

論文94-31B-7-20

# 시변 2상 최적화 및 이의 신경회로망 학습에의 응용

## (Time-Varying Two-Phase Optimization and its Application to Neural Network Learning)

明 鉉\*, 金鍾煥\*

(Hyeon Myeong and Jong-Hwan Kim)

## 要約

이상 신경회로망은 제한 조건이 있는 최적화 문제가 주어졌을 때, 적절하고 정확한 해를 찾아준다. 시변 프로그래밍 신경회로망은 시변 최적화 문제를 풀 수 있도록 고안된 수정 급경사 알고리즘의 한 종류이다. 본 논문에서는 이상 신경망과 시변 신경망의 장점들을 살려서 고안한 시변 이상 신경망을 제안한다. 다층 신경회로망의 학습은 시변 최적화 문제로 여겨질 수 있기 때문에 제안된 알고리즘으로써 다층 신경망의 학습 즉, 시스템 동정과 함수 모사에 적용시켜 보고, 또한 신경망의 학습시 연결 가중 계수에 제한 조건이 있는 경우에도 학습이 가능함을 보인다.

## Abstract

A two-phase neural network finds exact feasible solutions for a constrained optimization programming problem. The time-varying programming neural network is a modified steepest-gradient algorithm which solves time-varying optimization problems. In this paper, we propose a time-varying two-phase optimization neural network which incorporates the merits of the two-phase neural network and the time-varying neural network. The proposed algorithm is applied to system identification and function approximation using a multi-layer perceptron. Particularly, training of a multi-layer perceptron is regarded as a time-varying optimization problem. Our algorithm can also be applied to the case where the weights are constrained. Simulation results prove the proposed algorithm is efficient for solving various optimization problems.

## 1. 서론

인공 신경회로망은 제한 조건이 있거나 없는 여러

종류의 최적화 문제들에 적용되어 왔었고 좋은 성능들을 보여 왔다.

Tank와 Hopfield 회로망(1986)은 수학적인 프로그래밍 문제를 푸는데 사용한 최초의 신경회로망이었다.<sup>(1)</sup> 그들의 논문에서는 선형 프로그래밍 문제를 페회로망으로 대응시켜서 전체적인 에너지 함수가 최소점을 찾을 때까지 점차 감소해 나가도록 하였다. 그러나 그들의 회로에서 평형점은 Kuhn-Tucker(K-

\*正會員, 韓國科學技術院 電氣 및 電子工學科  
(Dept. of Elec. Eng., Korea Advanced  
Institute of Science and Technology (KAIST)  
接受日字: 1993年 10月 11日

T)의 최적 조건을 만족시키지 못하기 때문에 본래 문제의 해가 아닐 수도 있다는 단점을 가지고 있었다.

많은 연구자들이 비선형 프로그래밍 문제를 풀기 위해서 목적 함수와 제한 함수를 모사하는 전자 회로를 사용하는 방법을 택했다. 1984년에 Chua와 Lin은 임의의 비선형성을 포함하는 비선형 문제를 풀 수 있도록 일반적인 방법을 제시했다.<sup>[2]</sup> 이 논문이 발표된 후에 Kennedy와 Chua(1988)는 Hopfield와 Tank의 회로망이 Chua와 Lin이 제안한 정규화된 비선형 프로그래밍 회로의 한 특별한 경우임을 보였다. Kennedy와 Chua는 Chua와 Lin의 정규화 비선형 프로그래밍 회로 모델을 동적 확장시켰을 때의 안정성을 보이고, 회로가 가지는 해의 안정성을 보장할 수 있음을 보여주었다.<sup>[3]</sup>

Kennedy와 Chua는 최적화 이론의 관점에서 K-T 최적 조건과 페널티 함수 정리를 둘 다 만족시키는 회로망을 개발하였다. 그러나 제한 조건이 있는 문제의 해가 적정 영역의 경계에 있을 때에는 그 회로망의 평형점이 적정 영역의 밖에 있는 근사해로 수렴해 버리고 만다.

Maa와 Shanblatt(1992)는 Kennedy와 Chua의 회로망에 존재하는 그러한 문제에 대한 대안으로서 이상(two-phase) 최적화 신경회로망을 제안하였다.

<sup>[4]</sup> 그 회로는 미리 선택된 시간 스위치에 의해서 상(phase)을 바꾸게 되고, 그 때에는 이전의 1상과 다른 식에 의해 회로가 동작하게 된다. 두 개의 상을 거친 후에는 K-T 최적 조건을 만족시키게 되어 평형점이 컨벡스(convex) 문제의 정확한 해가 되고, 예전보다 더 작은 페널티 계수(penalty parameter)로써 정확한 해를 찾을 수 있게 된다.

1992년에는 연립 방정식이나 연관된 문제를 푸는 다른 형태의 신경회로망들이 많은 연구가들에 의해 제안되었다. 그러한 신경망의 예를 들자면 Zhang등이 제안한 조건부 최적화를 위한 이차 신경망<sup>[5]</sup>, Zhang과 Constantinides에 의해 제안된 라그랑지 프로그래밍 신경망<sup>[6]</sup>, Cichocki와 Unbehauen의 신경망<sup>[7]</sup> 등이 있다. 그러나 그들의 주요 관심사는 시불변하는 문제를 위한 정적인 최적화에 있었다.

많은 실제적인 문제에서 자주 생기는 시변 최적화 문제를 다룰 때에는 정적 최적화의 기법을 도입하였을 때에 그 문제의 시변하는 특성으로 인한 영향을 고려해야만 한다. 시변 비선형 프로그래밍 신경회로망이 Feng와 Zhao에 의해 1992년에 제안되었다.<sup>[8]</sup> 그들은 시간에 따라 변하는 함수의 시간 변화율을 고려함으로써 최적해를 찾기 위한 수정 급경사법을 고안하였고, 그 알고리즘의 수렴성을 증명하였다.

본 논문에서는 이상 신경회로망과 시변 신경회로망의 장점들을 이용한 시변 이상 신경회로망이 제안된다. 그리고 다층 신경회로망의 학습은 시변 최적화 문제로 여길 수 있기 때문에 제안된 알고리즘을 써서 다층 신경망을 이용한 동적 시스템의 동정(identification)에 응용해 본다. 또한, 다층 신경회로망의 연결 가중 계수(connection weight)의 크기가 제한되는 실제적인 경우를 고려해 보고, 이를 시변 비선형 프로그래밍으로 간주하여 제안한 알고리즘을 적용시켜 본다.

본 논문은 다음과 같이 구성된다. II절에서는 문제의 수학적 정의를 포함하여 몇 가지 기본적인 정리에 대하여 알아보고, 근래에 연구되어진 두 종류의 최적화 신경회로망에 대해 소개한다. III절에서는 시변 이상 신경회로망에 대해 설명하고, 신경회로망 학습에 적용시킬 때의 구체적인 알고리즘을 유도한다. IV절에서는 컴퓨터 시뮬레이션을 통하여 시변 이상 신경회로망의 유용성과 신경회로망 학습에의 적용 가능성을 알아본다. 그리고 마지막 V절에서는 결론을 다루었다.

## II. 최적화 신경회로망

### 1. 수학적 기초<sup>[4]</sup>

프로그램(P)를 다음과 같이 정의하자 :

다음의 제한 조건을 만족하면서  $f(x)$ 를 최소화하는 문제

$$g_i(x) \leq 0, \dots, g_r(x) \leq 0, \quad h_j(x) = 0, \dots, h_m(x) = 0 \quad (1)$$

여기서  $f$ 와  $g_i$ 는  $R^n$ 에 속하는 함수이고,  $h_j$ 도  $m \leq n$ 인  $R^n$ 에 속하는 함수이다.  $f$ 와  $g_i$ 가 볼록(convex) 함수들이고,  $h_j$ 는 볼록이며 오목(concave)인 아파인(affine) 함수일 때, (P)는 컨벡스 프로그램이라고 불리운다. 벡터  $x$ 가 만일 (P)의  $r+m$ 개 제한 조건을 만족하기만 하면  $x$ 는 적정해(feasible solution)라고 불리운다.

$g_{r+1} = h_j, g_{r+2} = -h_j$ 라고 정의하면 (P)와 본질적으로 같은, 확장 프로그램(P')은 다음과 같이 표현될 수 있다 :

다음의 제한 조건을 만족하면서  $f(x)$ 를 최소화하는 문제

$$g_i(x) \leq 0, \quad 1 \leq i \leq r+2m \quad (2)$$

다음의 정리는 Kuhn-Tucker의 최적화 정리로 알

려져 있다

K-T 정리 : (P)는 컨벡스 프로그램이고,  $\bar{x}$ 는 (P)의 적정해라고 하자. 또한, 각각의  $g_i$ 와  $h_j$ 는  $\bar{x}$ 에서 미분가능하다고 가정하고  $\bar{x}$ 는 정규점(regular point)이라고 생각하자. 그러면 다음의 두 조건을 만족하는  $\lambda = [\lambda_1 \dots \lambda_r]^T$ 와  $\mu = [\mu_1 \dots \mu_m]^T$ 가  $\bar{x}$ 와 함께 존재하기만 하면  $\bar{x}$ 는 (P)의 최적해이다.

- i.  $\lambda_i \geq 0, g_i(\bar{x}) \leq 0, \lambda_i g_i(\bar{x}) = 0, i = 0, \dots, r$
- ii.  $\nabla f(\bar{x}) + \sum_{i=1}^r \lambda_i \nabla g_i(\bar{x}) + \sum_{j=1}^m \mu_j \nabla h_j(\bar{x}) = 0$  (3)

여기서 경사도(gradient)인  $\nabla g_i(x), \nabla h_j(x), i \in \{i | g_i(x) = 0\}, 1 \leq j \leq m$ 가 선형 독립일 때  $x$ 는 정규점이라고 한다. 다음의 정리는 페널티 함수 정리(Penalty function theorem)로 알려져 있다.

페널티 함수 정리 : 페널티 계수  $\{s_k\}_{k=1}^\infty$ 는 음수가 아니고, 엄격히 증가하여 무한대로 발산하는 수열이라고 할 때 다음과 같은 함수를 정의하자.

$$L(s, x) = f(x) + \frac{s}{2} \sum_{i=1}^{r+2m} (g_i^+(x))^2 \quad (4)$$

여기서  $g_i^+(x) = \max(0, g_i(x))$  이다

$L(s_k, x)$ 의 최소점을  $x_k$ 라고 하면, 수열  $\{x_k\}_{k=1}^\infty$ 의 임의의 극한값은 (P)과 (P)의 최적해가 된다. 또한,  $x_k \rightarrow \bar{x}$  이고  $\bar{x}$ 가 정규점이라면  $s_k g_i^+(x_k) \rightarrow \lambda_i$  이 되고 이 때의  $\lambda_i$ 는  $g_i$ 의 라그랑지 적수(lagrange multiplier)가 된다.

2. 이상 최적화 신경회로망

(P)과 동일한 수학적 표기를 사용하면, Kennedy와 Chua의 회로망(1988)은 다음과 같이 정의된다. [3]

$$\dot{x} = -\nabla f(x) - s \left[ \sum_{i=1}^{r+2m} g_i^+(x) \nabla g_i(x) \right] \quad (5)$$

이렇게 되면 페널티 함수 즉, Lyapunov 함수  $L(s, x)$ 의 시간 미분치는  $dL(s, x)/dt = -\dot{x}^2 \leq 0$  이 되어서 안정성이 보장된다.

Maa와 Shanblatt의 신경망(1992)은 아래와 같이 수식화되는 두 개의 상으로 구성된다. [4] 첫 번째 상 ( $0 \leq t < t_1$ )에서는

$$\dot{x} = -\nabla f(x) - s \left[ \sum_{i=1}^r g_i^+(x) \nabla g_i(x) + \sum_{j=1}^m h_j(x) \nabla h_j(x) \right] \quad (6)$$

이고, 여기서  $t_1$ 은 기정 스위칭 시간이다.

그리고 두 번째 상 ( $t \geq t_1$ )에서는

$$\dot{x} = -\nabla f - \left[ \sum_{i=1}^r \nabla g_i (s g_i^+ + \lambda) + \sum_{j=1}^m \nabla h_j (s h_j + \mu) \right] \quad (7)$$

이다. 여기서  $\dot{\lambda} = \epsilon (s g_i^+), \dot{\mu} = \epsilon (s h_j)$ 이고,  $\epsilon$ 은 작은 양의 상수이다.

이 회로는  $g_i^+(x)=0, h_j(x)=0, \lambda > 0$  이고,  $\nabla f + \nabla g \lambda + \nabla h \mu = 0$ 일때 평형점에 도달한다. 이렇게 되면 이 회로는 K-T의 최적 조건을 만족하게 되므로, 신경망의 평형점이 바로 컨벡스 프로그램 (P)의 정확한 총체적 해(global solution)가 된다.

3. 시변 프로그래밍 신경회로망

시변 함수  $f(x, t)$ 의 최소점을 찾는 문제를 시변 비선형 프로그래밍이라고 부른다. Feng와 Zhao (1992) [8]는 시변 함수의 최적해를 찾기 위해 그 함수의 시간 변화치를 고려해 넣음으로써 수정된 급경사 방법을 제안하고, 이의 수렴성을 증명하였으며, 또한 그 알고리즘을 구현하기 위해서 신경회로망에 기초한 아날로그 회로를 제안하였다.

1) 비조건부 시변 비선형 프로그래밍

시변 프로그래밍의 정의는 다음을 만족하는  $\bar{x}$ 를 찾는 것이다.

$$\bar{x} = \lim_{t \rightarrow \infty} \arg \min_x f(x, t) \quad (8)$$

수정된 급경사법은 다음과 같다 :

시변 함수가 다음의 부등식을 만족한다고 가정하자.

$$\left\| \left( \frac{\partial^2 f(x, t)}{\partial x \partial x^T} \right)^{-1} \right\| < \frac{1}{\mu}, \quad \forall x, t \quad (9)$$

이 때,  $\mu$ 는 양의 상수이다. 이 때의 알고리즘은

$$\dot{x}(t) = - \left( \alpha_0 + \frac{\partial f / \partial t}{\|\partial f / \partial x\|^2} \right) \frac{\partial f}{\partial x} \quad (10)$$

가 되고, 이 때  $\alpha_0$ 는 양의 상수이다. 여기서 만일  $df / dt$ 가 균일 연속(uniformly continuous)이라면,  $x(t)$ 는  $\bar{x}$ 로 수렴하게 된다. [8]

2) 조건부 시변 비선형 프로그래밍

제한 조건이 있는 시변 비선형 프로그래밍의 정의는 다음과 같은 문제에 있어서 적정해  $x = [x_1, \dots, x_n]^T$ 를 찾는 것이다 :

$$\lim_{t \rightarrow \infty} \arg \min_x f(x_1, \dots, x_n, t) \quad (11)$$

이 때의 제한 조건은 다음과 같다.

$$g_i(x) \leq 0, \dots, g_r(x) \leq 0 \quad (12)$$

위의 프로그램에 대한 알고리즘은 다음과 같이 주어진다.

$$I_i = \frac{\partial f}{\partial x_i} + s \sum_{j=1}^r i_j \frac{\partial g_j}{\partial x_i}$$

라고 하고, 이 때  $i_j = g'_j(x) = \max(0, g'_j(x))$  이라고 하자. 그러면,

$$\dot{x}_i(t) = -I_i - \frac{I_i}{\sum_{i=1}^n I_i^2} \frac{\partial f}{\partial t} \quad (13)$$

가 된다. (증명은 참고 문헌<sup>8)</sup> 참고)

### III. 시변 이상 신경회로망

시변 문제를 푸는 데 있어서, 위의 신경망들은 K-T의 최적화 조건을 고려하지 않았다. 이번 절에서는 정확한 적정해를 찾아 주고 최적성을 보장하는 이상 신경회로망의 장점을 이용하여 시변 문제에 대한 이상 알고리즘을 제안한다. 그리고 또한 이 알고리즘을 신경회로망의 학습에 적용할 수 있음을 보인다. 이를 위하여 앞 절에서와 마찬가지로 문제가 컨벡스하고 (9)의 부등식을 만족하며  $df / dt$ 는 균일 연속이라고 가정하자.

#### 1. 조건부 시변 이상 신경회로망

제한 조건이 있는 시변 프로그램은 다음과 같다.

시변 프로그램 (TVP) : 다음의 조건들을 만족하며  $f(x, t)$ 를 최소화한다.

$$g_i(x) \leq 0, \dots, g_r(x) \leq 0, h_i(x) = 0, \dots, h_m(x) = 0. \quad (14)$$

제안된 알고리즘은 다음의 수식으로 표현된다.

$$\begin{aligned} \dot{x}(t) &= -\alpha(x, t)G(x) \\ \alpha(x, t) &= \alpha_0 + \frac{\partial f / \partial t}{\|G(x)\|^2} \end{aligned} \quad (15)$$

여기서  $\alpha_0 > 0$ ,  $\alpha(x, t) \neq 0$  이고, ( $\alpha = 0$ 일 때는  $\alpha$ 는 이전의 0이 아닌 값을 유지한다.)

$$G(x) = \begin{cases} \nabla f + s \left[ \sum_{i=1}^r g'_i \nabla g_i + \sum_{j=1}^m h_j \nabla h_j \right] & (0 \leq t < t_i) \\ \nabla f + \sum_{i=1}^r \nabla g_i (s g'_i + \lambda) + \sum_{j=1}^m \nabla h_j (s h_j + \mu) & (t \geq t_i) \end{cases} \quad (16)$$

이다. 뒷식에서  $\lambda = \epsilon (s g'_i)$ ,  $\mu = \epsilon (s h_j)$  이다. 해의 최적성 여부를 판단하기 위해서  $x(t)$ 가 어떤 상수의 값으로 수렴했다고 가정하자. 만일  $\partial f / \partial t$ 가 0

으로 수렴했다면, 이는 다음과 같이 K-T의 최적화 조건을 만족하게 된다.

$$\nabla f + \sum_{i=1}^r \lambda_i \nabla g_i + \sum_{j=1}^m \mu_j \nabla h_j = 0.$$

그렇지 않고 만일  $\partial f / \partial t$ 가 수렴하지 않았을 때는 다음의 조건을 만족하지만 않는다면 마찬가지로 K-T의 최적화 조건을 만족하게 된다.

$$\left\| \nabla f + \sum_{i=1}^r \lambda_i \nabla g_i + \sum_{j=1}^m \mu_j \nabla h_j \right\|^2 = -\frac{1}{\alpha_0} \frac{\partial f}{\partial t}$$

그러나 우리가  $\alpha(x, t)$ 를 0이 되지 않도록 선택했기 때문에  $x(t)$ 는  $G(x)$ 가 0이 될 때에만 수렴하게 된다. 그러므로 제안된 시변 이상 신경회로망은  $x(t)$ 가 수렴하기만 하면 언제나 최적화 조건을 만족한다고 볼 수 있다.

제한 조건이 없는 시변 이상 신경회로망은 (14)식의 제한 조건식을 무시함으로써 얻어질 수 있다. 그러면 그 식은 Feng와 Zhao에 의해 제안된 시변 최적화 신경회로망과 같은 식이 된다. 다음절에서는 신경회로망의 학습에 위의 알고리즘을 적용시켜 보기로 한다.

#### 2. 신경회로망 학습에의 응용

다층 신경회로망의 학습에 있어서 우리가 조정할 수 있는 변수는 각 층에 있는 연결 가중 계수들이다. 그러므로 신경회로망의 학습 문제를 다음과 같이 수식화 할 수 있다 :

플랜트의 출력과 신경회로망의 출력간의 오차가 최소화되도록 각 층의 연결 가중 계수를 찾는다.

$$w^* = \min_w f(w, t) = \min_w \frac{1}{2} (y - \hat{y})^2 = \min_w \frac{1}{2} (F(u(t)) - NN[w, u(t)])^2 \quad (17)$$

여기서  $w$ 는 각 층 간의 연결 가중 계수이고,  $y$ 는 플랜트의 출력,  $\hat{y}$ 는 다층 신경회로망의 출력,  $F(\cdot)$ 는 플랜트의 기능을 나타내는 함수,  $NN[\cdot]$ 은 다층 신경망의 기능을 나타내는 함수이다.

신경회로망의 학습에 있어서 다음의 두 가지 경우를 생각해 본다.

첫째는,  $w_i (1 \leq i \leq N)$ 에 아무런 제한 조건이 없을 때인데, 이 때의 문제는 시변 비선형 문제의 비조건부 최적화 문제와 원리적으로 같다고 할 수 있다. 이 때의 알고리즘은 다음과 같다 :

$$\begin{aligned} \dot{w} &= -\alpha(w, t) \frac{\partial f}{\partial w} \\ \alpha(w, t) &= \alpha_0 + \frac{\partial f / \partial t}{\|\partial f / \partial w\|^2} \end{aligned} \quad (18)$$

여기서  $\alpha_0 > 0$ ,  $\alpha(w, t) \neq 0$  이다.  $\alpha = 0$  일 때는  $\alpha$  는 이전의 0이 아닌 값을 유지한다.

$\partial f / \partial w$  즉,  $\nabla_w f$ 와 같은 항은 기존의 오차 역전파 방법 (Error Back Propagation : EBP)에서의 경사치와 꼭 같다. 그리고  $\partial f / \partial t$  항은 다음과 같이 근사화할 수 있다.

$$\frac{\partial f(w, t)}{\partial t} \approx \frac{f(w, t) - f(w, t - \Delta t)}{\Delta t}$$

여기서  $f(w, t - \Delta t) = (y(t - \Delta t) - \hat{y}(t - \Delta t))^2$ 이고,  $\hat{y}(t - \Delta t)$ 는  $w(t)$ 로부터 얻어진다.

둘째로,  $w_i$ 가 제한되어 있을 때인데, 이는 실제적인 경우에 전자 회로로 결합 가중치를 구현할 때 크기는 어떤 제한된 값을 갖게 된다는 것을 의미한다. 즉, 다층 신경회로망을 하드웨어로 구현할 때, 각 층의 연결 가중 계수는 제한된 값만을 표현할 수 있기 때문에 이 알고리즘을 수식화할 때에 제한 조건을 사용하는 것이 필요한 것이다. 이 때의 알고리즘을 유도해 보자.

$w$ 의 제한 조건이  $|w_i| \leq M$  와 같이 주어졌다고 하면, (14)식으로부터

$$g_i(w) = w_i - M, \quad g_{i+N}(w) = -w_i - M \quad \text{이고}$$

$$\nabla g_i(w) = e_i, \quad \nabla g_{i+N}(w) = -e_i$$

여기서  $e_i$ 는  $R^N$ 에서의 단위 벡터이다. 예를 들면,  $e_1 = [1 \ 0 \ 0 \ \dots \ 0]^T, \dots, e_N = [0 \ 0 \ \dots \ 0 \ 1]^T$ 이다.

그러므로 알고리즘은 다음과 같이 정리될 수 있다.

$$\dot{w}(t) = -\alpha(w, t)G(w)$$

$$\alpha(w, t) = \alpha_0 + \frac{\partial f / \partial t}{\|G(w)\|^2} \quad (19)$$

여기서  $\alpha_0 > 0$ ,  $\alpha(w, t) \neq 0$  이고,

$$G(w) = [G_1(w), G_2(w) \dots G_N(w)]^T$$

첫 번째 상에서는

$$G_1(w) = \nabla_w f + s(\max(0, w_i - M) - \max(0, -w_i - M)) \quad (20)$$

그리고 두 번째 상에서는

$$G_2(w) = \nabla_w f + s(\max(0, w_i - M) - \max(0, -w_i - M)) + \lambda_i \quad (21)$$

가 된다. 여기서  $\lambda_i = \varepsilon s(\max(0, w_i - M) - \max(0, -w_i - M))$ 이다.

만일 비선형 제한 조건과 같은 다른 종류의 제한 조건을 가진다 하더라도 위의 시변 이상 신경회로망을 이용하면 간편하게 구현할 수 있다.

### V. 컴퓨터 시뮬레이션

이상 신경회로망과 시변 비선형 프로그래밍 신경회로망의 시뮬레이션 예는 참고 문헌들에 잘 나와 있으며, 여기서는 참고 문헌에 나와 있는 것과 비슷한 예를 가지고 제안된 시변 이상 신경회로망의 시뮬레이션 결과를 보여준다. 이 시뮬레이션에서 미분 방정식들은 Runge-Kutta의 4차 방법으로 계산하였으며, 시간 간격(1 스텝)은 0.01초에 해당한다.

#### 1. 시변 이상 신경회로망

이 절에서는, 제안된 시변 이상 신경회로망을 이용한 시변 최적화의 예를 보여준다. 두 개의 예는 모두 각각 시변 프로그래밍과 시불변 프로그래밍을 비교하여 보여준다.

1)에 1 : 이차식 프로그램 ( $QP_1$ )은 다음을 최소화하는 문제이다.<sup>[4]</sup>

$$f(x) = x_1^2 + x_2^2 + x_1x_2 + 3x_1 + 3x_2$$

이 때  $g_1$  에서  $g_6$ 의 제한 조건이 다음과 같다 :

$$g_1(x) = \frac{5}{12}x_1 - x_2 - \frac{35}{12} \leq 0$$

$$g_2(x) = \frac{5}{2}x_1 + x_2 - \frac{35}{2} \leq 0$$

$$g_3(x) = -x_1 - 5 \leq 0$$

$$g_4(x) = x_2 - 5 \leq 0$$

$$g_5(x) = x_1 - 3 \leq 0$$

$$g_6(x) = -x_1 + 3 \leq 0.$$

그림 1 은 ( $QP_1$ ) 문제에 있어서 4개의 서로 다른 초기치를 주었을 때에  $x_1$ 과  $x_2$ 의 궤적을 나타내고 있다. 원하는 정확한 해든  $x = (3.0000, -1.6667)$ 이다. 첫 번째 상에서는 해가 정확한 값이 아닌  $x = (2.8428, -1.7779)$ 로 수렴했으나, 두 번째 상에서는 그 평형점이 정확한 값인  $x = (3.0000, -1.6667)$ 로 옮겨갔음을 알 수 있다. 첫 번째 상에서는 100 스텝 (1.0 초)만에  $x_1$ 과  $x_2$ 가 수렴했고, 두 번째 상에서는 4,300 스텝 (43.0 초)만에 정확한 해로 수렴했다.

제안된 알고리즘의 장점을 살펴보기 위해 다음과 같은 시변 프로그래밍을 생각해본다. 시변 이차식 프로그램 ( $TVQP_1$ ):

$$f(x, t) = (3 + e^{-t} \sin t)(x_1 + x_2) + (1 + e^{-t})(x_1 + x_2)^2 - x_1 x_2$$

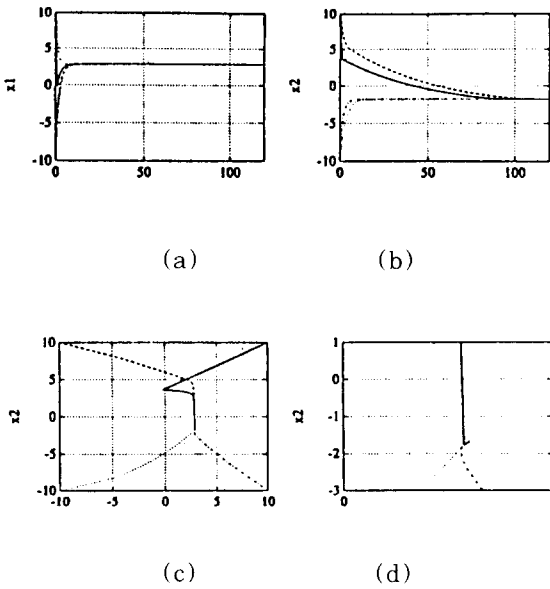


그림 1. 예 1 :  $s=50$ ,  $\epsilon = 0.2$ ,  $\bar{x} = (3.000, -1.667)$  일 때의  $(QP_1)$  의 궤적  
 (a)  $x_1$ 의 궤적 (b)  $x_2$ 의 궤적  
 (c)  $x_0 = (10, 10), (-10, 10), (10, -10), (-10, -10)$  일 때 1상에서의  $x_1$ 과  $x_2$ 의 궤적  
 (d) 2상에서의  $x_1$ 과  $x_2$ 의 궤적.

Fig. 1. Example 1 : Trajectories for  $(QP_1)$  with  $s=50$  and  $\epsilon = 0.2$ ,  $\bar{x} = (3.000, -1.667)$ .  
 (a) trajectory of  $x_1$ , (b) trajectory of  $x_2$ ,  
 (c) trajectory of  $x_1$  and  $x_2$  in phase-1,  $x_0 = (10, 10), (-10, 10), (10, -10), (-10, -10)$ ,  
 (d) trajectory of  $x_1$  and  $x_2$  in phase-2.

이 때의 제한 조건은  $(QP_1)$ 과 같고, 그림 2 에 그 결과를 나타내었다.

그림 2 에서는 그림 1 에서와 같은 방식으로 정확한 해로 수렴한다.  $f(x, t)$ 가 첫번째 상에서는 약 120 스텝 (1.2 초) 정도에서 수렴하고, 두 번째 상에서는 약 4,300 스텝 (43.0 초)만에 정확한 해로 수렴하였다. 시변임에도 불구하고  $(QP_1)$ 에서와 거의 같은 시간이 걸림을 알 수 있다. 이는 시변 이상 신경회로망이 시변하는 궤적을 따라서 최소점을 찾아갈 수 있음을 보여준다. 이 회로망은 매순간의 시간 스텝마다 다르게 주어지는 함수의 최소점을 찾아가려는 방향으로 동작하게 된다.

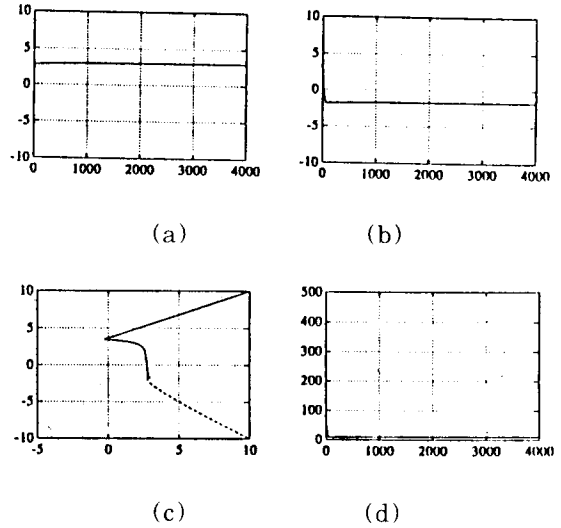


그림 2. 예 1 :  $s=50$ ,  $\epsilon = 0.2$ ,  $\bar{x} = (3.000, -1.667)$  일 때의  $(TV QP_1)$  의 궤적  
 (a)  $x_1$ 의 궤적. (b)  $x_2$ 의 궤적  
 (c)  $x_0 = (10, 10), (10, -10)$ 일 때  $x_1$ 과  $x_2$ 의 궤적  
 (d)  $f(x, t)$ 의 궤적.

Fig. 2. Example 1 : Trajectories for  $(TV QP_1)$  with  $s=50$  and  $\epsilon = 0.2$ ,  $\bar{x} = (3.000, -1.667)$ .  
 (a) trajectory of  $x_1$ , (b) trajectory of  $x_2$ ,  
 (c) trajectory of  $x_1$  and  $x_2$ ,  $x_0 = (10, 10), (10, -10)$ ,  
 (d) plot of  $f(x, t)$ .

2) 예 2 : 비선형 프로그램  $(NP_1)$  은 다음의 함수를 최소화하는 것이다. [4]

$$f(x) = x_1^2 + (x_2 - 1)^2$$

이 때의 제한 조건은 다음과 같이 주어진다.

$$g(x) = x_2 - x_1^2 = 0$$

그림 3은  $(NP_1)$  문제에 있어서 서로 다른 두 개의 초기치에 따른  $x_1$ 과  $x_2$ 의 궤적을 보여준다. 정확한 해는  $\bar{x} = (\pm 0.707, 0.500)$  이다. 그림 3은 그림 1 과 같은 방식으로 설명되어 질 수 있다. 첫번째 상에서는 550 스텝 (5.5 초)만에 수렴하였고, 두번째 상에서는 21,800 스텝 (218 초)만에 수렴하였다.

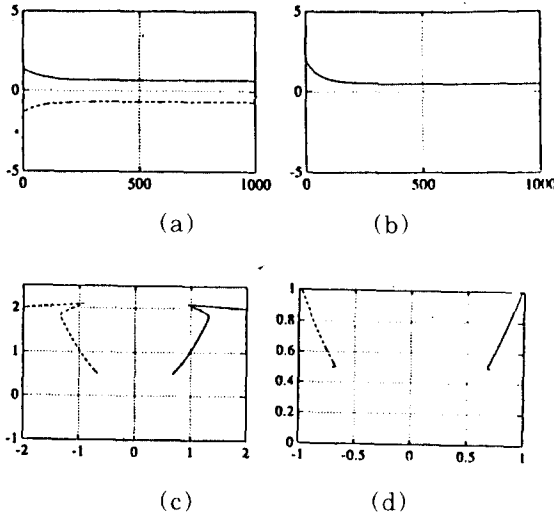


그림 3. 예 2 :  $s=50, \epsilon=0.2, \bar{x}=(\pm 0.707, 0.500)$  일 때의  $(NP_1)$  의 궤적  
 (a)  $x_1$ 의 궤적 (b)  $x_2$ 의 궤적  
 (c)  $x_0=(2, 2), (-2, 2)$  일 때 1상에서의  $x_1$ 과  $x_2$ 의 궤적  
 (d) 2상에서의  $x_1$ 과  $x_2$ 의 궤적

Fig. 3. Example 2 : Trajectories for  $(NP_1)$  with  $s=50$  and  $\epsilon=0.2, \bar{x}=(\pm 0.707, 0.500)$ .  
 (a) trajectory of  $x_1$ , (b) trajectory of  $x_2$ ,  
 (c) trajectory of  $x_1$  and  $x_2$  in phase-1,  $x_0=(2, 2), (-2, 2)$ ,  
 (d) trajectory of  $x_1$  and  $x_2$  in phase-2.

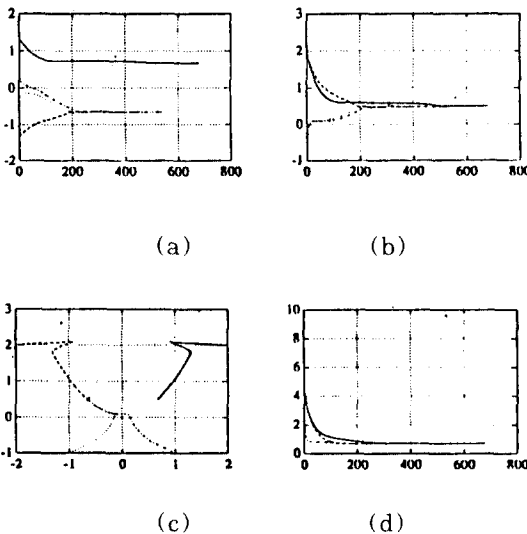


그림 4. 예 2 :  $s=50, \epsilon=0.2, \bar{x}=(\pm 0.707, 0.500)$  일 때의  $(TVNP_1)$  의 궤적  
 (a)  $x_1$ 의 궤적 (b)  $x_2$ 의 궤적  
 (c)  $x_0=(2, 2), (-2, 2), (1, -1), (-1, -1)$  일 때  $x_1$ 과  $x_2$ 의 궤적 (d)  $f(x, t)$ 의 궤적.

Fig. 4. Example 2 : Trajectories for  $(TVNP_1)$  with  $s=50$  and  $\epsilon=0.2, \bar{x}=(\pm 0.707, 0.500)$ .  
 (a) trajectory of  $x_1$ , (b) trajectory of  $x_2$ ,  
 (c) trajectory of  $x_1$  and  $x_2, x_0=(2, 2), (-2, 2), (1, -1), (-1, -1)$ , (d) plot of  $f(x, t)$ .

그림 4 는 다음과 같은 시변 비선형 프로그램  $(TVNP_1)$  이다 :

$$f(x, t) = e^{-t} \sin t \cdot x_1 + (1 + e^{-t})x_1^2 + (x_2 - 1)^2$$

이 때의 제한 조건은  $(NP_1)$ 에서와 같다.

그림 4 에서는  $f(x, t)$ 가 시변 함수이기 때문에  $x_1$ 과  $x_2$ 의 궤적이 부드럽지 못하고 약간 거칠게 변화함을 보여준다. 첫 번째 상에서는 160 스텝 (1.6 초)만에 수렴하였고, 두 번째 상에서는 700 스텝 (7.0 초)만에 정확한 해로 수렴하였다. 이  $(TVNP_1)$  문제의 수렴 속도는  $(NP_1)$  문제보다 더 빠름을 알 수 있는데, 그 이유는 비선형적인 특징이 비선형 프로그래밍에서는 더 잘 최소점을 찾도록 궤적을 만들어 주기 때문이다.

위의 시뮬레이션을 통하여 시변 문제들은 시불변 문제들에서와 똑같은 해로 수렴함을 알 수 있었고, 더불어  $f(x, t)$ 는 시간이 지남에 따라 최소화된다 것을 보였다. 위의 시뮬레이션들로부터 시변 이상 신경회로망의 유용성이 분명히 보여졌다.

## 2. 신경회로망 학습에의 적용

### 1) 예 3 : 함수 모사

함수 모사를 위해 다음과 같은 플랜트 함수를 고려해 보자.

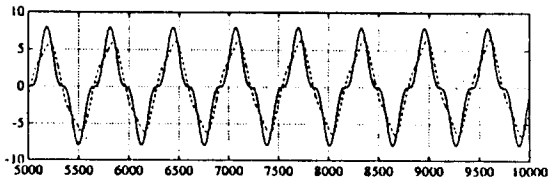
$$F(u(t)) = u(t)^3 \quad u(t) = 2 \sin t$$

이 때 신경회로망은  $N_{1,4,2,1}$  즉, 입력 노드 1개, 각 각 4개와 2개의 노드를 가진 2개의 은닉층, 그리고 1개의 출력 노드로 구성된 3층 구조이다.

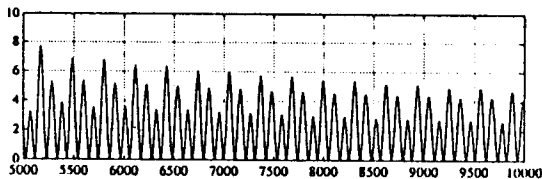
그림 5 에서 신경회로망 학습 계수  $\lambda (= \alpha_0)$ 는 1로 고정시켰고, 기존의 EBP 방법과 제한된 시변 이상 신경망의 두 가지 기법을 비교하였다. EBP 방법

을 이용한 그림 5(a) 에서 오차는 시간에 따라 감소하고는 있지만, 제안된 기법을 이용한 그림 5(c) 보다는 성능이 좋지 못하다. 이 결과로 보았을 때, 오차 함수의 시간 변화를 고려했을 때에 더욱 빠른 수렴 속도를 보여준다는 사실을 알 수 있다.

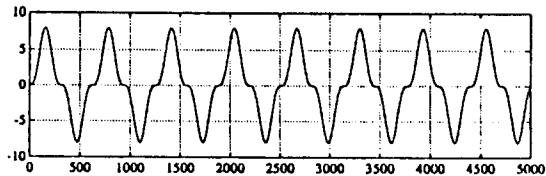
(b) plot of error function in (a). (c) outputs of plant and identification model with proposed method ( $\lambda =1$ ). (d) plot of error function in (c).



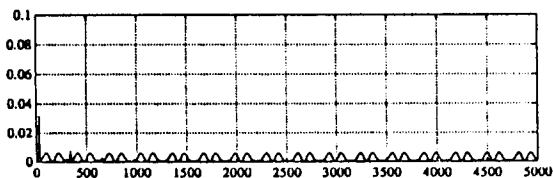
(a)



(b)



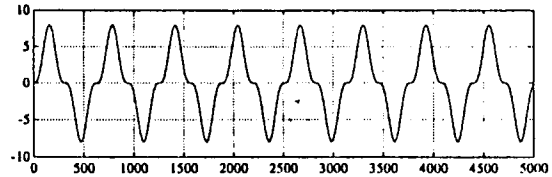
(c)



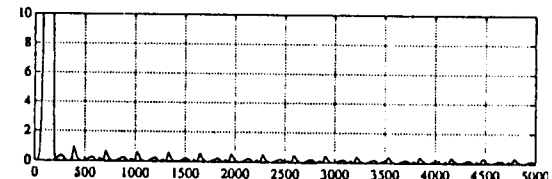
(d)

그림 5. 예 3 : 함수 모사(a) 플랜트 출력과 동정 모델의 비교 (기존 EBP 방법,  $\lambda =1$ ) (b) (a)에서의 오차 함수 (c) 플랜트 출력과 동정 모델의 비교 (제안된 방법,  $\lambda =1$ ) (d) (c)에서의 오차 함수

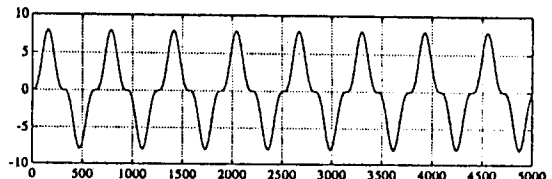
Fig. 5. Example 3 : Function learning (a) outputs of plant and identification model with conventional EBP ( $\lambda =1$ ).



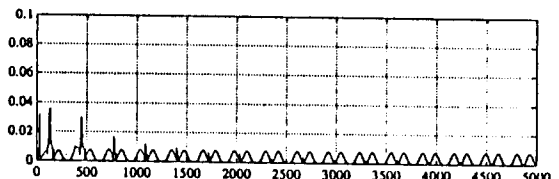
(a)



(b)



(c)



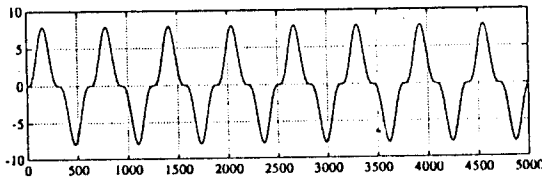
(d)

그림 6. 예 3 : 함수 모사(a) 플랜트 출력과 동정 모델의 비교 (기존 EBP 방법,  $\lambda =20$ ) (b) (a)에서의 오차 함수 (c) 플랜트 출력과 동정 모델의 비교 (제안된 방법,  $\lambda =20$ ) (d) (c)에서의 오차 함수

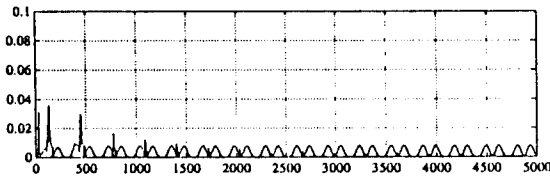
Fig. 6. Example 3 : Function learning (a) outputs of plant and identification model with conventional EBP ( $\lambda =20$ ). (b) plot of error function in (a). (c) outputs of plant and identification model with proposed method ( $\lambda =20$ ). (d) plot of error function in (c).



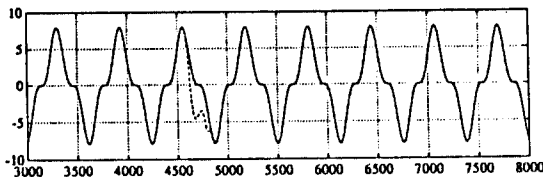
그림 6에서는 신경회로망의 학습 계수  $\lambda = 20$  일 때 두 알고리즘 즉, 기존 EBP 방법과 시변 이상 신경망의 비교를 보여준다. 기존 EBP 방법을 사용한 그림 6(a)는 그림 5(a) 보다는 좋은 결과를 보여주지만 제안된 알고리즘인 그림 6(c) 보다는 좋지 않다. 이 결과들로부터 기존 EBP 방법은 학습 계수  $\lambda$ 의 변화에 아주 민감하나, 제안된 알고리즘은  $\lambda$ 의 선택에 별로 상관하지 않음을 알 수 있는데 그 주된 이유는 오차가 빨리 변할 때에는 연결 가중 계수의 수정이  $\partial f / \partial t$  항에 주로 의존하기 때문이다. 그러므로 제안된 방법을 이용할 때에는 학습 계수의 선택에 별로 신경 쓰지 않아도 되고, 온라인 수정을 할 때에도 좋은 성능을 보일 수 있음을 알 수 있다.



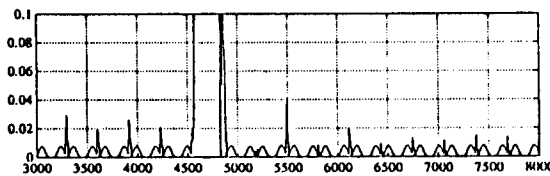
(a)



(b)



(c)



(d)

그림 7. 예 3 : 연결 강도에 제한 조건이 있을 때 (a)  $|w_i| \leq 5.0$  이고  $s=1$  일 때 플랜트 출력과 동정 모델의 비교 (b) (a)에서의 오차 함수 (c)  $|w_i| \leq 5.0$  이고  $s=10$  일 때 플랜트 출력과 동정 모델의 비교 (d) (c)에서의 오차 함수

Fig. 7. Example 3 : Weights with constraints(a) outputs of plant and identification model with proposed method when  $|w_i| \leq 5.0$  and  $s=1$ , (b) plot of error function in (a), (c) outputs of plant and identification model with proposed method when  $|w_i| \leq 5.0$  and  $s=10$ , (d) plot of error function in (c).

그림 7은 연결 가중 계수에 제한 조건이 있을 때의 결과를 보여준다. 여기서  $w$ 의 제한 조건은  $|w_i| \leq 5.0$  과 같이 주었다. 이상 회로망에서 두 개의 다른 페널티 계수  $s$ 를 주었을 때에 약간씩 다른 특성이 언어짐을 그림에서 볼 수 있다.  $s = 10$  으로 하였을 때, 오차는  $k \approx 4500$  스텝에서 갑자기 커졌고  $k \approx 5000$  스텝에서 줄어듦을 볼 수 있다. 이것은 더 큰 페널티 계수가 주어졌을 때 회로가 극부 최소점을 빠져 나가게 되어 얼마간의 시간이 지난 후에 다른 최소점을 찾게 되기 때문이다. 서로 다른 상황에서의 연결 가중 계수의 크기가 표 1,에 비교되어 있다. 그림 6(a)와 (c)에서의 연결 가중 계수는 제한되어 있지 않은 반면에, 그림 7에서의 연결 가중 계수는  $[-5, 5]$ 의 값들로 제한되어 있다.

표 1. 각 층에서의 연결 가중 계수의 비교  
Table 1. Comparison of the weights in each layer.

	Fig. 6(a)	Fig. 6(c)	Fig. 7(a)	Fig. 7(c)
$w_{10}$	-1.582019	0.843440	-0.182791	-0.005353
	0.661660	0.015609	4.424918	4.407977
	-3.543147	-3.257611	0.079929	-0.166597
	0.793852	-1.001976	-4.415043	-4.357703
	0.252792	-0.139354	-0.170542	-0.081049
	0.339848	0.086916	4.399084	4.533397
	-4.004342	-7.983026	-0.212952	-0.339230
	1.318211	-0.689535	-4.389630	-4.340104
$w_{21}$	-1.758491	-0.074925	4.587071	3.947982
	-4.114518	-1.874860	-4.596229	-4.447386
	0.337965	0.162614	3.588896	2.780646
	-6.062905	-3.868206	-4.623275	-4.468673
	-0.422846	0.592425	4.496541	4.594921
	-2.562591	-1.664099	-4.503751	-4.609183
	-0.245103	-0.324201	2.715618	3.915390
-2.955832	-2.668714	-4.486100	-4.625654	
$w_{32}$	8.933095	2.297673	-0.831354	-1.076667
	4.043088	1.533927	-0.976419	-0.800034

2) 예 4 : 동적 플랜트의 동정

여기서는 제한된 알고리즘을 동적 플랜트의 동정에 이용한다. 이 예에서 사용되는 플랜트는 다음과 같은 형태이다. [9] 이 모델은 참고 문헌 [9] 에 의하면 모델 IV로 알려져 있다.

$$y_p(k+1) = f[y_p(k), y_p(k-1), y_p(k-2), u(k), u(k-1)]$$

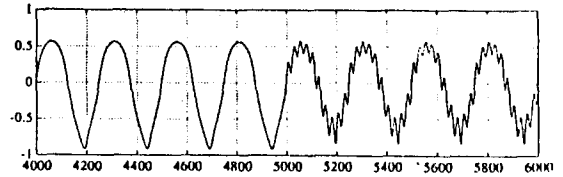
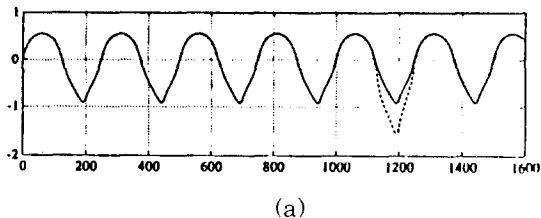
이 때 미지의 함수  $f$  는 다음의 형태이다.

$$f[x_1, x_2, x_3, x_4, x_5] = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_1^2 + x_2^2}$$

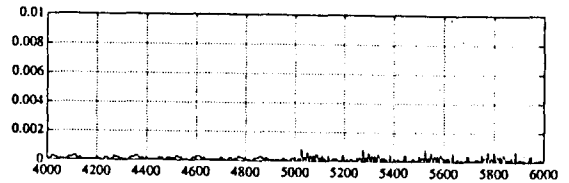
여기서는 참고 문헌에 나온 신경망 구조( $N_{5,20,10,1}^3$ )보다 간단한 형태인  $N_{5,10,10,1}^3$  구조를 이용하였다.

그림 8(a) 는 입력이  $u(k) = \sin(2\pi k/250)$  으로 주어졌을 때 온라인 학습의 결과를 보여준다. 첫번째 사이클에서도 잘 적응함을 알 수 있다. 약 1,200 스텝에서 더 좋은 최소점을 찾기 위해 국부 최소점을 빠져나감을 알 수 있다. 수렴하는 데에는 약 2,000 스텝(20 초)만이 걸렸다. 그림 8(b)에서는 5,000 스텝에서 입력이  $u(k) = 0.8\sin(2\pi k/250) + 0.2\sin(2\pi k/250)$  으로 변했을 때의 결과를 보여준다. 입력이 갑자기 변화했을 때에도 신경망은 플랜트의 출력을 곧장 따라가게 된다.

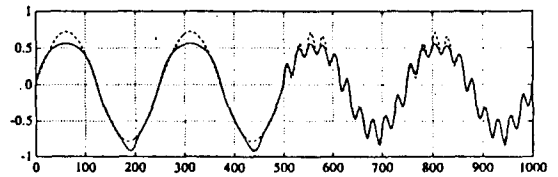
학습이 끝난 후에 온라인 수정을 하지 않았을 때의 신경망의 출력 결과가 그림 8(d)에 나타나 있다. 참고 문헌 [9] 의 그림 16과 비교를 하였을 때, 거의 같은 결과를 보임을 알 수 있다. 그러나 [9] 에서는 약 100,000 스텝의 학습을 거친 반면, 본 논문에서는 약 10,000 스텝의 학습만을 온라인으로 하였다. 또한, [9] 에서는 적절한 학습 계수를 선택해 주어야 하는 반면에 여기서는 학습 계수의 선택에 있어서도 신경 쓸 필요가 없었다. 노드의 갯수가 많아서 연결 가중 계수의 값을 분산시키는 연결들이 신경회로망 자체안에 많이 있게 되어 연결 가중 계수가  $|w_i|$  2.0 으로 제한된 값으로 수렴한다. 이 때문에 여기에서는 두 번째 상을 굳이 이용하지 않아도 되었다. 이 예들은 동적 플랜트의 동정에 있어서의 온라인 학습의 가능성과 유용성을 보여준다고 할 수 있다.



(b)



(c)



(d)

그림 8. 예 4 : 동적 플랜트의 학습 (a) 플랜트 출력과 동정 모델의 비교 (b)  $k = 5000$  에서 입력이 변화했을 때 플랜트 출력과 동정 모델의 비교 (c) (b)에서의 오차 함수 (d) 학습후에 연결 강도를 수정하지 않았을 때 플랜트 출력과 동정 모델의 비교

Fig. 8. Example 4 : Dynamic plant learning (a) outputs of plant and identification model, (b) outputs of plant and identification model when the input changed at  $k = 5000$ , (c) plot of error function in (b), (d) outputs of plant and identification model when no adaptation is carried out after learning.

V. 결론

본 논문에서는 시변하는 비선형 프로그래밍 문제를 푸는데 있어서 시변 이상 신경회로망을 이용할 수 있음을 컴퓨터 시뮬레이션을 통하여 보여주었다. 이 회로는 i) 문제가 컨벡스이어야 한다, ii) 시변 회로망

의 부등식을 만족하여야 한다. 그리고 iii) 최소화하려는 함수의 시간 미분이 균일 연속이어야 한다는 가정하에서 정확하고 최적한 해를 찾을 수 있었다. 컨벡스한 문제가 아닌 경우에도 이 알고리즘은 총체적이거나 국부적일 수도 있는 해를 찾는데 쓰일 수 있음이 보여졌다. 시변 이상 신경회로망을 이용한 다층 신경회로망은 좋은 성능의 학습 결과를 보여 주었다. 제안된 알고리즘은 다음과 같은 특성을 가짐을 알 수 있었다. 즉, 온라인 학습에서 좋은 적응성을 가졌고, 학습 계수의 선택에 있어서 기존의 EBP 방법보다 덜 민감하다는 것이다. 그렇기 때문에 온라인 적응이 필요한 제어 문제에 곧바로 응용 가능하다. 또한, 본 알고리즘의 큰 장점은 이상 신경망을 사용하기 때문에 연결 가중 계수에 여러 가지 제한 조건이 있을 때에도 학습이 가능하다는 점이다.

#### 參 考 文 獻

- [1] D.W. Tank, J. J. Hopfield, "Simple 'Neural' optimization networks: An A/D converter signal decision circuit, and a linear programming circuit." *IEEE Trans. circuits and syst.*, vol. CAS-33, pp.533-541, May 1986.
- [2] L. O. Chua, G.-N. Lin, "Nonlinear programming without computation." *IEEE Trans. Circuits Syst.*, vol. CAS-31, pp.182-188, Feb. 1984.
- [3] M. P. Kennedy, L. O. Chua, "Neural networks for nonlinear programming." *IEEE Trans. circuits and syst.*, vol. 35, no.5, pp.554-562, May 1988.
- [4] C. Y. Maa, M. A. Shanblatt, "A Two-phase optimization neural network." *IEEE Trans. neural networks*, vol.3, no.6, pp.1003-1009, Nov. 1992.
- [5] S. Zhang, X. Zhu, L-H. Zou, "Secondorder neural nets for constrained optimization." *IEEE Trans. circuits and syst.*, vol.3, no.6, pp. 1021-1024, Nov. 1992.
- [6] S. Zhang, A. G. Constantinides, "Lagrange programming neural networks." *IEEE Trans. circuits and syst.*, vol.39, no.7, pp.441-452, July 1992.
- [7] A. Cichocki, R. Unbehauen, "Neural networks for solving systems of linear equations and related problems." *IEEE Trans. Circuits Syst.*, vol.39, no.2, pp. 124-138, Feb. 1992.
- [8] C. B. Feng, Y. Zhao, "Time-varying nonlinear programming and its realization via neural networks." *Proc. ACC*, pp.978-982, 1992.
- [9] K. S. Narendra, K. Parthasarathy, "Identification and control of dynamical systems using neural networks." *IEEE Trans. neural networks*, vol.1, no.1, pp.4-26, March 1990.
- [10] D. G. Luenberger, *Introduction to linear and nonlinear programming*, Addison-Wesley, 1973.

#### 著 者 紹 介



明 鉉(正會員)

1970年 8月 15日生. 1992年 2月 한국과학기술원 전기 및 전자공학과 공학사 학위 취득. 1994년 2월 한국과학기술원 전기 및 전자공학과 공학석사 학위 취득. 1994년 ~ 현재 한국과학기술원 전기 및 전자공학과 박사과정 재학중



金 鍾 煥(正會員)

1957年 7月 13日生. 1981年 서울대 공대 전자공학과 졸업. 1983년 동 대학원 전자공학과 졸업(석사). 1987년 동 대학원 전자공학과 졸업(공학). 1992년 8월 ~ 1993년 8월 미국 Purdue 대학 교환교수. 현재 한국과학기술원 전기 및 전자 공학과 부교수. 1988년 제 3 회 춘강 학술상 수상