

회귀 신경망과 유한 상태 자동기계 동정화

원성환* · 송익호** · 민황기*** · 안태훈***

A Class of Recurrent Neural Networks for the Identification of Finite State Automata

Sung-Hwan Won · Iick-Ho Song · Hwang-Ki Min · Tae-Hun An

요 약

이 논문에서는 열개가 새로운 회귀 신경망을 제안하고, 그 신경망이 어떤 이산 시간 동적 시스템도 동정화할 수 있음을 보인다. 또한, 제안한 신경망을 써서 유한 상태 자동기계를 부호화, 동정화, 그리고 추출하는 데에 적용하여 그 성능을 살펴본다. 제안한 신경망에 고친 비용함수를 쓰고 혼합 그리디 모의 담금질 방법으로 학습시키면 유한 상태 자동기계를 동정화하는 성능이 일반적으로 다른 기법보다 더 낫다는 것을 모의실험으로 보인다.

ABSTRACT

A class of recurrent neural networks is proposed and proven to be capable of identifying any discrete-time dynamical system. The applications of the proposed network are addressed in the encoding, identification, and extraction of finite state automata. Simulation results show that the identification of finite state automata using the proposed network, trained by the hybrid greedy simulated annealing with a modified error function in the learning stage, exhibits generally better performance than other conventional identification schemes.

Keywords : Cost Function, System Identification, Finite State Automata, Hybrid Greedy Simulated Annealing, Recurrent Neural Network

1. 서론

동적 시스템(dynamical system)은 여러 분야에서 자연계를 모형화하는 데에 자주 쓰인다[1]. 동적 시스템 가운데 이산 시간 동적 시스템

(discrete-time dynamical system: 줄여서, 이시동시)은 여러 사람들이 연구해 왔는데[2], 컴퓨터 모의실험과 디지털 하드웨어를 통한 구현의 관점에서 어떤 시스템이든지 이시동시로 나타내는 것이 유리하다고 알려져 있다[3].

* 삼성전자 DMC 연구소 책임연구원

** 교신저자 한국과학기술원 전기 및 전자공학과 교수 (isong@ieee.org)

*** 한국과학기술원 전기 및 전자공학과 박사과정

접수일자 : 2012년 1월 10일, 수정일자 : 2012년 1월 26일, 심사완료일자 : 2012년 1월 30일

상태 공간이 이산적인 이시동시를 자동기계 (automaton)라고 부르고, 상태, 입력 부호, 출력 부호가 모두 유한개인 자동기계를 유한 상태 자동기계(finite state automaton: 줄여서, 유상자) 또는 유한상태 기계(finite state machine)라고 부른다[4]. 모든 계산 시스템의 바탕이 되는 튜링 기계를 외부 저장 매체 또는 기록 매체를 가진 유상자로 볼 수 있기 때문에, 유상자를 다룸으로써 계산 시스템을 이해하기 위한 기본적인 통찰력을 얻을 수 있다[5].

한편, 입출력 수열 짝으로 동적 시스템을 동정화하는(identification) 데에 회귀 신경망(recurrent neural network), 정보 이론 모형, 그리고 웨이블릿을 바탕으로 한 모형들이 주로 쓰여 왔다[6]. 이 가운데 회귀 신경망은 그 열개를 주어진 문제에 알맞도록 융통성 있게 정할 수 있고, 학습 방법 또한 이제까지의 여러 매개변수 최적화 알고리즘들 가운데에서 알맞은 것으로 고를 수 있기 때문에, 동적 시스템 동정화에 널리 쓰여 왔다[7]. 이제까지 여러 논문에서 뉴런들의 전달함수가 모두 시그모이드 함수인 회귀 신경망으로 유상자를 동정화하려 하였지만[8], 긴 수열들로 회귀 신경망을 제대로 학습시키지 못하였고, 결과적으로 일반화 성능이 나쁘다는 문제점이 드러났다. 따라서 몇몇 연구에서는 [9-10] 만족할 만한 일반화 성능을 얻고자 짧은 입출력 수열들을 여러 짝 써서 유상자를 동정화하는 방법들을 생각하였다. 최근에는 입출력 수열 한 짝으로 유상자를 동정화하는 온라인 동정화 방법이[11] 제안되었으나, 이 방법은 회귀 신경망을 학습시킬 때 너무 긴 입출력 수열 짝이 필요하다는 문제점을 지니고 있다.

이 논문에서는 새로운 회귀 신경망을 제안하고, 제안한 신경망이 어떤 이시동시든지 동정화할 수 있음을 보인다. 또한, 제안한 신경망을 써서 유상자를 부호화, 동정화, 그리고 추출하는 문제를 다룬다. 제안한 신경망의 상태들은 동정화할 시스템의 상태들을 바로 나타내므로, 매우 간단하게 제안한 신경망에 유상자를 부호화할 수 있다. 그 뿐만 아니라, 제안한 신경망의 열개가 유상자와 관련하여 잘 정의되어 있기 때문에 유상자의 추출(extraction) 또한 매우 간단하다.

II. 제안한 신경망

1. 이산 시간 동적 시스템

크기가 $n_U \times 1$ 인 벡터

$$\mathbf{U}[t] = [U_1[t] \ U_2[t] \ \dots \ U_{n_U}[t]]^T \quad (1)$$

이 이시동시의 입력이라면, 이 이시동시의 출력은

$$\begin{aligned} \mathbf{Y}[t] &= [Y_1[t] \ Y_2[t] \ \dots \ Y_{n_Y}[t]]^T \\ &= \mathbf{G}([\mathbf{U}^T[t] \ \mathbf{X}^T[t-1]]^T) \end{aligned} \quad (2)$$

처럼 쓸 수 있다. 여기서, $t=1,2,\dots$ 이고, $\mathbf{X}[t]$ 는 크기가 $n_X \times 1$ 인 상태 벡터로서 동적 시스템의 현재 상태를 뜻하며, 일반적으로 아래와 같이 상태 함수 \mathbf{F} 를 써서 나타낼 수 있다.

$$\begin{aligned} \mathbf{X}[t] &= [X_1[t] \ X_2[t] \ \dots \ X_{n_X}[t]]^T \\ &= \mathbf{F}([\mathbf{U}^T[t] \ \mathbf{X}^T[t-1]]^T) \end{aligned} \quad (3)$$

이때, 초기 상태는 $\mathbf{X}[0] = \mathbf{X}_0$ 으로 둔다. 한편, (2)와 (3)에서 출력 함수

$\mathbf{G}(\cdot) = [G_1(\cdot) \ G_2(\cdot) \ \dots \ G_{n_Y}(\cdot)]^T$ 의 크기는 $n_Y \times 1$ 이고, 다음 상태

함수 $\mathbf{F}(\cdot) = [F_1(\cdot) \ F_2(\cdot) \ \dots \ F_{n_X}(\cdot)]^T$ 의 크기는 $n_X \times 1$ 이며, G_i 와 F_i 는 $(n_U + n_X)$ 차원 공간에서 정의된 스칼라 함수이다.

2. 제안한 신경망

다층 퍼셉트론(multilayer perceptron)은 어떤 유한한 연속 함수도 어렵할 수 있다는 보편적 어림 정리에(universal approximation theorem)[12] 바탕을 두어, 이 논문에서는 다층 퍼셉트론을 여러 개 묶어서 그림 1에 보인 회귀 신경망을 제안한다. 제안한 신경망은 다층 퍼셉트론 둘이 이루는 0째

층부터 넷째 층까지로 나타낼 수 있다. 이때, 0층, 첫째 층, 둘째 층이 이루는 다층 퍼셉트론은 제안한 신경망이 동적 시스템의 상태를 어림하는 데에 쓰이고, 둘째 층, 셋째 층, 넷째 층이 이루는 다층 퍼셉트론은 제안한 신경망이 동적 시스템의 출력을 어림하도록 꾸민 것이다. 제안한 신경망에서 $i=1,2,3,4$ 일 때, n_i 는 i 째 층에 있는 뉴런의 개수를 나타내고, φ_i 는 i 째 층에 있는 뉴런들의 전달함수를 (transfer function) 나타내며, w_i 는 $(i-1)$ 째 층과 i 째 층 사이의 가중값을 나타내는 행렬로서 크기는 $n_i \times n_{i-1}$ 이다. 또한, w_r 은 둘째 층과 첫째 층 사이의 가중값을 나타내는 행렬로서 크기가 $n_1 \times n_2$ 이고, w_5 는 0층과 셋째 층 사이의 가중값을 나타내는 행렬로서 크기가 $n_3 \times n_0$ 이며, b_i 는 크기가 $n_i \times 1$ 인 i 째 층의 문턱값 (bias) 벡터이다.

제안한 신경망의 입력이

$$\mathbf{u}[t] = [u_1[t] \ u_2[t] \ \cdots \ u_{n_0}[t]]^T \quad (4)$$

이면, 출력은

$$\begin{aligned} \mathbf{y}[t] &= [y_1[t] \ y_2[t] \ \cdots \ y_{n_4}[t]]^T \\ &= \varphi_4(\mathbf{w}_4 \varphi_3(\mathbf{w}_5 \mathbf{u}[t] \\ &\quad + \mathbf{w}_3 \mathbf{x}[t-1] + \mathbf{b}_3) + \mathbf{b}_4) \end{aligned} \quad (5)$$

이고, 둘째 층의 출력, 곧 상태는,

$$\begin{aligned} \mathbf{x}[t] &= [x_1[t] \ x_2[t] \ \cdots \ x_{n_2}[t]]^T \\ &= \varphi_2(\mathbf{w}_2 \varphi_1(\mathbf{w}_1 \mathbf{u}[t] \\ &\quad + \mathbf{w}_r \mathbf{x}[t-1] + \mathbf{b}_1) + \mathbf{b}_2) \end{aligned} \quad (6)$$

이다. 이때, 초기 상태 $\mathbf{x}[0] = \mathbf{x}_0$ 으로 둔다. 식 (5)와 (6)에서 밑줄을 그어 쓴 것은, 보기를 들어, 크기가 $m \times n$ 인 행렬 \mathbf{x} 가 있을 때,

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f(x_{1,1}) & f(x_{1,2}) & \cdots & f(x_{1,m}) \\ f(x_{2,1}) & f(x_{2,2}) & \cdots & f(x_{2,m}) \\ \vdots & \vdots & \ddots & \vdots \\ f(x_{m,1}) & f(x_{m,2}) & \cdots & f(x_{m,m}) \end{bmatrix} \quad (7)$$

처럼 1차원 영역에서 정의된 스칼라 함수들로

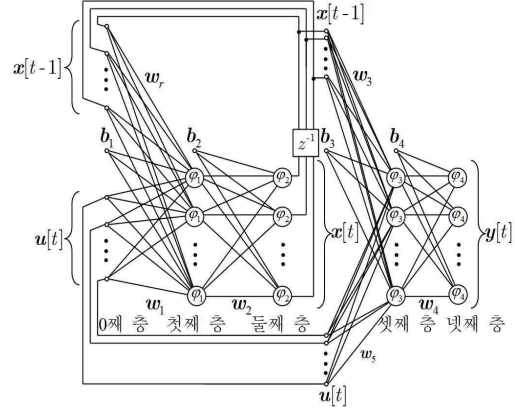


그림 1. 제안한 신경망의 열개
Fig. 1. Structure of the proposed network

이루어진 행렬을 뜻한다. 여기서 $x_{i,j}$ 은 행렬 \mathbf{x} 의 i 째 행 j 째 열의 원소이다.

3. 제안한 신경망의 동정화 능력 증명

정의역이 k 차원의 초입방체 $[0,1]^k$ 이고 연속인 스칼라 함수 공간을 $C([0,1]^k)$ 라고 하자.

정리 1: 함수 G_i 와 F_i 가 모두 $C([0,1]^{n_v+n_x})$ 의 원소이고 F_i 의 치역이 구간 $[0,1]$ 의 부분 집합인 어떤 이시동시도 제안한 신경망으로 동정화할 수 있다.

증명: 먼저, $n_0 = n_v$, $n_2 = n_x$, $n_4 = n_y$ 라고 두고 $\varphi_2(x) = \varphi_4(x) = x$, $\mathbf{x}_0 = \mathbf{X}_0$ 라고 두면, (5)와 (6)을 각각 아래와 같이 쓸 수 있다.

$$\mathbf{y}[t] = \mathbf{w}_4 \varphi_3(\mathbf{w}_5 \mathbf{u}[t] + \mathbf{w}_3 \mathbf{x}[t-1] + \mathbf{b}_3) + \mathbf{b}_4, \quad (8)$$

$$\mathbf{x}[t] = \mathbf{w}_2 \varphi_1(\mathbf{w}_1 \mathbf{u}[t] + \mathbf{w}_r \mathbf{x}[t-1] + \mathbf{b}_1) + \mathbf{b}_2. \quad (9)$$

이때, 전달함수 φ_1 과 φ_3 은 상수 함수가 아니면서 유한하고 단조 증가하는 연속 함수라고 하자. 그러면, $\{G_i\}_{i=1}^{n_4}$ 와 $\{F_j\}_{j=1}^{n_2}$ 가 연속이라는 것과 보편적 어림 정리에서, 제안한 신경망의 상태와 출력이 아래 두 부등식을 만족시키는 뉴런 개수 n_1 과

n_3 이 존재하고, 가중값 행렬 $\{w_i\}_{i=1}^4, \{b_i\}_{i=1}^4, w_5$ 와 w_r 이 존재함을 알 수 있다.

$$\|G([U^T[t]X^T[t-1]]^T) - y[t]\| < \varepsilon_o, \quad (10)$$

$$\|F([U^T[t]X^T[t-1]]^T) - x[t]\| < \varepsilon_s. \quad (11)$$

이때, $t = 1, 2, \dots$ 이고, ε_o 과 ε_s 는 0보다 큰 수이며, $\|\cdot\|$ 은 유클리드 거리이다.

III. 유한 상태 자동기계의 부호화, 동정화, 그리고 추출

1. 유한 상태 자동기계

유상자는 흔히 $M = (Q, \Sigma, \Gamma, \eta, \nu, q_I)$ 처럼 원소 여섯 개로 이루어진 모음으로 나타난다. 여기서, $Q = \{q_0, q_1, \dots, q_{|Q|-1}\}$ 은 상태 집합이고 $\Sigma = \{\sigma_0, \sigma_1, \dots, \sigma_{|\Sigma|-1}\}$ 은 입력 부호 집합이며, $\Gamma = \{\gamma_0, \gamma_1, \dots, \gamma_{|\Gamma|-1}\}$ 은 출력 부호 집합이다. 또한, $\eta: \Sigma \times Q \rightarrow Q$ 는 다음 상태 함수이고 $\nu: \Sigma \times Q \rightarrow \Gamma$ 는 출력 함수이며, q_I 는 초기 상태로서 Q 의 원소이다. 유상자의 상태, 입력, 출력들을 표준 이진법으로 부호화했다고 (standard binary encoded) 두면, 열벡터 q_i 와 σ_j 와 γ_k 의 크기는 각각 $\lceil \log_2 |Q| \rceil \times 1$, $\lceil \log_2 |\Sigma| \rceil \times 1$, $\lceil \log_2 |\Gamma| \rceil \times 1$ 이다. 또한, 출력 함수

$$\nu: \{0,1\}^{\lceil \log_2 |\Sigma| \rceil + \lceil \log_2 |Q| \rceil} \rightarrow \{0,1\}^{\lceil \log_2 |\Gamma| \rceil} \text{ 와}$$

다음 상태 함수

$$\eta: \{0,1\}^{\lceil \log_2 |\Sigma| \rceil + \lceil \log_2 |Q| \rceil} \rightarrow \{0,1\}^{\lceil \log_2 |Q| \rceil} \text{ 는}$$

부울 함수이다.

2. 부호화 방법

부호화 방법을 이야기하기에 앞서서, 몇 가지 기호들을 뜻매김하자. 집합 Σ 와 Q 의 데카르트 곱 $\Sigma \times Q$ 의 원소를 벡터

$$\begin{aligned} \rho_k &= [\sigma_{\lceil k/|Q| \rceil}^T \ q_k^T \text{mod}(|Q|)]^T \\ &= [\rho_{1,k} \ \rho_{2,k} \\ &\quad \dots \ \rho_{\lceil \log_2 |\Sigma| \rceil + \lceil \log_2 |Q| \rceil, k}]^T \end{aligned} \quad (12)$$

라 하고, 크기가 $(\lceil \log_2 |\Sigma| \rceil + \lceil \log_2 |Q| \rceil) \times \lceil \log_2 |Q| \rceil$ 인 행렬

$$P = [\rho_0 \ \rho_1 \ \dots \ \rho_{|\Sigma||Q|-1}] \quad (13)$$

을 생각하자. 여기서 $\text{mod}(\cdot)$ 는 모듈로(modulo) 연산이고, ρ_k 는 $(\lceil \log_2 |\Sigma| \rceil + \lceil \log_2 |Q| \rceil)$ 차원 열벡터이다. 다음으로, 세 조건

$$\lim_{x \rightarrow -\infty} \varphi(x) = 0, \quad (14)$$

$$\lim_{x \rightarrow \infty} \varphi(x) = 1, \quad (15)$$

$$\varphi(x) = 1 - \varphi(-x) \quad (16)$$

을 모두 만족시키는 단조 증가 함수 φ 의 집합을 Φ 라 두자. 또한, 함수

$$\text{sgn}(x) = \begin{cases} 1, & x > 0, \\ 0, & x = 0, \\ -1, & x < 0 \end{cases} \quad (17)$$

을 부호(sign) 함수라 하고, 크기가 $m \times n$ 이고 원소들이 모두 1인 행렬을 $\mathbf{1}_{m,n}$ 이라 쓰며, 행렬 A 의 i 째 행의 원소를 모두 더한 것이 i 째 원소인 열벡터를 $\text{sum}(A)$ 라고 하자. 이제, 제안한 신경망에 유상자를 부호화하는 방법을 이야기하자.

정리 2: 유상자 $M = (Q, \Sigma, \Gamma, \eta, \nu, q_I)$ 를 생각하자. 먼저, $\varphi_i \in \Phi$ 이고 초기 상태 $x[0] = x_0 = q_I$ 일 때, 제안한 신경망의 뉴런의 개수를 아래와 같이 두자.

$$n_0 = \lceil \log_2 |\Sigma| \rceil, \quad (18)$$

$$n_1 = n_3 = |\Sigma||Q|, \quad (19)$$

$$n_2 = \lceil \log_2 |Q| \rceil, \quad (20)$$

$$n_4 = \lceil \log_2 |\Gamma| \rceil. \quad (21)$$

또한,

$$0 < \beta_l < \frac{1}{2(n_l-1)}, \quad l = 1, 3, \quad (22)$$

$$0 < \beta_2 < \frac{1}{2n_2}, \quad (23)$$

$$0 < \beta_4 < \frac{1}{2}, \quad (24)$$

$$\xi_l > \frac{-\varphi_l^{-1}(\beta_l)}{\frac{1}{2} - n_2 \beta_2}, \quad l = 1, 3, \quad (25)$$

$$\xi_l > \frac{-\varphi_l^{-1}(\beta_l)}{\frac{1}{2} - (n_{l-1} - 1)\beta_{l-1}}, \quad l = 2, 4 \quad (26)$$

을 만족시키는 $\{\beta_l\}_{l=1}^4$ 와 $\{\xi_l\}_{l=1}^4$ 가 주어졌을 때, 제안한 신경망의 매개변수들을 다음과 같이 골랐다고 두자.

$$[w_1 \ w_r] = \xi_1 \underline{\text{sgn}} \left(P^T - \frac{1}{2} \mathbf{1}_{n_1, n_0 + n_2} \right), \quad (27)$$

$$w_2 = \xi_2 [\eta(\rho_0) \ \eta(\rho_1) \cdots \ \eta(\rho_{n_1-1})], \quad (28)$$

$$[w_5 \ w_3] = \xi_3 \underline{\text{sgn}} \left(P^T - \frac{1}{2} \mathbf{1}_{n_1, n_0 + n_2} \right), \quad (29)$$

$$w_4 = \xi_4 [\nu(\rho_0) \ \nu(\rho_1) \cdots \ \nu(\rho_{n_1-1})], \quad (30)$$

$$b_l = \begin{cases} \frac{\xi_l}{2} \mathbf{1}_{n_l, 1} - \xi_l (\sum(P))^T, & l = 1, 3, \\ -\frac{\xi_l}{2} \mathbf{1}_{n_l, 1}, & l = 2, 4. \end{cases} \quad (31)$$

그러면, $t = 1, 2, \dots$ 일 때,

$$|\hat{x}_j[t] - x_j[t]| < \beta_2, \quad j = 1, 2, \dots, n_2 \quad (32)$$

이고

$$|\hat{y}_i[t] - y_i[t]| < \beta_4, \quad i = 1, 2, \dots, n_4 \quad (33)$$

이다. 여기서,

$$\hat{\mathbf{x}}[t] = [\hat{x}_1[t] \ \hat{x}_2[t] \cdots \ \hat{x}_{n_2}[t]]^T \quad (34)$$

는 입력이 $\mathbf{u}[t]$ 일 때 M 의 표준 이진 부호화된 상태이고,

$$\hat{\mathbf{y}}[t] = [\hat{y}_1[t] \ \hat{y}_2[t] \cdots \ \hat{y}_{n_4}[t]]^T \quad (35)$$

는 M 의 표준 이진 부호화된 출력이다.

증명: [13] 참조.

제안한 신경망의 매개변수들을 정리 2에 보인 것처럼 알맞게 두면, 제안한 신경망의 상태의 원소와 M 의 상태의 원소와의 차가 늘 β_2 보다 작다는 것을 (32)가 보장한다. 또한, 제안한 신경망의 출력의 원소와 M 의 출력의 원소와의 차가 언제나 β_4 보다 작다는 것을 (33)이 보장한다. 이때, (23)의 β_2 와 (24)의 β_4 는 바라는 만큼 작게 잡을 수 있다.

3. 제안한 신경망을 써서 유한 상태 자동기계 동정화하기

2에서 이야기한 부호화 방법을 바탕으로 제안한 신경망을 써서 유상자를 동정화하는 것을 생각해 보자. 유상자를 동정화한다는 것은 그 유상자에 대한 다른 정보 없이, 주어진 입력력 수열 짝을 맞추는 모형을 세우는 것이다. 한편 동정화 하려는 유상자보다 상태의 개수가 더 많은 모형으로 입출력 수열 짝을 맞추면 그 모형은 일반화 성능이 좋지 않을 가능성이 높다. 따라서, 이 논문에서는 일반화 성능을 염두에 두어, 상태의 개수가 가장 적은 모형으로 유상자를 동정화 하려고 한다.

1) 초기 설정

입출력 수열 짝을 살펴보자 $|\Sigma|$ 와 $|\Gamma|$ 를 얻고, 네 층에 있는 뉴런들의 전달함수 $\{\varphi_l\}_{l=1}^4$ 는 Φ 에서 아무렇게나 고른다. 한편, 제안한 신경망으로 입출력 수열 짝을 맞추도록 하는 상태 개수들 가운데 가장 작은 값을 $|Q|$ 라고 하고, $|Q|$ 의 초기 값을 $2|\Sigma|$ 로 둔 다음 알맞은 $|Q|$ 의 값을 시행착오법으로 찾아나간다. 일단 $|\Sigma|$, $|\Gamma|$, $|Q|$ 가 정해지면, (18)-(21)에서 $\{n_l\}_{l=0}^4$ 를 얻을 수 있다. 또

한, (22)-(26)에서 $\{\beta_i\}_{i=1}^4$ 와 $\{\zeta_i\}_{i=1}^4$ 를 알맞게 고르면, (27), (29), (31)에서 (w_2 와 w_4 를 뺀) 모든 가중값 행렬과 문턱값 벡터를 얻을 수 있다.

2) 혼합 그리디 모의 담금질 방법을 써서 매개변수 조절하기

제안한 신경망으로 주어진 유상자를 동정화하려면 학습 알고리즘을 써서 w_2 와 w_4 를 알맞게 정해줘야 하는데, 이 논문에서는 혼합 그리디 모의 담금질 (hybrid greedy simulated annealing: 줄여서, 혼그모담) 방법을[14] 학습 알고리즘으로 쓴다. 혼그모담 방법은 지역 최적화 알고리즘과 전역 최적화 알고리즘을 섞은 매개변수 최적화 기법으로서, 지역 최적화 알고리즘으로는 레벤버그-마퀴트 (Levenberg-Marquardt: 줄여서, 레마) 알고리즘을 [15], [16] 쓰고, 전역 최적화 알고리즘으로는 그리디 모의 담금질 (greedy simulated annealing: 줄여서, 그모담) 방법을 [17] 쓴다. 레마 알고리즘은 신경망을 학습시키는 데 널리 쓰여 왔으며[18], 확률적으로 최적점을 찾아가는 알고리즘인 그모담 방법은 지역 최적점에 잘 빠지지 않는다는 장점이 있다. 이 두 방법을 섞은 혼그모담 방법은 전역 최적점으로 수렴한다는 것이 수학적으로 보장되어 있으며 수렴 속도도 다른 전역 최적화 알고리즘과 견주어 빠르다고 알려져 있다 [14].

한편, 제안한 신경망을 혼그모담 방법으로 학습시킬 때 아래 비용 함수를 쓴다.

$$\begin{aligned}
 E(w_2, w_4) &= \mu \sum_{t=1}^{\tilde{t}} \| \hat{y}[t] - y[t] \|^2 \\
 &+ (1-\mu) \sum_{t=1}^{\tilde{t}} \left\| \frac{1}{4} \mathbf{1}_{n_2,1} - \left(\mathbf{x}[t] - \frac{1}{2} \mathbf{1}_{n_2,1} \right) \right\|^2 \\
 &= \left\{ \sum_{t=1}^{\tilde{t}} \sum_{i=1}^{n_1} \epsilon_{4,i}^2[t] + \sum_{t=1}^{\tilde{t}} \sum_{j=1}^{n_2} \epsilon_{2,j}^2[t] \right\} \quad (36)
 \end{aligned}$$

위 (36)에서 $\hat{y}[t]$ 는 유상자의 출력 수열이고 $y[t]$ 와 $x[t]$ 는 각각 제안한 신경망의 출력과 상태

이며, \tilde{t} 는 입출력 수열의 길이이다. 한편, μ 는 구간 $(0, 1)$ 에 있는 수이고, \circ 은 아다마르 곱을 뜻하며,

$$\epsilon_{4,i}[t] = \mu^{\frac{1}{2}} (\hat{y}_i[t] - y_i[t]) \quad (37)$$

이고

$$\epsilon_{2,j}[t] = (1-\mu)^{\frac{1}{2}} \left\{ \frac{1}{4} - \left(x_j[t] - \frac{1}{2} \right)^2 \right\} \quad (38)$$

이다. 식 (36)의 첫째 항의 값은 $\hat{y}[t]$ 가 $y[t]$ 와 비슷할수록 작아지며, 둘째 항의 값은 $x[t]$ 의 모든 원소들의 값이 0 또는 1일 때 0으로 가장 작다. 이를 바탕으로 제안한 신경망의 출력과 상태가 각각 동정화하려는 유상자의 출력과 상태와 비슷할수록 비용함수 $E(w_2, w_4)$ 가 작아진다는 것을 알 수 있다. 식 (36)은 입출력 수열이 한 짝일 때의 비용함수인데, 수열이 여러 짝일 때에는 (36)을 일 반화하여 비용함수를 나타낼 수 있다.

이제, 혼그모담 방법을 써서 (36)을 최소화하는 w_2 와 w_4 를 얻는 과정을 자세하게 다루어 보도록 하자.

i) 초기화. 풀이 행렬 w_2 와 w_4 를 초기화한다. 곧, w_2 와 w_4 의 원소들을 각각 구간 $[0, \zeta_2]$ 와 $[0, \zeta_4]$ 에서 아무렇게나 고른다. 되풀이 횟수 τ 는 1로 두고, 초기 온도 T_0 과 최대 되풀이 횟수 $\tilde{\tau}$ 는 충분히 크게 잡아 풀이 벡터가 지역 최저점에 수렴하지 않도록 한다. 학습 목표 E_ϕ 는 충분히 작은 값으로 둔다.

ii) 담금질. 역 담금질 과정에 따라 온도를

$$T = \frac{T_0}{\tau} \quad (39)$$

로 낮춘다.

iii) 생성. 먼저, i 가 2일 때는 크기가 $n_2 \times n_1$ 이고, i 가 4일 때는 크기가 $n_4 \times n_1$ 이며, 서로 독립이고 구간 $(-\pi/2, \pi/2)$ 에 고르게 분포된 확률변수들로 이루어진 행렬을 v_i 라 하자. 현재의 풀이 행렬 w_i 로부터 새로운 풀이 행렬 $w_i^{(0)}$ 을

$$\omega_i^{(0)} = \mathbf{w}_i + T \tan(\mathbf{v}_i), \quad i = 2, 4 \quad (40)$$

으로 얻는다.

iv) 레마 알고리즘. 먼저, 크기가 $\{(n_2 + n_4)\tilde{t}\} \times 1$ 인 벡터 $\epsilon(\mathbf{w}_2, \mathbf{w}_4)$ 를

$$\epsilon(\mathbf{w}_2, \mathbf{w}_4) = [\tilde{\epsilon}_1^T \tilde{\epsilon}_2^T]^T \quad (41)$$

이라고 뜻매김하자. 여기서, $i = 2, 4$ 이고 $t = 1, 2, \dots, \tilde{t}$ 일 때, $\tilde{\epsilon}_i = [\epsilon_i^T[1] \epsilon_i^T[2] \dots \epsilon_i^T[\tilde{t}]]^T$ 이고 $\epsilon_i[t] = [\epsilon_{i,1}[t] \epsilon_{i,2}[t] \dots \epsilon_{i,n_i}[t]]^T$ 이다. 덧붙여, 크기가 $\{(n_2 + n_4)\tilde{t}\} \times \{n_1(n_2 + n_4)\}$ 인 행렬

$$\begin{aligned} \mathbf{J}(\mathbf{w}_2, \mathbf{w}_4) &= \frac{\partial \epsilon(\mathbf{w}_2, \mathbf{w}_4)}{\partial (\mathbf{w}_2, \mathbf{w}_4)} \\ &= [\mathbf{J}_2(\mathbf{w}_2, \mathbf{w}_4) \quad \mathbf{J}_4(\mathbf{w}_2, \mathbf{w}_4)] \end{aligned} \quad (42)$$

를 $\epsilon(\mathbf{w}_2, \mathbf{w}_4)$ 의 야코비 (Jacobian) 행렬이라고 뜻매김하자. 여기서, $i = 2, 4$ 이고 $j = 2, 4$ 일 때,

$$\mathbf{J}_j(\mathbf{w}_2, \mathbf{w}_4) = \left[\left(\frac{\partial \tilde{\epsilon}_i}{\partial \mathbf{w}_j} \right)^T \left(\frac{\partial \tilde{\epsilon}_i}{\partial \mathbf{w}_j} \right)^T \right]^T \quad (43)$$

이고,

$$\begin{aligned} &\frac{\partial \tilde{\epsilon}_i}{\partial \mathbf{w}_j} \\ &= \begin{bmatrix} \frac{\partial \epsilon_{i,1}[1]}{\partial w_{j,1,1}} & \frac{\partial \epsilon_{i,1}[1]}{\partial w_{j,2,1}} & \dots & \frac{\partial \epsilon_{i,1}[1]}{\partial w_{j,n_j n_1}} \\ \frac{\partial \epsilon_{i,2}[1]}{\partial w_{j,1,1}} & \frac{\partial \epsilon_{i,2}[1]}{\partial w_{j,2,1}} & \dots & \frac{\partial \epsilon_{i,2}[1]}{\partial w_{j,n_j n_1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \epsilon_{i,n_i}[\tilde{t}]}{\partial w_{j,1,1}} & \frac{\partial \epsilon_{i,n_i}[\tilde{t}]}{\partial w_{j,2,1}} & \dots & \frac{\partial \epsilon_{i,n_i}[\tilde{t}]}{\partial w_{j,n_j n_1}} \end{bmatrix} \end{aligned} \quad (44)$$

이다. 이때, $w_{j,m,n}$ 은 \mathbf{w}_j 의 m 째 행 n 째 열에 있는 원소를 뜻한다. 식 (42)의 야코비 행렬을 써서 아래 레마 갱신 과정

$$\omega_j \xrightarrow{((i+1)/\tau_{LM})} \omega_j \quad (45)$$

를 되풀이하러 $(\omega_2^{(i/\tau_{LM})}, \omega_4^{(i/\tau_{LM})})$ 에서 $\omega_j^{((i+1)/\tau_{LM})}$ 를 얻는다. 이때, 자연수 τ_{LM} 은 (45)를 되풀이하러 횟수이고

$$\begin{aligned} &\omega_j^{((i+1)/\tau_{LM})} \\ &= \begin{bmatrix} \omega_{j,1}^{((i+1)/\tau_{LM})} & \omega_{j,n_j+1}^{((i+1)/\tau_{LM})} & \dots & \omega_{j,(n_1-1)n_j+1}^{((i+1)/\tau_{LM})} \\ \omega_{j,2}^{((i+1)/\tau_{LM})} & \omega_{j,n_j+2}^{((i+1)/\tau_{LM})} & \dots & \omega_{j,(n_1-1)n_j+2}^{((i+1)/\tau_{LM})} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{j,n_j}^{((i+1)/\tau_{LM})} & \omega_{j,2n_j}^{((i+1)/\tau_{LM})} & \dots & \omega_{j,n_1 n_j}^{((i+1)/\tau_{LM})} \end{bmatrix} \end{aligned} \quad (46)$$

이며, $i = 0, 1, \dots, \tau_{LM} - 1$ 이고 $j = 2, 4$ 이다. 또한, \mathbf{I}_m 은 크기가 $m \times m$ 인 단위행렬이며, $\omega_{j,k}^{((i+1)/\tau_{LM})}$ 은 $\omega_j^{((i+1)/\tau_{LM})}$ 의 k 째 원소이고, $\text{vec}(\mathbf{A})$ 는 행렬 \mathbf{A} 를 벡터화한 것, 다시 말해서, 행렬 \mathbf{A} 의 열들을 순서대로 한 줄로 위아래로 둔 열벡터를 뜻한다. 식 (46)에서 $\omega_j^{((i+1)/\tau_{LM})}$ 은 $\omega_j^{((i+1)/\tau_{LM})}$ 을 ‘역 벡터화’한 것임을 눈여겨보도록 하자.

한편, (45)에서 λ 는 발걸음 너비를 조정하는 인자이며 (damping factor), λ 가 커질수록 레마 알고리즘은 표준 최급강하법과 (standard gradient descent) 가까워지고, 작아질수록 뉴턴 알고리즘과 (Newton's algorithm) 가까워진다. 이 논문에서는 레마 알고리즘을 시작할 때, λ 를 아주 작은 값 λ_0 으로 초기화한다. 식 (45)에서 얻은 새로운 풀이 벡터의 비용이 원래 벡터의 비용보다 작으면 (크면) 새 (원래) 벡터를 저장하고 λ 를 1보다 큰 수 λ_1 로 나눈다 (λ_1 과 곱한다). 그다음 i 를 1만큼 (0만큼) 늘리고 같은 과정을 되풀이한다. 끝으로, λ 가 $\tilde{\lambda}$ 에 이르면, i 가 $\tau_{LM} - 1$ 이 아니더라도 레마 알고리즘을 마친다.

위에서 말한 과정을 거쳐 $(\omega_2^{(0)}, \omega_4^{(0)})$ 에서 $(\omega_2^{(1)}, \omega_4^{(1)})$ 을 만든다.

v) 비용 계산. 비용 $E(w_2, w_4)$ 와 $E(\omega_2^{(1)}, \omega_4^{(1)})$ 을 계산한다.

vi) 고르기. 혼그모담 방법의 해를 다음과 같이 정한다.

$$(w_2, w_4) = \begin{cases} (\omega_2^{(1)}, \omega_4^{(1)}), & E(w_2, w_4) \geq E(\omega_2^{(1)}, \omega_4^{(1)}) \text{ 일 때} \\ (w_2, w_4), & E(w_2, w_4) < E(\omega_2^{(1)}, \omega_4^{(1)}) \text{ 일 때.} \end{cases} \quad (47)$$

vii) 끝내기. 비용 $E(w_2, w_4)$ 가 E_{ϕ} 보다 작거나 τ 가 $\tilde{\tau}$ 에 이르면, 혼그모담 방법을 끝내고, 그렇지 않으면, $\tau = \tau + 1$ 로 두고 ii)로 간다. 여기서, 혼그모담 방법이 끝났을 때 $E(w_2, w_4)$ 가 E_{ϕ} 보다 작거나 같으면 학습이 성공했다고 하고, 그렇지 않으면 학습이 실패했다고 한다.

III. 3.다. 상태 개수 최소화

앞에서 이야기했듯이, 이 논문에서는 제안한 신경망이 주어진 유상자를 잘 동정화하면서 좋은 일반화 성능을 보이도록 상태 개수의 최솟값 $|Q|$ 를 시행착오법으로 찾아낸다. 이를 좀더 자세히 살펴보자. 먼저, 처음에 $|Q|=2|I|$ 로 두고, 혼그모담 방법을 써서 제안한 신경망을 학습시킨다. 이 학습이 성공적이면 (성공적이지 않으면) $|Q|$ 를 1만큼 줄인 (늘린) 값을 새로 $|Q|$ 로 둔 다음 제안한 신경망을 처음부터 다시 학습시킨다. 이와 같이 학습이 성공적이면 (성공적이지 않으면) 학습에 실패할 (성공할) 때까지 $|Q|$ 의 값을 1씩 줄이면서 (늘리면서) III.3.가와 III.3.나를 되풀이하고, 마지막으로 성공한 학습에서 쓴 $|Q|$ 를 상태 개수의 최솟값으로 받아들인다. 동정화의 모든 과정을 그림 2에 간추려 나타내었다.

IV. 모의실험 결과

이 절에서는 제안한 방법으로 유상자를 동정화하는 성능을 이제까지의 방법들과 견주어 본다. 동정화가 끝나면 유상자의 추출은 매우 간단하기 때문에 동정화만 다루겠다.

1. 검사 문제와 설정

주어진 유상자를 동정화하였을 때, 제안한 방법의 일반화 성능을 다른 기법들과 견주기 위해 토미타[19] 자동기계 7개로 모의실험을 하였다. 표 1에 자세히 나타낸 대로 토미타 자동기계마다 수열의 총 개수와 가장 긴 수열의 길이를 다르게 하여 입력 수열을 만들고 출력 수열을 얻었다. 이때, μ 는 0.5로 두었고, 제안한 신경망에서 뉴런의 전달함수를

$$\varphi_l(x) = \frac{1}{1 + e^{-x}}, \quad l = 1, 2, 3, 4 \quad (48)$$

처럼 두었으며,

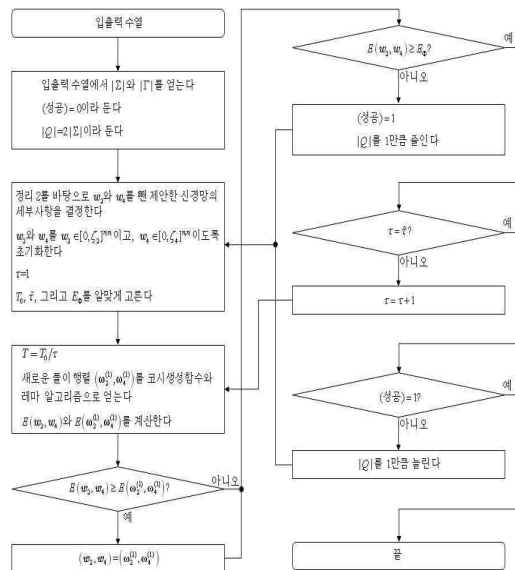


그림 2. 제안한 동정화 방법의 블록선도
Fig. 2. Block diagram of the proposed identification scheme

$$\beta_l = \frac{1}{4(n_l - 1)}, \quad l = 1, 2, 3, 4, \quad (49)$$

$$\zeta_1 = \zeta_3 = \frac{200 \ln \frac{1 - \beta_2}{\beta_2}}{1 - 2n_2\beta_2}, \quad (50)$$

$$\zeta_2 = \zeta_4 = \frac{2.4 \ln \frac{1 - \beta_2}{\beta_2}}{1 - 2(n_1 - 1)\beta_1} \quad (51)$$

로 두었고, w_2 와 w_4 의 원소들은 구간 $[0, \zeta_2]$ 에서 아무렇게나 골랐다. 그리고 여러 짝의 입력력 수열을 써서 각 유상자마다 100 번씩 반복하여 동정화를 했다.

2. 모의실험 결과 및 일반화 성능 검사

제한한 신경망으로 토미타 자동기계 7 개를 각각 100 번씩 동정화한 결과는 모두 성공적이었으며 올바른 상태 개수 $|Q|$ 를 찾았다. 찾은 상태 개

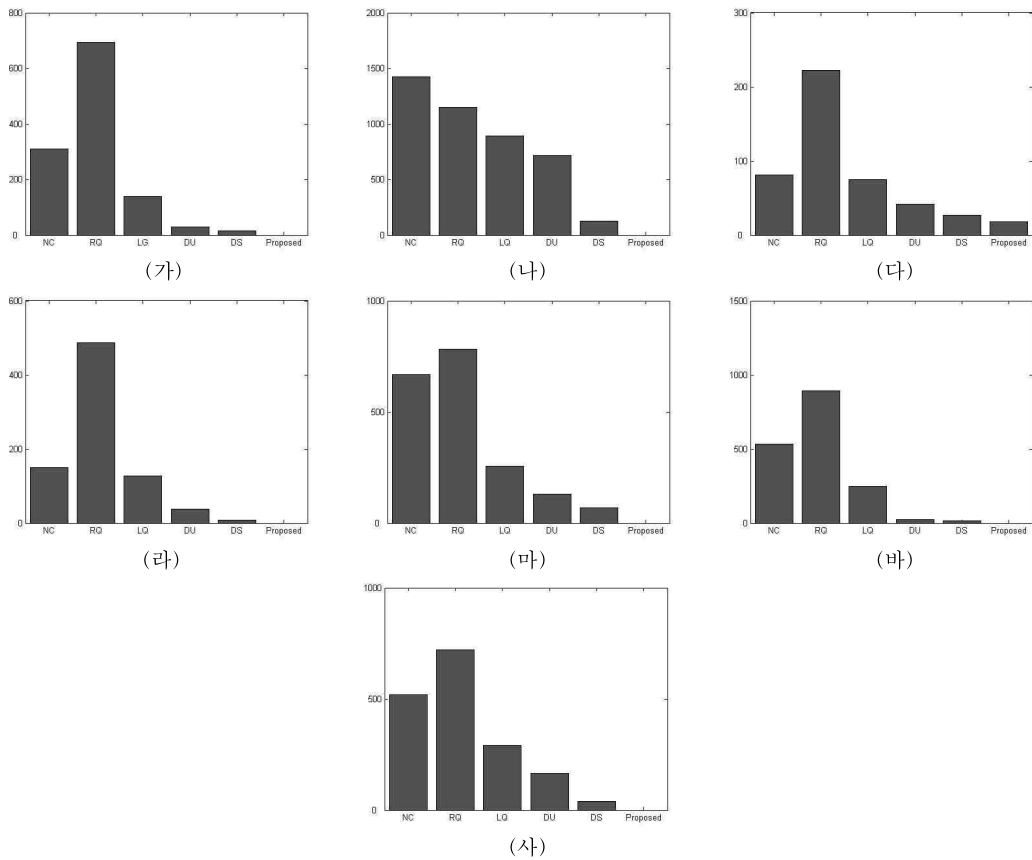


그림 3. 여섯 가지 동정화 방법의 일반화 성능: 비클러스터링 (no clustering) 기법 (NC), 강화 양자화 기법 (rigid quantization: 줄여서, RQ), 학습하고 양자화하는 (quantize) 기법 (LQ), 포기의 (Forgy) 알고리즘을 사용한 비지도 (unsupervised) 방식의 동적인 온라인 클러스터링과 상태 추출 기법 (dynamic online clustering and state extraction: 줄여서, DOLCE) (DU), 합침 모형을 사용한 지도 (supervised) 방식의 DOLCE (DS), 그리고 제안한 방법. (가) 토미타 1', (나) 토미타 2', (다) 토미타 3', (라) 토미타 4', (마) 토미타 5, (바) 토미타 6, (사) 토미타 7'

Fig. 3. Generalization performances of six identification methods: no clustering (NC), rigid quantization (RQ), learn then quantize (LQ), dynamic online clustering and state extraction (DOLCE) in unsupervised mod using Forgy's algorithm (DU), DOLCE in supervised mode using a mixture model (DS), and the proposed scheme (proposed) (가) Tomita 1', (나) Tomita 2', (다) Tomita 3', (라) Tomita 4', (마) Tomita 5, (바) Tomita 6, (사) Tomita 7'

표 1. 동정화 때 쓴 수열들의 세부사항과 동정화로 얻은 상태 개수

Table 1. Specifications of sequences used in the identification and estimated value of $|Q|$

문제	가장 긴 수열의 길이	수열의 개수	$ Q $
토미타 1'	5	60	1
토미타 2'	8	45	2
토미타 3'	7	100	4
토미타 4'	10	100	3
토미타 5	10	80	4
토미타 6	6	60	3
토미타 7'	10	100	4

수를 표 1에 나타내었다. 한편, 길이가 3000인 입력 수열을 아무렇게나 만들고, 7개의 유상자에서 출력 수열을 얻은 뒤에 제안한 방법과 기존의 여러 방법들의 일반화 성능을 검사하고 그림 3에 그 결과를 보였다. 세로축에는 잘못 분류한 비트들의 수를 나타내었는데, 이는 초기 가중값을 다르게 하여 100번 실험한 결과를 평균한 값이다. 가로축에는 여러 방법들의 이름을 간추려 적었다. 그림 3에서 알 수 있듯이, 제안한 기법은 다른 기법들과 견주어 뛰어난 성능을 보여준다.

V. 결론

이 논문에서는 새로운 열개의 회귀 신경망을 제안하였다. 제안한 신경망은 어떤 이시동시든지 동정화할 수 있는 것이다. 또한, 제안한 신경망에 고친 비용함수를 쓰고 혼그모담 방법으로 학습시키면 유상자를 동정화하는 성능이 다른 기법들보다 더 낫다는 것을 모의실험으로 보였다. 제안한 신경망은 유상자와 관련하여 잘 정의된 열개를 가진 덕분에, 유상자를 부호화하고 동정화하고 추출할 때 다른 회귀 신경망들보다 유리하다는 장점이 있다.

후 기

이 논문은 교육과학기술부의 재원으로 한국연구재단이 선정하여 지원하는 도약연구사업(2011-0016462)의 연구 결과 가운데 하나입니다.

참고문헌

- [1] R. Roy, T. W. Murphy Jr., T. D. Maier, Z. Gills, and E. R. Hunt, "Dynamical control of a chaotic laser: Experimental stabilization of a globally coupled system," *Phys. Rev. Lett.*, vol. 68, no. 9, pp. 1259-1262, Mar. 1992.
- [2] P. Tichavský, C. H. Muravchik, and A. Nehorai, "Posterior Cramér-Rao bounds for discrete-time nonlinear filtering," *IEEE Trans. Signal Process.*, vol. 46, no. 5, pp. 1386-1396, May 1998.
- [3] Z. Yi and K. K. Tan, "Multistability of discrete-time recurrent neural networks with unsaturating piecewise linear activation functions," *IEEE Trans. Neural Netw.*, vol. 15, no. 2, pp. 329-336, Mar. 2004.
- [4] J. F. Kolen and S. C. Kremer, *A Field Guide to Dynamical Recurrent Networks*, Piscataway, NJ: IEEE Press, 2001.
- [5] M. Minsky, *Computation: Finite and Infinite Machines*, Englewood Cliffs, NJ: Prentice-Hall, 1967.
- [6] R. Ghanem and F. Romeo, "A wavelet-based approach for model and parameter identification of non-linear systems," *Int. J. Non-Linear Mech.*, vol. 36, no. 5, pp. 835-859, July 2001.
- [7] C.-H. Lee and C.-C. Teng, "Identification and control of dynamic systems using recurrent fuzzy neural networks," *IEEE Trans. Fuzz. Syst.*, vol. 8, no. 4, pp. 349-366, Aug. 2000.
- [8] M. Gori, M. Maggini, E. Martinelli, and G. Soda, "Inductive inference from noisy examples using the hybrid finite state filter," *IEEE Trans. Neural Netw.*, vol. 9, no. 3, pp. 571-575, Mar. 1998.
- [9] S. Das and M. Mozer, "Dynamic on-line clustering and state extraction: An approach to symbolic learning," *Neural Netw.*, vol. 11, no. 1, pp. 53-64, Jan. 1998.
- [10] K. Arai and R. Nakano, "Stable behavior in

a recurrent neural network for a finite state machine," *Neural Netw.*, vol. 13, no. 6, pp. 667-680, July 2000.

- [11] I. Gabrijel and A. Dobnikar, "On-line identification and reconstruction of finite automata with generalized recurrent neural networks," *Neural Netw.*, vol. 16, no. 1, pp. 101-120, Jan. 2003.
- [12] K.-I. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Netw.*, vol. 2, no. 3, pp. 183-192, May 1989.
- [13] S. H. Won, *A Class of Recurrent Neural Networks for Finite State Automata and Discrete-Time Chaotic Systems*, Ph.D. Dissertation, Dept. Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Korea, 2011.
- [14] Y. Lee, J.-S. Lee, S.-Y. Lee, and C. H. Park, "Improving generalization capability of neural networks based on simulated annealing," *Proc. Cong. Evolut. Comput.*, Singapore, Sep. 2007, pp. 3447-3453.
- [15] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly J. Appl. Math.*, vol. 2, no. 2, pp. 164-168, June 1944.
- [16] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *J. Soc. Ind., Appl. Math.*, vol. 11, no. 2, pp. 431-441, June 1963.
- [17] P. J. M. V. Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*, Dordrecht, The Netherlands: Kluwer, 1987.
- [18] G. Lera and M. Pinzolas, "Neighborhood based Levenberg-Marquardt algorithm for neural network training," *IEEE Trans. Neural Netw.*, vol. 13, no. 5, pp. 1200-1203, Sep. 2002.
- [19] K. P. Unnikrishnan and K. P. Venugopal, "Alopex: A correlation-based algorithm for feedforward and recurrent neural networks,"

Neural Comput., vol. 6, no. 3, pp. 469-490, May 1994.

저자약력

원 성 환 (Sung-Hwan Won)



비회원

2005년 2월: 한국과학기술원
전기 및 전자공
학과 공학사
2007년 2월: 한국과학기술원
전기 및 전자공
학과 공학석사
2011년 8월: 한국과학기술원
전기 및 전자공
학과 공학박사
2011년 8월~현재: 삼성전자
DMC 연구소
책임연구원

<관심분야> 회귀 신경망, 기지국 표준

송 익 호 (Iick-Ho Song)



종신회원

1982년 2월, 1984년 2월: 서울
대학교 전자공학과 공학
사 (준최우등), 공학석사
1985년 8월, 1987년 5월: 펜실
베이니아대학교 전기공학과
공학석사, 공학박사
1987년 3월~1988년 2월: 벨
통신연구소 연구원
1988년 3월~현재: 한국과학기술
원 전기 및 전자공학과
조교수, 부교수, 교수
1995년 1월~현재: 한국통신학
회 논문지 편집위원, 편집
부위원장
대한전자공학회, 한국음향학회,
한국통신학회 평생회원,
IET 석학회원, IEEE 석학회원

<관심분야> 통계학적 신호처리와 통신이론, 신호
검파와 추정, 이동통신

