) \end{aligned} \quad (16)$$

가 된다. 여기에서의 $u_i(k)$ 는 분산 신경망의 출력벡터 u 중의 i 번째 요소로서 분산 신경망 중의 i 번째 유니트가 선택된 경우에만 값이 1이고 나머지 경우에는 0인 binary 값이다. 따라서 최종 출력에서 발생된 오차는 분산 신경망에 의해 선택된 부분 신경망에만 전파되어 연결강도 벡터가 학습한다.

다음, 분산 신경망은 비감독 학습으로써 자율신경 유니트 사이의 역제적 연결을 이용하여 입력 벡터가 인가되면 입력벡터와 가장 유사한 연결강도를 갖는 유니트만 활성화되는 "Winner-Take-All (WTA)" 규칙을 이용한다. 분산 신경망내의 i 번째 유니트 위치를 c_i 라고 하면, 입력벡터 x 가 인가될 때의 유니트 위치이동은 다음과 같은 학습규칙을 따른다.

$$c_i(k+1) = \begin{cases} (1-\alpha)c_i(k) + \alpha x, & \text{for winning unit} \\ c_i(k), & \text{otherwise} \end{cases} \quad (17)$$

여기에서 $\alpha(0 < \alpha < 1)$ 는 학습률로서 큰 초기값으로 부터 점차 작은 최종값으로 학습 스텝(또는 반복횟수)에 따라 점차 줄어드는 함수로 선정한다면 초기에 시스템은 정확한 유니트의 위치를 빠르게 학습할 수 있다. 그러나, α 가 크다면 각 학습 스텝에 의해 야기되는 동요가 크다. 그러므로, 유니트의 위치가 점진적으로 평행상태로 안정화하기 위해서는 α 를 점차 0으로 수렴하도록 감소시켜야 한다. 이 학습 규칙은 활성화된 유니트의 연결강도 벡터만 입력 벡터 방향으로 이동하게 되므로 학습한 후 유니트들은 현재의 위치에서 가장 가까운 데이터군의 중심으로 이동하게 된다.

4. 시뮬레이션 및 검토

SODN이 비선형 시스템을 식별할 수 있음을 보이기 위해서 세 개의 비선형 동적 시스템 모델을 예제로 적용하였다[4,8]. 예제 1과 2는 SODN에서 M은 20이고, 예제 3에서는 30을 사용하였다. 학습 신경망의 학습계수 η 는 여러 차례의 반복실험을 통해서 최적값을 구하였으며 0.1로 선정하였다. 모멘텀 계수는 0.9를 사용하였다. 또한 분산 신경망의 학습계수 α 는 예제 1과 3에 대하여는 0.9를 사용하였고, 예제 2에서는 분산 신경망의 중심위치가 점진적으로 안정화하도록 1/iteration로 하였다. 학습 신경망의 초기 연결강도는 영역이 [-0.3, 0.3]인 랜덤값으로 초기화하였다. 예제 1, 2에서 SODN과 MNN의 일반화 특성은 실험결과의 평균오차로 비교하였다.

$$\text{평균오차} = \frac{\sum_{k=1}^K |y(k) - \hat{y}(k)|}{K} \quad (18)$$

[예제1] 식별하려는 플랜트는 다음과 같은 차분방정식으로 표현된다.

$$y(k+1) = 0.3y(k) + 0.6y(k-1) + f(u(k)) \quad (19)$$

여기에서, 미지의 비선형 함수는 $f(u) = u^3 + 0.3u^2 - 0.4u$ 이다. 플랜트를 식별하기 위해서 아래와 같은 차분 방정식으로 표현되는 직병렬 모델을 사용한다.

$$\hat{y}(k+1) = 0.3y(k) + 0.6y(k-1) + F(u(k)) \quad (20)$$

여기에서 SODN의 입력과 출력은 각각 $u(k)$ 및 $F(u(k))$ 이다. 훈련에 사용된 데이터는 (-1, 1) 사이에 균등하게 분포하는 1,000개의 불규칙 입력으로 그림 3에 나타냈다. SODN의 학습 성능을 실험하기 위해서 MNN과 결과를 비교하였는데 실험에 사용한 MNN은 SODN과 비슷한 파라미터수를 갖도록 각각 한 개의 입출력 유니트를 갖고 12개의 유니트로 구성된 두 개의 은닉층을 갖는 4층 신경망(1-12-12-1)으로 구성하였다. MNN의 학습 파라미터는 반복실험을 통하여 결정하였으며 학습률 및 모멘텀 계수로 각각 0.01, 0.9를 사용하였다. 그림 4는 SODN과 MNN의 학습결과인 SSE(Sum-Squared Error)를 비교한 곡선이다. SODN은 MNN보다 빨리 수렴함을 알 수 있다. 100 iteration 동안의 학습 소요시간은 SODN이 21초이고 MNN이 86초로, 구조적으로 지역연결 신경망인 SODN은 학습 시간이 MNN에 비해 4배 이상 빠름을 알 수 있다. 이 때 시뮬레이션은 PC(Pentium-60MHz) 상에서 수행하였다. 그림 5는 시스템 출력과 SODN에 의해서 식별된 결과를 보여준다. 테스트에 사용한 입력 값은

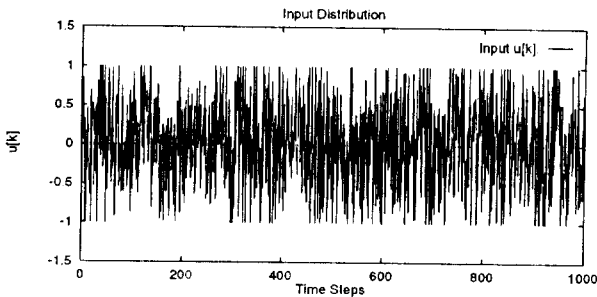


그림 3 예제 1의 학습에 사용한 입력 $u(k)$ 의 분포
Fig. 3 Distribution of Input $u(k)$ used for training Example 1

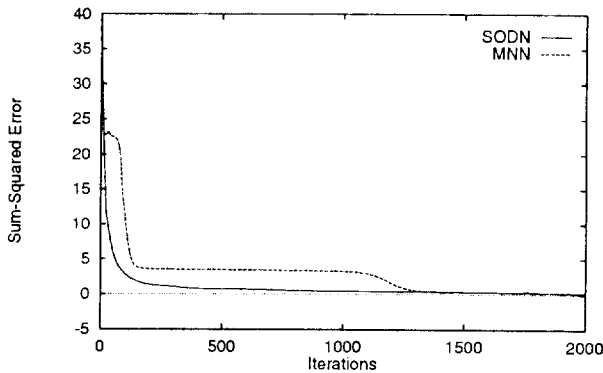


그림 4 예제1에 대한 SODN과 MNN의 학습성능 비교
Fig. 4 Learning performance comparison to SODN and MNN for Example 1

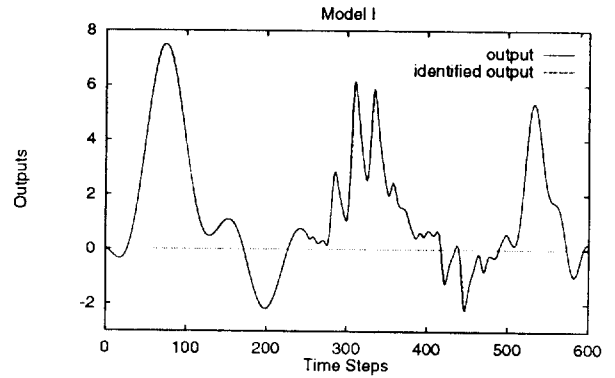


그림 5 예제 1에 대한 시스템 식별 결과
Fig. 5 Result of system identification for Example 1

표 1 예제 1의 학습조건 및 결과
Table 1 Learning conditions and results for the Example 1

	파라미터 수	학습오차 (SSE)	평균오차	학습회수
SODN	160	0.02025	0.01951	10,000
MNN	183	0.02251	0.02042	10,000

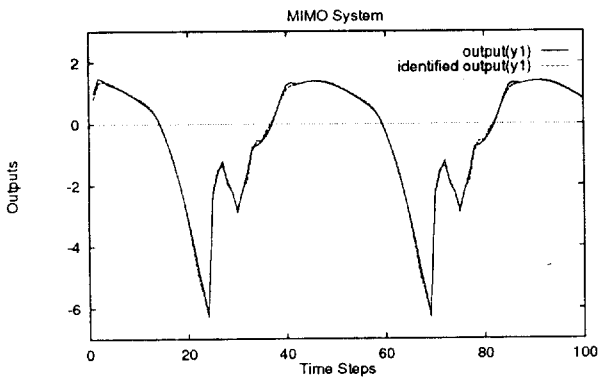
$$u(k) = \begin{cases} \sin(2\pi k/250), & 0 < k \leq 500 \\ 0.5 \sin(2\pi k/250) + 0.5 \sin(2\pi k/25), & 500 < k \leq 800 \end{cases} \quad (21)$$

이다. 결과로부터 알 수 있듯이 플랜트의 출력과 SODN에 의해서 식별된 출력이 육안으로는 오차를 구별할 수 없을 정도로 식별 능력이 우수함을 보여준다. SODN과 MNN의 비교내용은 표 1에 정리하였다.

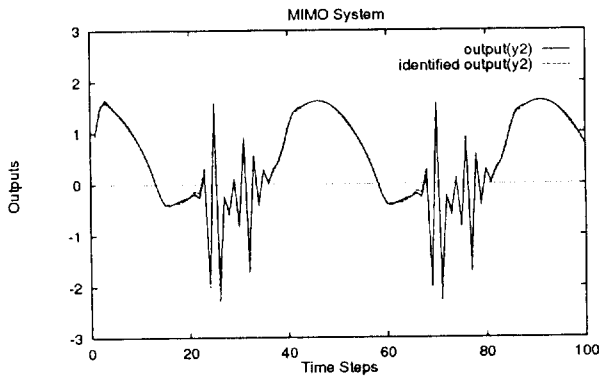
[예제2] 이 예제는 SISO 플랜트와 동일한 방법으로 MIMO(multi-input/multi-output) 플랜트를 식별할 수 있음을 보여준다. 플랜트는 다음 방정식으로 표현된다.

$$\begin{aligned} y_1(k+1) &= \frac{y_1(k)}{1+y_2^2(k)} + u_1(k), \\ y_2(k+1) &= \frac{y_1(k)y_2(k)}{1+y_2^2(k)} + u_2(k) \end{aligned} \quad (22)$$

입출력 관계는 네 개의 입력($y_1(k), y_2(k), u_1(k), u_2(k)$)과 두 개의 출력($y_1(k+1), y_2(k+1)$)을 갖는 SODN을 사용한다. 예제 1과 같이 훈련에 사용된 데이터는 [-1, 1] 사이에 균등하게 분포하는 불규칙 입력을 갖는 1,000개의 데이터를 사용하였다. 이 예제에서도 SODN의 학습성능을 실험하기 위해서 MNN과 결과를 비교하였는데 실험에 사용한 MNN은 SODN과 비슷한 파라미터수를 갖도록 네 개의 입력 유니트를 갖고 15개의 유니트로 구성된 두 개의 은닉층 및 두 개의 출력 유니트를 갖는 4층 신경망(4-15-15-2)으로 구성하였다. MNN의 학습 파라미터는 예제 1과 동일한 값을 사용하였다. 그림 6에 나타낸 결과는 테스트 입력벡터로 $[\sin(2\pi k/25), \cos(2\pi k/25)]^T$ 를 사용한 경우이다. 결과로부터 알 수 있듯이 플랜트의 출력과



(a)



(b)

그림 6 예제 2에 대한 시스템 식별 결과
Fig. 6 Result of system identification for Example 2

표 2 예제 2의 학습조건 및 결과
Table 2 Learning conditions and results for the Example 2

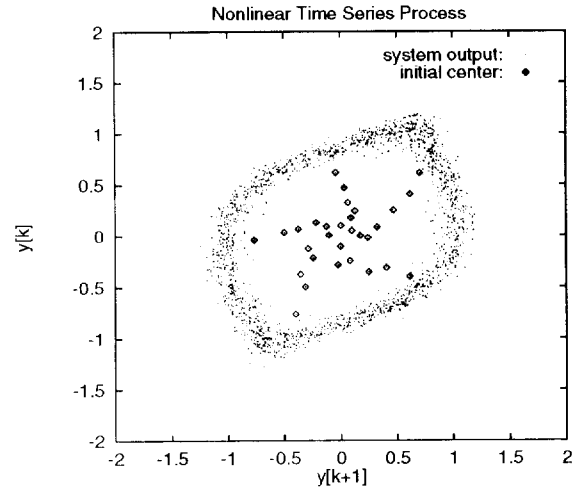
	파라미터 수	학습오차 (SSE)	평균오차		학습회수
			y1	y2	
SODN	340	0.04686	0.03028	0.03123	10,000
MNN	347	0.05936	0.06475	0.04109	10,000

SODN에 의해서 식별된 출력은 약간의 오차가 존재하나 다변수 비선형 시스템을 잘 식별함을 보여준다. 표 2에 두 신경망 모델에 대한 학습실험 결과를 요약하였다.

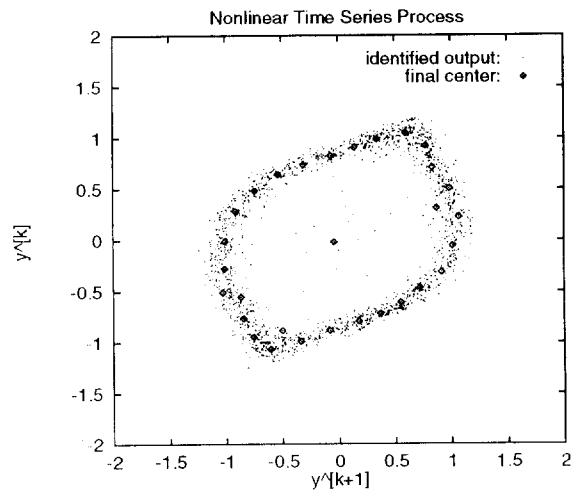
[예제3] 이 예제는 다음 식과 같이 표현되는 비선형 시계열 시스템을 식별하는 것이다.

$$y(k+1) = (0.8 - 0.5 \exp(-y^2(k)))y(k) - (0.3 + 0.9 \exp(-y^2(k)))y(k-1) + 0.1 \sin(\pi y(k)) + \epsilon(k) \quad (23)$$

여기에서 $\epsilon(k)$ 는 표준편차가 0.05인 Gaussian 잡음이다. 시스템에 대한 입력은 $[y(k), y(k-1)]$ 이고 훈련 데이터는 초기



(a)



(b)

그림 7 시계열 시스템에 대한 훈련 데이터 및 식별 결과
Fig. 7 Training data and identification result for time-series system

조건이 $y(0)=0.05, y(1)=0.1$ 인 그림 7(a)에 나타난 1,500개의 데이터를 사용하였다. 초기의 분산 신경망 중심 위치벡터의 분포는 그림 7(a)에 나타낸바와 같이 훈련 데이터값으로 부터 추출하여 초기화하였다. 학습이 진행됨에 따라 분산 신경망의 위치벡터 분포는 입력 데이터의 분포에 의해 최적의 위치로 이동하게되고 최종적인 위치벡터의 분포는 그림 7(b)와 같이 된다. 식별 결과는 그림 7(b)에 제시하였고, 다음 식의 입출력 관계에 의해서 생성된 결과이다.

$$\hat{y}(k+1) = F(\hat{y}(k), \hat{y}(k-1)) \quad (24)$$

테스트에서도 초기조건은 $\hat{y}(0)=0.05, \hat{y}(1)=0.1$ 이다. 결과로부터 명백히 알 수 있는 사실은 SODN은 복잡한 시계열 시스템의 동특성을 매우 근사하게 모델링한다는 것이다.

5. 결 론

신경망은 비선형 동적 시스템을 모델링하고 식별하는데 유용한 일반적인 기법이다. MNN과 같은 일반적인 모델은 가장 널리 사용하는 모델이지만 학습이 어렵고 오차 평면이 매우 복잡하다는 사실은 널리 알려져 있다. 이것은 MNN의 심각한 단점이며 복잡한 시스템의 식별과 같은 문제에서 잠재적인 덫이기도 하다. 그러나, 복잡한 문제를 다수의 단순한 문제로 분할 학습하는 SODN은 입력공간을 분할하여 지역적 학습을 수행하므로 학습 속도가 빠르며, 다른 지역연결 신경망의 단점인 전체적 일반화 능력이 개선된 신경망으로 비선형 동적 시스템 식별에 적합한 모델이다. 이 신경망의 구조는 입력 공간의 분할 역할을 담당하는 분산 신경망과 공간에 대한 학습을 담당하는 학습 신경망으로 구성된다.

이 논문에서는 SODN이 비선형 동적 시스템 식별기로서 적합함을 보이기 위해서 이산 비선형 동적 시스템을 대상으로 시뮬레이션을 수행하였다. 또한, SODN의 학습성능을 입증하기 위해서 MNN과의 비교실험을 수행하였다. 시뮬레이션을 통해서 SODN은 비선형 동적 시스템 식별기로 사용될 수 있음을 보였고, SODN은 MNN에 비해 학습 시간 및 속도가 매우 빠름을 알 수 있었다. 결과로부터 SODN은 간단한 비선형 시스템 및 MIMO 시스템을 매우 잘 식별할 수 있음이 입증되었다. 특히 복잡한 시계열 시스템의 동특성도 실제 시스템과 근사하게 모델링함을 알 수 있었다. 이상의 SODN에 의한 비선형 시스템 식별 능력을 기초로 SODN은 복잡한 비선형 시스템의 제어기 구현에 유용하게 사용할 수 있을 것으로 기대된다.

본 연구는 1994년도 학술진흥재단의 지원에 의해 수행된 결과의 일부입니다.

참 고 문 헌

[1] W. R. Miller, R. S. Sutton, and P. J. Werbos, *Neural Networks for Control*, Cambridge, MA, MIT Press, 1990.

[2] G. Cybenko, "Approximation by superposition of a sigmoidal function," *Mathematics of Control, Signal, and Systems*, vol. 2, pp. 303-314, 1989.

[3] K. Hornik, M. Stinchcombe, and H. White "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359-366, 1989.

[4] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4-27, 1991.

[5] S.-Z. Qin, H.-T. Su, and T. J. McAvoy, "Comparison of four neural net learning methods for dynamic system identification," *IEEE Transactions on Neural Networks*, vol. 3, no. 1, pp. 122-130, 1992.

[6] P. S. Sastry, G. Santharam, and K. P. Unnikrishnan, "Memory neuron networks for identification and control of dynamical systems," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 306-319, 1994.

[7] S. Chen and S. A. Billings, "Neural networks for non-

linear dynamic system modelling and identification," *International Journal of Control*, vol. 56, no. 2, pp. 319-346, 1992.

[8] L.-C. Chen, W.-C. Chen, F.-Y. Chang, "Hybrid learning algorithm for Gaussian potential function networks," *IEEE Proceedings-D*, vol. 140, no. 6, 1993.

[9] R. A. Jacobs and M. I. Jordan, "Learning piecewise control strategies in a modular neural network architecture," *IEEE Transaction System, man, and Cybernetics*, vol. 23, no. 2, pp. 337-345, 1993.

[10] 김형석, 최종수, 김성중, "시스템 정밀 모델링을 위한 자율 분산 신경망," *전자공학회논문지*, 제31권, B편, 제11호, pp. 151-162, 1994.

[11] 김형석, 최종수, 김종만, 김성중, "용이한 신경망 학습을 위한 자율적 분산 학습망에 관한 연구," *전기학회논문지* 제43권 10호, pp.1762-1770, 1994.

[12] H. Kim, J. S. Choi, J. M. Kim, S. J. Kim, and C.-S. Lin, "The Self-Organized Distributed Networks: An efficient learning architecture for highly nonlinear function approximation," *International Conf. on Neural Information Processing*, vol.1, pp.248-254, Oct., Seoul, 1994.

[13] H. Kim, J. S. Choi, and C.-S. Lin, "Self-Organized Distributed Networks for learning highly nonlinear mapping," *Proceedings of artificial Neural Networks in Engineering(ANNIE'94)*, pp.109-114, Nov., St. Louis, U.S.A., 1994.

저 자 소 개



최 종 수 (崔鍾洙)

1969년 1월 18일생. 1990년 전북대 공대 전기공학과 졸업. 1992년 동 대학원 전기공학과 졸업(석사). 1994년 동 대학원 전기공학과 박사과정 수료. 현재 포스콘 기술 연구소 근무중



김 형 석 (金炯奭)

1956년 1월 21일생. 1980년 한양대 공대 전자공학과 졸업. 1982년 전북대 대학원 전기공학과 졸업(석사). 1992년 미국 미조리대학교 전기 및 컴퓨터 공학과 졸업(공학박). 현재 전북대학교 제어계측공학과 조교수



김 성 중 (金聖中)

1942년 3월 7일생. 1965년 전북대 공대 전기공학과 졸업. 1985년 전남대 전기공학과 졸업(공학박). 1981년~1982년 미국 Mis-souri대학 연수. 현재 전북대 공대 제어계측공학과 교수