

論文98-35C-11-10

RAN을 위한 개선된 학습 방법

(An Improved Learning Approach for the Resource-Allocating Network(RAN))

崔鍾洙*, 權五臣**, 金炯奭***

(Jong-Soo Choi, Oh-Shin Kwon, and Hyong-Suk Kim)

요 약

정적 시스템 모델링을 위해 RBF 신경회로망의 은닉 유닛을 자동으로 생성하는 ERAN을 제안한다. ERAN은 관측 데이터의 신규성을 기반으로 새로운 은닉 유닛을 할당하는 RAN의 성능을 개선한 것이다. ERAN의 학습 과정은 새로운 은닉 유닛의 생성과 네트워크 파라미터 학습을 포함한다. 네트워크는 초기에 0개의 은닉 유닛으로 시작하여 세 가지의 은닉 유닛 생성 판별기준을 만족할 경우에만 새로운 은닉 유닛을 생성시킨다. 네트워크의 파라미터는 LMS 알고리즘을 이용하여 조정한다. 제안한 ERAN의 성능은 순차 학습 및 랜덤 학습을 갖는 비선형 정적 시스템 모델링 문제에 대하여 RAN의 결과와 성능을 비교한다. 두 실험에 대하여 ERAN은 RAN 보다 적은 은닉 유닛을 가지고 정확성이 더 우수한 RBF 신경회로망을 구현할 수 있음을 보인다.

Abstract

The enhanced resource-allocating network(ERAN) that adaptively generates hidden units of radial basis function(RBF) network for systems modeling has been proposed. The ERAN is an improved version of the resource-allocating network(RAN) that allocates new hidden units based on the novelty of observation data. The learning process of the ERAN involves allocation of new hidden units and adjusting the network parameters. The network starts with no hidden units. As observation data are received, the network adds a hidden units only if the three network growth criteria are satisfied. The network parameters are adjusted by the LMS algorithm. The performance of the ERAN is compared with the RAN for nonlinear static systems modeling problem with sequential and random learning. For two simulations, the ERAN has been shown to realize RBF networks with better accuracy with fewer hidden units.

* 正會員, 浦港産業科學研究院 自動化研究部門

(Automation Research Division, RIST)

** 正會員, 群山大學校 制御計測工學科

(Dept. of Control & Instrumentation Eng., Kunsan National Univ.)

*** 正會員, 全北大學校 電氣電子制御工學部, 메카트로닉스研究中心

(School of Electrical Eng. and Mechatronics Research Center, Chonbuk National Univ.)

接受日字:1998年9月7日, 수정완료일:1998年11月3日

I. 서론

RBF(radial basis function) 신경회로망은 주요한 신경회로망 모델 중의 하나로서 매핑을 위해 기저함수라 하는 국부 감응함수를 이용한다. RBF 신경회로망은 일반화 능력, 계산 능력 및 생물학적 타당성에서 기존 다층 신경회로망의 개선된 모델로 생각할 수 있고, 함수 근사화, 패턴 분류, 예측 및 시스템 모델링 등 다양한 분야에 적용되어 그 유용성이 입증되고 있다^[1-5].

RBF 신경회로망을 구현하기 위해서는 은닉 유닛

의 수와 각 은닉 유닛에 대한 중심 및 폭을 결정해야 한다. 기존의 접근법에 있어서 RBF 유닛은 통상 Gaussian 함수로 선정되고 은닉 유닛 수와 중심 및 폭은 입력 데이터를 기반으로 사전에 전처리과정을 거쳐 결정된다. Sutanto 등^[6]은 RBF의 중심을 결정하기 위해 주어진 데이터 공간에 클러스터 중심을 반복적 방법으로 할당하는 mean-tracking 클러스터링 알고리즘을 제안하였고, Kuncheva^[7]는 유전 알고리즘을 이용하여 RBF 유닛의 프로토타입을 초기화하는 방법을 제안하였다. 한편, Platt^[8]은 훈련 데이터의 신규성을 기반으로 새로운 은닉 유닛을 할당하여 신경회로망을 구성하는 RAN(resource-allocating network)을 제안하였다. RAN은 근본적으로 Gaussian 유닛을 사용하는 RBF 신경회로망으로서 다음과 같은 장점을 갖는다. 첫째, 동일한 정확도를 가질 지라도 RAN은 Moody 등^[11]의 네트워크보다 더 작은 파라미터를 필요로 한다. 둘째, Moody^[9]의 hashing 접근법과 같이 RAN은 보간 되는 함수의 복잡성을 반영하기 위해 은닉 유닛의 수를 자동으로 조절한다. 마지막으로 RAN은 Poggio 등^[10]의 연구와 같이 오차를 기반으로 Gaussian 유닛의 중심을 조절한다.

그러나, Platt의 RAN은 데이터의 신규성을 관측하여 은닉 유닛을 할당하기 때문에 훈련 데이터에 대하여 너무 과도한 은닉 유닛을 생성하는 문제점이 있고 Gaussian 유닛의 폭을 학습하지 않는다. 이러한 단점을 해결하기 위해서 Kadirkamanathan 등^[11]은 RAN의 파라미터를 조정하기 위해 LMS 알고리즘 대신 EKF(extended Kalman filter)를 적용하여 순차 학습(sequential learning)에 대한 준최적의 네트워크를 구성하는 방법을 제안하였다. 이 방법은 EKF의 빠른 수렴 속도를 이용하여 은닉 유닛의 생성을 조절하는 것으로 생성을 판별하는 기준은 RAN과 동일하다.

이 논문에서는 RAN의 성능을 개선하기 위해서 입력공간에서 은닉 유닛이 동작하는 영역을 고려하여 생성을 판별하고 RBF 유닛의 폭을 학습에 의해 조절하는 ERAN(Enhanced RAN)을 제안한다. 제안하는 알고리즘은 과도한 은닉 유닛의 생성을 억제하기 위해서 이미 존재하는 은닉 유닛의 활성화 정도(activation degree)를 관측하여 새로운 은닉 유닛 생성을 결정하는 판별기준 제공한다. 제안한 ERAN의 성능을 평가하기 위해서 비선형 정적 시스템 모델링

문제에 적용한다.

II. Resource-Allocating Network(RAN)

Platt이 제안한 RAN은 크기가 고정된 네트워크를 갖는 학습에 있어서 NP-complete 문제를 극복하기 위한 수단으로 개발되었다^[8]. RAN은 입력 패턴에 대한 출력 응답이 은닉 유닛 응답의 선형 결합인 단일 은닉층 신경회로망으로서 출력은 다음과 같이 주어진다.

$$f(\mathbf{x}) = a_0 + \sum_{k=1}^K a_k \phi_k(\mathbf{x}) \quad (1)$$

여기에서 $\phi_k(\mathbf{x})$ 는 입력 \mathbf{x} 에 대한 은닉 유닛의 응답이다. a_k 는 출력 유닛과 은닉 유닛 사이의 연결강도이고, a_0 는 바이어스 항이다. RAN의 은닉 유닛 응답은 다음과 같이 주어진다.

$$\phi_k(\mathbf{x}) = \exp\left\{-\frac{1}{\sigma_k^2} \|\mathbf{x} - \mathbf{u}_k\|^2\right\} \quad (2)$$

여기에서 \mathbf{u}_k 와 σ_k 는 각각 Gaussian의 중심과 폭이다. 식 (1)은 바이어스 항 a_0 를 제외하면 근본적으로 Gaussian을 활성화 함수로 갖는 RBF 신경회로망이다. Gaussian의 중심 \mathbf{u}_k 는 저장된 패턴으로 생각할 수 있고, 연결강도 a_k 는 특정 출력에 대한 각 은닉 유닛의 공헌도를 정의하는 요소이다.

RAN은 초기에 은닉 유닛을 갖지 않고 훈련을 시작한다. 첫번째 관측 데이터 (\mathbf{x}_0, y_0) 는 바이어스 항 $a_0 = y_0$ 를 초기화하는데 이용된다. 관측 데이터들이 입력됨에 따라 네트워크는 새로운 은닉 유닛을 추가함으로써 자동으로 생성된다. 관측 데이터 (\mathbf{x}_n, y_n) 에 대하여 다음의 두 판별규칙을 만족하게되면 새로운 은닉 유닛이 추가된다.

$$d_{\min} = \|\mathbf{x}_n - \mathbf{u}_{nr}\| > \epsilon_n \quad (3)$$

$$E_n = y_n - f(\mathbf{x}_n) > E_{\min} \quad (4)$$

여기에서 \mathbf{u}_{nr} 은 입력공간에서 \mathbf{x}_n 에 가장 가까운 은닉 유닛의 중심이다. ϵ_n 과 E_{\min} 은 적절히 선택되어지는 문턱값(threshold)이다. 첫번째 판별기준은 입력과 저장된 패턴(은닉 유닛) 사이의 거리에 기반을 두고 있고, 두번째 판별기준은 목표치와 네트워크 출력

치 사이의 오차에 기반을 두고 있다. E_{\min} 은 네트워크 출력에 대한 원하는 정확도를 나타내기 위해 선정된다. 그리고 거리 ϵ_n 은 입력공간의 분해능 스케일을 나타내는 것으로서 관측한 패턴의 수가 증가함에 따라 점점 줄어들게 된다. 즉, 초기에 $\epsilon_n = \epsilon_{\max}$ 로 시작하여 다음 식과 같이 감쇄한다.

$$\epsilon_n = \max\{\epsilon_{\max} \gamma^n, \epsilon_{\min}\} \quad (5)$$

여기에서 $\gamma(0 < \gamma < 1)$ 는 감쇄상수이다. 학습이 진행됨에 따라 ϵ_n 값은 ϵ_{\min} 에 도달할 때까지 감쇄한다.

새로운 은닉 유닛이 네트워크에 추가되면 이 유닛과 결합되는 파라미터들은 다음과 같이 할당된다.

$$a_{K+1} = E_n \quad (6)$$

$$\mathbf{u}_{K+1} = \mathbf{x}_n \quad (7)$$

$$\sigma_{K+1} = x \|\mathbf{x}_n - \mathbf{u}_m\| \quad (8)$$

여기에서 x 는 입력공간에서 은닉 유닛 응답의 중첩을 결정하는 요소이다.

관측 데이터 (\mathbf{x}_n, y_n) 가 식 (3) 및 (4)의 은닉 유닛 생성을 위한 판별기준을 만족하지 않으면, 네트워크 파라미터 벡터 $\mathbf{w} = [a_0, \dots, a_K, \mathbf{u}_1^T, \dots, \mathbf{u}_K^T]^T$ 는 LMS 알고리즘을 이용하여 다음과 같이 갱신된다.

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta E_n \mathbf{a}_n \quad (9)$$

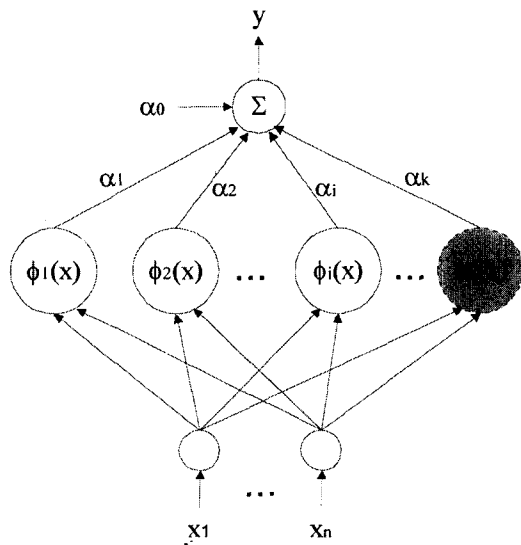


그림 1. RAN의 구조
Fig. 1. The structure of the RAN.

여기에서 η 는 학습률이고 $\mathbf{a}_n = \nabla_w f(\mathbf{x}_n)$ 은 함수 $f(\mathbf{x})$ 의 그레디언트이다. 그림 1에 RAN의 기본 구조를 나타내었다. 그림 1에서 k 번째 은닉 유닛 $\phi_k(x)$ 는 학습이 진행됨에 따라 새로이 생성되는 은닉 유닛을 의미한다.

III. 제안한 Enhanced Resource-Allocating Network(ERAN)

기존 RAN의 성능을 개선하기 위하여 관측 데이터로부터 은닉 유닛을 생성시키는 새로운 판별기준과 RBF 유닛의 폭 조절기능을 포함한 ERAN을 제안한다. 기존의 RAN은 기본적으로 관측된 데이터의 신구성을 판단하여 은닉 유닛을 할당하고 있다. 식 (3)과 (4)의 판별기준에 따라 관측된 훈련 데이터 입력과 가장 가까운 은닉 유닛 중심과의 거리가 ϵ_n 보다 크고, 훈련 데이터 출력과 네트워크 출력 사이의 오차가 E_{\min} 보다 크면 새로운 유닛이 생성되게 된다. 이 알고리즘은 입력공간 상에서 존재하는 은닉 유닛들이 동작하고 있는 영역을 고려하지 않고 있기 때문에 훈련 데이터에 대하여 과도한 은닉 유닛이 생성되는 문제를 초래할 수 있다. 그리고 RAN의 은닉 유닛 생성의 판단은 두 개의 문턱값 ϵ_n 와 E_{\min} 에 전적으로 의존한다.

또한, RAN의 학습 알고리즘은 은닉 유닛의 중심 (\mathbf{u}_k)과 출력 연결강도(a_k)를 학습대상으로 하고 은닉 유닛의 폭(σ_k)은 학습에 포함하지 않는다. 은닉 유닛의 폭이 학습되지 않고 고정되어 있으면 훈련 입력 패턴이 순차적으로 관측되는 순차 학습(sequential learning)일 경우에는 문제가 되지 않지만 훈련 입력 패턴이 랜덤하게 관측되는 랜덤 학습(random learning)에서는 coarse한 입력공간의 분해를 초래하여 학습을 어렵게 만들 수 있고 모델링 성능을 저하시키는 문제를 유발시킬 수 있다. 즉, 랜덤 학습에서 현재의 관측 데이터가 이전에 관측된 데이터와 거리 상으로 멀리 떨어져 있고 두 데이터 사이에 다른 관측 데이터가 존재하지 않는다고 가정하면, 이 때 새로 생성되는 은닉 유닛의 폭은 식 (8)과 같이 현재 관측된 데이터와 가장 가까운 거리에 있는 은닉 유닛 사이의 유클리드 거리에 의해서 결정된다. 이 경우 새로 생성되는 은닉 유닛은 상대적으로 큰 폭을 갖

게 되어 입력공간을 coarse하게 분해한다. RAN에서 은닉 유니트의 폭은 학습되지 않으므로 생성시 결정된 폭은 학습이 종료될 때까지 변화하지 않고 그대로 유지된다.

Kardirkamanathan 등^[11]은 RAN의 모델링 능력을 개선하기 위한 연구를 수행하였는데, 이 들의 연구는 은닉 유니트 생성 알고리즘은 기존의 RAN 방법을 그대로 사용하고 있고 단지 파라미터 학습을 위해 LMS 학습 규칙을 EKF 알고리즘으로 대체한 것이다. 다시 말해서, EKF 알고리즘이 LMS보다 학습 수렴속도가 빠른 점을 이용한 것으로 최종적으로 생성되는 은닉 유니트의 수가 기존 RAN보다 적다.

1. 은닉 유니트 생성 알고리즘

제안하는 ERAN은 RAN의 성능을 개선시키기 위해서 입력공간 상에서 이미 존재하는 은닉 유니트의 동작영역을 고려하여 은닉 유니트를 생성하는 새로운 판별기준을 추가하여 과도한 은닉 유니트가 할당되는 것을 방지한다. 추가되는 은닉 유니트 판별기준은 이미 존재하는 은닉 유니트의 활성화 정도(activation degree)를 계산한 후 가장 큰 활성화값과 미리 정의된 문턱값과의 비교를 통해서 새로운 은닉 유니트를 생성시키는 방법이다. 활성화 정도에 대한 문턱값을 δ 라 정의하자. 은닉 유니트의 수가 K 개라면 RAN에서의 두 판별기준과 제안하는 아래의 판별기준을 모두 만족해야 은닉 유니트가 생성된다.

$$\phi_{\max} = \max\{\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \dots, \phi_K(\mathbf{x})\} < \delta \quad (10)$$

여기에서 $\phi_{\max}(\mathbf{x})$ 는 은닉 유니트 중 활성화 정도가 가장 큰 은닉 유니트의 활성화값이고, $\phi_i(\mathbf{x})$ ($i=1, 2, \dots, K$)는 식 (2)와 같이 표현되는 각 RBF 은닉 유니트 함수이다. 제안한 판별기준은 가장 큰 은닉 유니트의 활성화값 ϕ_{\max} 가 문턱값 δ 와 비교되는데, 만일 $\phi_{\max} \geq \delta$ 이면 입력공간 상에서 현재의 입력 \mathbf{x} 와 충분히 가까운 은닉 유니트가 존재함을 의미하므로 은닉 유니트는 생성되지 않는다. 제안한 판별기준은 퍼지 추론 시스템의 설계에서 각 입력공간의 멤버십 함수는 동작영역에서 주어진 입력에 대하여 멤버십 함수 $\mu_A(\mathbf{x})$ 가 $\mu_A(\mathbf{x}) \geq \epsilon$ ($\epsilon=0.5$)인 퍼지집합 A 가 존재해야 하는 ϵ -completeness^[12] 조건과 유사한 개념이다. 제안한 판별기준은 기존 RAN의 두 판별기준과

병렬로 동작하여 은닉 유니트 생성을 결정하는데 참여하게 된다. 따라서, 관측된 데이터에 대하여 과도한 은닉 유니트가 할당되는 것을 방지할 수 있다.

2. ERAN의 파라미터 학습

관측된 데이터에 대하여 새로운 은닉 유니트 생성을 위한 판별기준을 만족하지 못하면 네트워크는 파라미터 벡터를 학습하게 된다. ERAN의 학습 알고리즘은 다음과 같이 정의되는 오차함수의 최소화에 기반을 둔다.

$$E = \frac{1}{2} [y^d(k) - y(k)]^2 \quad (11)$$

여기에서 y^d 는 목표출력이다. 식 (11)에 정의된 함수는 전체 훈련 데이터의 오차를 종합한 오차가 아니고 개별 훈련 데이터의 오차이다. 이러한 함수 정의는 알고리즘의 실시간 구현에 적합하다. 이 논문에서 고려하는 학습 알고리즘은 LMS 규칙을 이용하는 감독학습으로 학습의 목적은 Gaussian 함수형태를 갖는 은닉 유니트의 중심 벡터 \mathbf{u} 와 폭 벡터 σ , 그리고 은닉 유니트의 출력과 출력단 사이의 연결강도 벡터 α 를 결정하는 것이다. 관측 데이터 (\mathbf{x}_n, y_n)가 식 (3), (4) 및 (10)의 은닉 유니트 생성을 위한 판별기준을 만족하지 않으면, 네트워크 파라미터 벡터 $\mathbf{w} = [\mathbf{u}_1^T, \dots, \mathbf{u}_K^T, \sigma_1^T, \dots, \sigma_K^T, \alpha_0, \dots, \alpha_K]^T$ 는 LMS 알고리즘을 이용하여 다음과 같이 갱신된다.

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta \Delta \mathbf{w}(t) \quad (12)$$

여기에서 η 는 학습률이다. 은닉 유니트의 중심 및 폭 벡터의 변화량은 식 (11)에 정의된 오차함수를 미분하고 연쇄법칙을 적용하여 구할 수 있다. 은닉 유니트의 수가 K 개이고 i 번째 입력시 j 번째 은닉 유니트의 중심, 폭 및 j 번째 은닉 유니트의 연결강도를 각각 u_j , σ_j 및 α_j 라 하면, 다음과 같은 갱신 규칙을 얻을 수 있다.

$$\Delta u_j = -\frac{\partial E}{\partial u_j} = (y^d - y) \frac{(\alpha_j - y)}{\sum_{j=1}^K \phi_j(x)} \phi_j(x) \frac{(x_i - u_j)}{\sigma_j^2}, \quad (13)$$

$$\Delta \sigma_j = -\frac{\partial E}{\partial \sigma_j} = (y^d - y) \frac{(\alpha_j - y)}{\sum_{j=1}^K \phi_j(x)} \phi_j(x) \frac{(x_i - u_j)^2}{\sigma_j^3}, \quad (14)$$

$$\Delta \alpha_j = -\frac{\partial E}{\partial \alpha_j} = (y^d - y) \frac{\phi_j(x)}{\sum_{j=1}^K \phi_j(x)}. \quad (15)$$

네트워크 파라미터 벡터 w 는 식 (13)~(15)의 갱신규칙을 포함하며 식 (12)의 $\Delta w(t)$ 는 다음과 같이 표현된다.

$$\Delta w(t) = [\Delta u_i, \Delta \sigma_i, \Delta \alpha_i] \quad (16)$$

ERAN의 전체 알고리즘을 단계별로 요약하면 다음과 같다.

[ERAN 알고리즘]

Step 1: 학습 파라미터 초기화

학습에 사용되는 문턱값 및 파라미터 $E_{min}, \epsilon_{max}, \epsilon_{min}, \delta, \gamma, \alpha$ 와 학습률 η 를 초기화하고, $\epsilon_n = \epsilon_{max}$ 로 초기화.

Step 2: 관련 연산 수행

1) 첫 번째 훈련 패턴 (x_0, y_0)가 인가될 경우 초기 바이어스 파라미터 a_0 는 y_0 로 초기화.

2) 훈련 패턴 (x_n, y_n)에 대하여 은닉 유닛의 출력(식(2)) 및 네트워크 출력(식(1))을 계산하고, 식(5)를 이용하여 ϵ_n 을 계산.

3) 은닉 유닛 생성 판별기준인 식(3), (4) 및 (10)을 계산.

Step 3: 은닉 유닛 생성

1) Step 2에서 계산된 결과를 이용하여 식 (3), (4) 및 (10)에 의해 은닉 유닛 생성을 판단.

2) 생성 기준을 만족하면 새로운 은닉 유닛을 생성하여 식 (6), (7) 및 (8)을 이용하여 파라미터를 초기화한 후 Step 2로 되돌아가고, 그렇지 않으면 Step 4로 진행.

Step 4: 네트워크 파라미터 학습

1) 식 (13)~(15)의 학습 규칙을 이용하여 파라미터를 학습.

2) 모든 훈련 패턴이 입력될 때까지 Step 2~4를 반복 수행.

Step 5: 성능 평가

성능을 평가하여 네트워크의 출력이 목표 성능지수에 도달하거나 설정한 반복학습 회수를 모두 수행하면 학습을 종료하고, 그렇지 않으면 Step 2로 되돌아감.

IV. 시뮬레이션

이 절에서는 제한한 ERAN의 성능을 평가하기 위해 두 개의 비선형 정적 시스템에 대한 모델링 실험

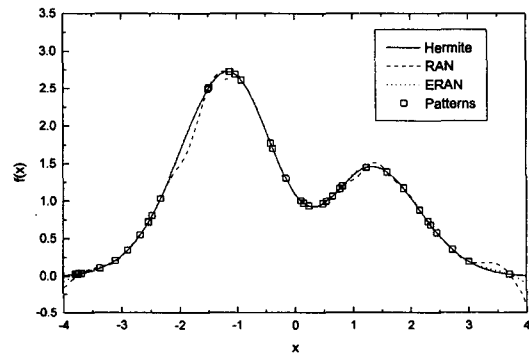
결과를 기술한다.

1. 1차원 비선형 시스템 모델링

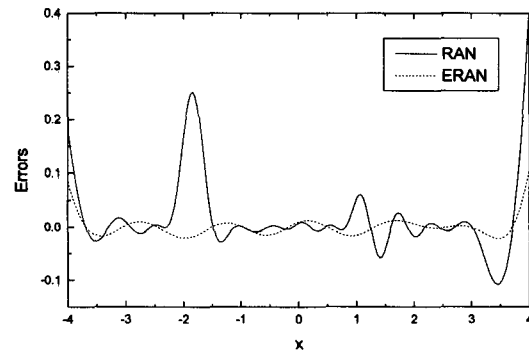
첫번째 비선형 시스템 모델링 실험의 예제로 식 (17)로 주어지는 Hermite polynomial^[11]을 사용한다.

$$f(x) = 1.1(1 - x + 2x^2) \exp\left(-\frac{1}{2}x^2\right). \quad (17)$$

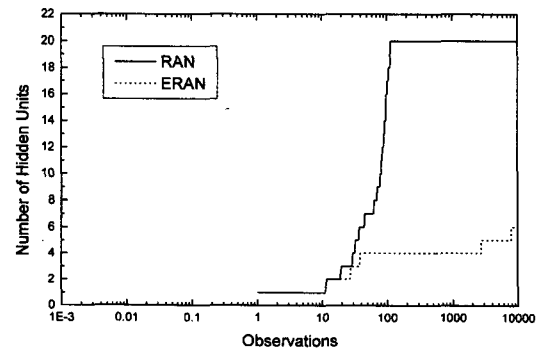
훈련을 위한 데이터는 $[-4, 4]$ 범위에서 40개의 데이터를 랜덤하게 추출하였다(그림 2(a)에서 \square 으로 표시됨).



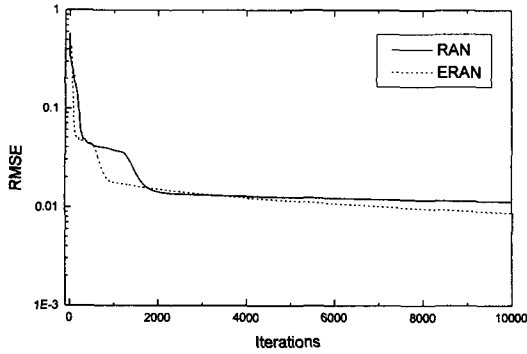
(a) 훈련 패턴 및 모델링 결과



(b) 모델링 오차



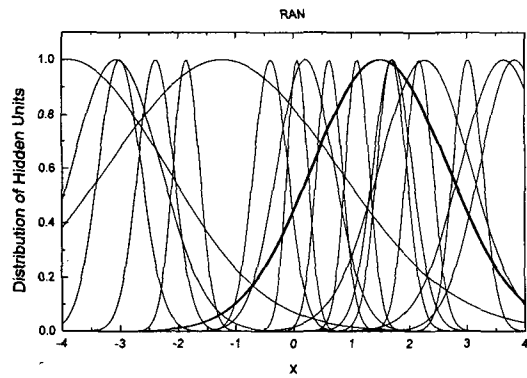
(c) 은닉 RBF 유닛의 생성 과정



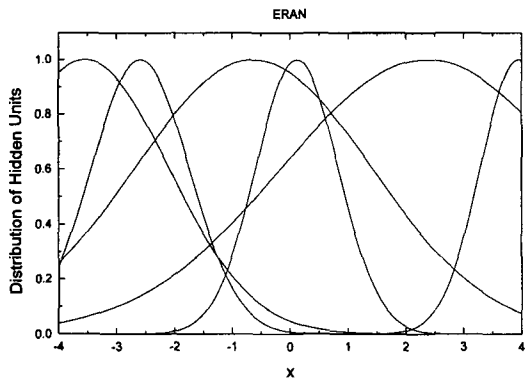
(d) RMS 오차

그림 2. 순차 패턴에 대한 모델링 결과
Fig. 2. The performance of RAN and ERAN with sequential patterns.

이 실험에서 사용된 문턱값 및 파라미터들은 $E_{min}=0.02$, $\epsilon_{max}=2.0$, $\epsilon_{min}=0.2$, $\delta=0.5$, $\gamma=0.977$, $\alpha=0.87$ 와 학습률 $\eta=0.05$ 로써 δ 이외에는 Kadiramanathan등^[11]이 사용한 값과 동일하다.



(a) RAN

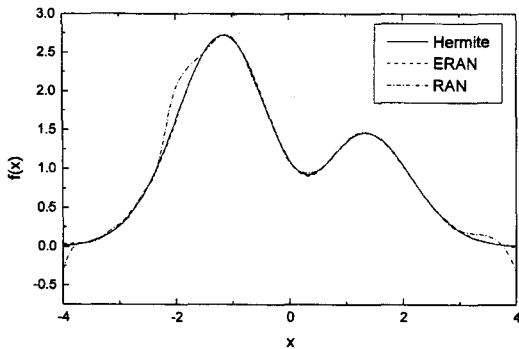


(b) ERAN

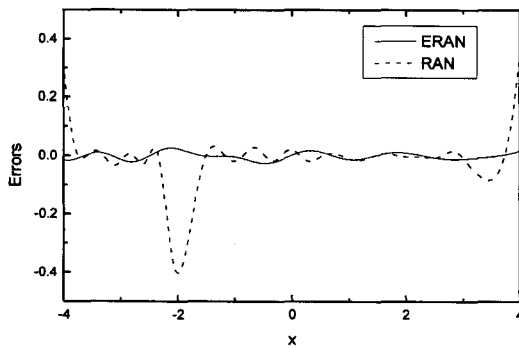
그림 3. 순차 패턴에 대한 은닉 유닛의 분포
Fig. 3. The distribution of hidden units for sequential patterns.

RAN과 ERAN의 결과를 그림 2~5와 표 1에 보였다. 그림 2와 3은 40개의 훈련 데이터를 순차적으로 학습시킨 결과이다. 그림 2(a)는 RAN과 ERAN의 모델링 결과 및 훈련 패턴을 나타내고 있고, 그림 2(b)는 모델링 오차를, 그림 2(c)는 은닉 RBF 유닛의 생성 과정을, 그리고 그림 2(d)는 학습 RMS 오차 곡선을 각각 보여준다. 또한 그림 3은 순차 학습에 대한 RAN과 ERAN의 은닉 유닛 분포 및 폭을 나타낸다. 그림 2와 표 1의 결과로부터 제안한 ERAN은 6개의 은닉 유닛을 가지고 20개의 은닉 유닛을 갖는 RAN 보다 더 적은 학습 및 테스트 모델링 오차를 갖는다는 것을 알 수 있다. 또한 그림 3에 나타난 은닉 유닛의 분포를 고려해 볼 때, 기존의 RAN은 훈련 데이터의 신규성 및 학습 오차만을 고려하여 은닉 유닛을 생성하기 때문에 입력공간에서 과도한 은닉 유닛이 생성되는데 비하여 제안한 ERAN은 훈련 데이터의 신규성과 오차 뿐만 아니라 은닉 유닛의 활성화 정도를 고려하여 은닉 유닛의 생성을 결정하기 때문에 RAN 보다 더 적은 은닉 유닛을 갖는 네트워크를 자동으로 생성함을 확인할 수 있다.

훈련 패턴이 랜덤하게 주어질 경우에 대한 실험도 수행하였다. 그림 4와 5는 랜덤 학습에 대한 결과로써 그림 4는 모델링 출력과 오차를 나타내고 그림 5는 두 모델의 은닉 유닛 분포와 폭을 나타낸다. 그림 4와 표 1에서 5개의 은닉 유닛이 생성된 ERAN은 20개의 은닉 유닛을 갖는 RAN 보다 더 적은 테스트 오차로 모델링을 수행함을 알 수 있다. 특히, ERAN은 훈련 패턴이 없는 영역에서도 모델링 성능이 우수한 반면, RAN은 그 영역에서 모델링 오차가 현저히 증가함을 알 수 있다. 즉, 많은 은닉 유닛을 갖고 있음에도 불구하고 보간 능력이 매우 저조하다. 또한 표 1에서 ERAN의 테스트 RMS 오차는 순차 학습의 결과보다 랜덤 학습의 경우가 더 적은 유닛을 갖고 있음에도 불구하고 더 적음을 알 수 있다. 그러나 RAN의 결과는 순차 학습과 랜덤 학습인 경우의 은닉 유닛 수가 동일함에도 불구하고 오차가 더 커짐을 알 수 있다. 이러한 결과는 3절에서 언급한 랜덤 학습시에 발생할 수 있는 coarse한 입력공간의 분해와도 연관이 있을 것으로 생각된다. 결과적으로 제안한 ERAN은 RAN에 비해 순차 및 랜덤 학습에 대하여 매우 적은 은닉 유닛을 가지고도 더 우수한 시스템 모델링 성능을 보여 주었다.



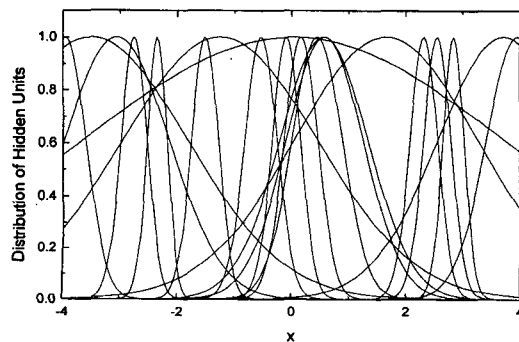
(a) 모델링 출력



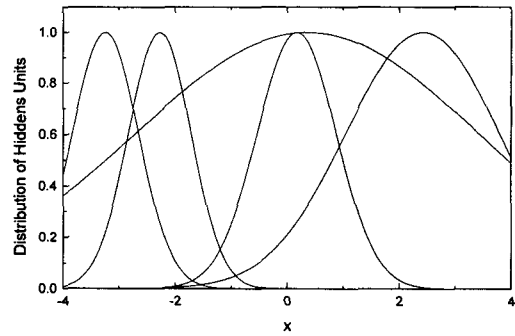
(b) 모델링 오차

그림 4. 랜덤 패턴에 대한 모델링 결과
Fig. 4. The performance of RAN and ERAN with random patterns.

동일한 실험에 대하여 Kadiramanathan 등^[11]이 제안한 RAN-EKF의 결과는 순차 학습에 대하여 13 개의 은닉 유닛이 생성되었고, 훈련 RMS 오차는 0.01보다 큰 값으로 수렴한 반면 ERAN의 결과는 6 개의 은닉 유닛이 생성되었고 훈련 RMS 오차는 0.0087이었다. 두 모델의 결과를 비교해 볼 때 제안한 ERAN은 RAN-EKF보다 모델링 성능이 우수함을 확인하였다.



(a) RAN



(b) ERAN

그림 5. 불규칙 패턴에 대한 은닉 유닛의 분포
Fig. 5. The distribution of hidden units for random patterns.

표 1. 1차원 시스템 모델링 성능 비교

Table 1. The Performance comparison for 1-D system modeling.

학습 패턴	모델	훈련 RMS 오차	테스트 RMS 오차	은닉 유닛 수
순차 학습	RAN	0.0114	0.0750	20
	ERAN	0.0087	0.0184	6
랜덤 학습	RAN	0.0108	0.0985	20
	ERAN	0.0109	0.0126	5

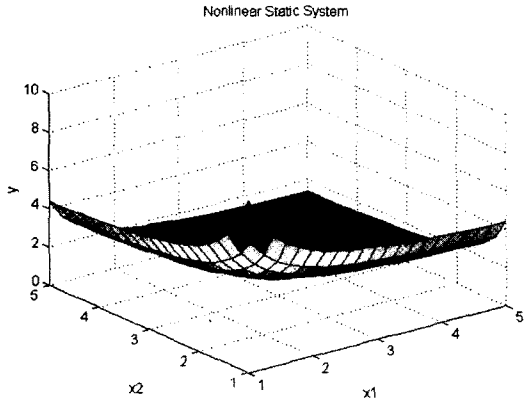
2. 2차원 비선형 정적 시스템 모델링

실험 대상 시스템은 Sugeno 등^[13]이 모델링의 성능을 평가하기 위해 사용한 비선형 정적 시스템으로서 다음 식과 같이 주어진다.

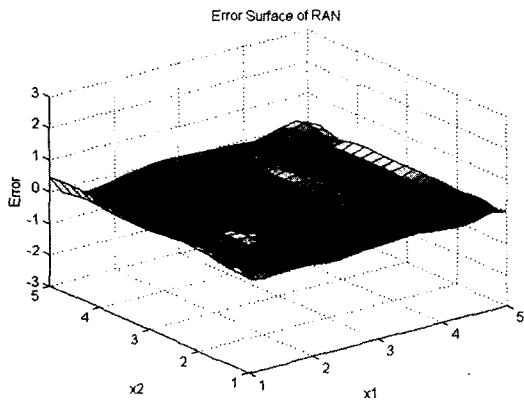
$$y = f(x_1, x_2) = (1 + x_1^{-2} + x_2^{-1.5})^2, \quad 1 \leq x_1, x_2 \leq 5 \quad (18)$$

이 시스템으로부터 x_1 및 x_2 에 대하여 0.5 간격으로 81개의 훈련 데이터를 수집하여 순차 학습에 대한 모델링 실험을 수행하였다. 이 실험에서 사용된 파라미터들은 $E_{\min} = 0.05$, $\epsilon_{\max} = 2.0$, $\epsilon_{\min} = 0.2$, $\delta = 0.6$, $\gamma = 0.977$, $\alpha = 0.87$ 와 학습률 $\eta = 0.02$ 이다. 그림 6은 두 입력에 대하여 각각 0.2간격으로 401개의 포인트에 대하여 테스트한 결과로써 그림 6(a)는 식 (18)의 2차원 비선형 정적 시스템에 대한 입출력 평면을 나타내고, 그림 6(b)와 6(c)는 각각 RAN과 ERAN에 의한 모델링 오차평면을 나타낸다. 또한 표 2는 모델링 실험에 대한 결과이다. 표 2로부터 모델링을 위해 생성된 RAN의 은닉 유닛 수는 35개로 ERAN의 12개에 비하여 약 3배 정도 많은 것에 비해 훈련 및 테스트 RMS 오차에는 큰 차이가 없었다. 그

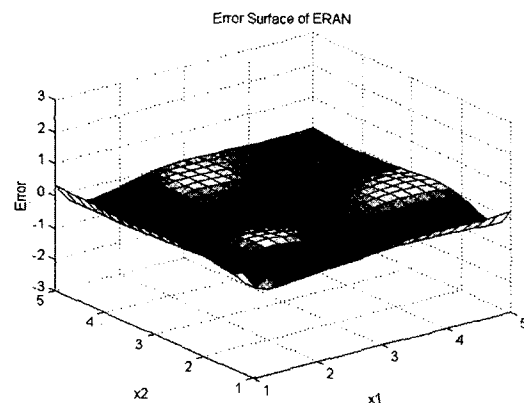
러나, 그림 6의 모델링 오차평면을 비교해 보면, 그림 6(c)의 ERAN은 그림 6(b)의 RAN에 비해 오차평면의 굴곡이 완만하게 나타남을 알 수 있다.



(a) 2차원 비선형 정적 시스템



(b) RAN의 모델링 오차평면



(c) ERAN의 모델링 오차평면

그림 6. 2차원 비선형 정적 시스템에 대한 실험 결과
Fig. 6. The results for 2-D nonlinear static system.

랜덤 학습에 대한 ERAN의 모델링 실험은 Sugeno

등^[13]이 랜덤하게 수집한 50개의 데이터를 훈련 패턴으로 사용하였고, 그 결과를 Sugeno 등이 제안한 퍼지 모델과 비교하였다. 비교를 위한 성능평가 지수 PI(performance index)는 다음과 같이 표현되는 MSE(mean square error)이다.

$$PI = \sum_{i=1}^m (y_i - \hat{y}_i)^2 / m \quad (19)$$

여기에서 m 은 패턴의 수이고, y_i 와 \hat{y}_i 는 각각 시스템 출력과 모델링 출력을 의미한다. ERAN을 이용하여 시스템을 모델링한 결과는 표 3에 나타낸바와 같이 최종적으로 6개의 은닉 유니트로 구성되었다. 실험에 사용된 파라미터들은 학습률 $\eta=0.04$ 를 제외하고 앞의 순차 학습에 사용한 값과 동일하다. 모델링 실험 결과로부터 ERAN의 성능지수 PI는 0.065로 Sugeno 등이 제시한 결과인 0.079에 비해 모델링 성능이 우수함을 알 수 있었다. Sugeno 등이 제안한 퍼지 모델에서 퍼지 규칙의 수는 역시 6개였으나 Sugeno 등의 퍼지 모델은 퍼지 규칙을 구성하기 위해서 삼각형과 사다리꼴 형태의 멤버십 함수를 혼용하고 있기 때문에 한 개의 멤버십 함수를 정의하기 위해서 2~4개의 파라미터를 필요로 한다. 그러므로, 네트워크의 파라미터 수 면에서도 ERAN은 Sugeno 등의 퍼지 모델에 비해 훨씬 적음을 알 수 있다. 표 3에는 시스템 모델링에 의해서 생성된 ERAN의 파라미터들을 나타냈다. 이 때의 바이어스값 a_0 는 -3.175이었다.

표 2. 2차원 시스템 모델링에 대한 성능 비교
Table 2. The Performance comparison for 2-D system modeling.

모델	훈련 RMS 오차	테스트 RMS 오차	은닉 유니트 수
RAN	0.206	0.170	35
ERAN	0.209	0.165	12

표 3. 시스템 모델링에 의해 생성된 ERAN의 파라미터

Table 3. The ERAN parameters generated by system modeling.

은닉 유니트 No.	중심벡터 (u_{1j}, u_{2j})	폭 벡터 $(\sigma_{1j}, \sigma_{2j})$	파라미터 (a_j)
1	(1.752, 2.152)	(1.025, 1.025)	-2.424
2	(4.334, 4.621)	(3.047, 3.047)	-3.148
3	(3.625, 1.575)	(1.872, 1.872)	-2.726
4	(1.847, 4.606)	(1.097, 1.097)	-2.090
5	(1.144, 1.484)	(6.309, 6.309)	12.451
6	(3.829, 3.639)	(2.496, 2.496)	1.254

IV. 결론

Platt이 제안한 RAN의 시스템 모델링 성능을 개선하기 위해 새로운 은닉 유니트 판별기준을 제안하고 파라미터 학습에 은닉 유니트의 폭을 포함시켜 기존 RAN을 확장한 ERAN을 제안하였다. 제안한 ERAN의 기본 아이디어는 RAN에서 발생하는 과도한 은닉 유니트의 생성을 방지하기 위해서 이미 존재하는 은닉 유니트의 활성화 정도를 관측하여 이 값을 새로운 은닉 유니트 생성 판별기준에 반영하는 것과 RBF 유니트의 폭을 학습하게 하여 모델링 성능을 향상시키는 것이다. 제안한 알고리즘의 성능을 시험하기 위해 비선형 정적 시스템 모델링 실험에 적용하였다. 실험결과로부터 순차 학습 및 랜덤 학습에 대하여 ERAN은 RAN에 비해 생성된 은닉 유니트의 수가 매우 적음에도 불구하고 모델링 성능이 우수함이 입증되었다.

제안한 ERAN은 학습초기에 0개의 은닉 유니트로 출발하여 훈련 데이터의 관측에 따라 RAN에 비해 훨씬 적은 RBF 은닉 유니트를 자동으로 생성할 수 있으므로 주어진 문제에 대한 준최적의 RBF 신경회로망을 구성할 수 있다.

참고 문헌

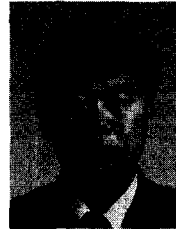
- [1] J. Moody and C. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, pp. 281-294, 1989.
- [2] M. Musavi, W. Amed, K. Chan, K. Faris, and D. Hummels, "On the training of radial basis function classifiers," *Neural Networks*, vol. 5, pp. 595-603, 1992.
- [3] J. Park and J. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Computation*, vol. 3, pp. 246-257, 1991.
- [4] C. Bishop, "Improving the generalization properties of radial basis function neural networks," *Neural Computation*, vol. 3, pp. 579-588, 1991.
- [5] S. Elanayar V.T. and Y. C. Shin, "Radial basis function neural network for approximation and estimation of non-linear stochastic dynamic systems," *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 594-603, 1994.
- [6] E. L. Sutanto, J. D. Mason, and K. Warwick, "Mean-tracking clustering algorithm for radial basis function center selection," *International Journal of Control*, vol. 67, no. 6, pp. 961-977, 1997.
- [7] L. I. Kuncheva, "Initializing of an RBF network by a genetic algorithm," *Neuro-computing*, vol. 14, pp. 273-288, 1997.
- [8] J. Platt, "A resource-allocating network for function approximation," *Neural Computation*, vol. 3, pp. 213-225, 1991.
- [9] J. Moody, "Fast learning in multi-resolution hierarchies," In *Advances in Neural Information Processing Systems*, I.D. Touretzky, ed., pp. 29-39, Morgan Kaufmann, San Mateo, 1989.
- [10] T. Poggio and F. Girosi, "Regularization algorithms for learning that are equivalent to multilayer networks," *Science*, vol. 247, pp. 978-982.
- [11] V. Kadirkamanathan and M. Niranjan, "A function estimation approach to sequential learning with neural networks," *Neural Computation*, vol.5, pp.954-975, 1993.
- [12] C.-C. Lee "Fuzzy logic in control systems: fuzzy logic controller - Part I, II" *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 2, pp. 404-435, 1990.
- [13] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 1, pp. 7-31, 1993.

저 자 소 개



崔鍾洙(正會員)

1969년 1월 18일생. 1990년 전북대학교 전기공학과 졸업(학사). 1992년 전북대학교 대학원 전기공학과 졸업(석사). 1996년 전북대학교 대학원 전기공학과 졸업(박사). 1994년 3월 ~ 1996년 7월 포스콘 기술연구소 선임연구원. 1996년 7월 ~ 현재 포항산업과학연구원 선임연구원. 주관심분야는 신경회로망, 퍼지 시스템, 지능 제어



權五臣(正會員)

1982년 전북대학교 전기공학과 졸업(학사). 1984년 전북대학교 대학원 전기공학과 졸업(석사). 1988년 전북대학교 대학원 전기공학과(박사). 1978년 ~ 1988년 전주공업전문대 전자계산학과 조교수. 1993년 ~ 1994년 일본 Osaka Electro-Communication University 객원 교수. 1989년 ~ 현재 군산대학교 제어계측공학과 부교수. 1997년 ~ 현재 군산대학교 공학연구소 소장. 주관심분야는 지능 제어, 퍼지 제어, 신경회로망 응용



金炯奭(正會員)

1956년 1월 21일생. 1980년 한양대학교 전자공학과 졸업(학사). 1982년 전북대학교 대학원 전기공학과 졸업(석사). 1992년 Dept. of Electrical and Computer Eng., University of Missouri-Columbia (Ph.D.). 1982년 ~ 1993년 국방과학연구소 선임연구원. 1993년 ~ 현재 전북대 전기.전자.제어공학부 조교수. 주관심분야는 신경회로망모델 개발, VLSI에 의한 신경회로망 하드웨어 구현, 로봇비전 및 무인헬기 원격제어