

Bacterial Foraging Algorithm을 이용한 Extreme Learning Machine의 파라미터 최적화

Parameter Optimization of Extreme Learning Machine Using Bacterial Foraging Algorithm

조재훈* · 이대종** · 전명근**

Jae-Hoon* Cho, Dae-Jong Lee**, Myung-Geun Chun**

* 충북대학교 전기전자컴퓨터공학부

** 충북대학교 BK21 충북정보기술사업단

요 약

최근 단일 은닉층을 갖는 전방향 신경회로망 구조로, 기존의 경사 기반 학습알고리즘들보다 학습 속도가 매우 우수한 ELM(Extreme Learning Machine)이 제안되었다. ELM 알고리즘은 입력 가중치들과 은닉 바이어스들의 초기값을 무작위로 선택하고 출력 가중치들은 Moore-Penrose(MP) 일반화된 역행렬 방법을 통하여 구해진다. 그러나 입력 가중치들과 은닉층 바이어스들의 초기값 선택이 어렵다는 단점을 갖고 있다. 본 논문에서는 최적화 알고리즘 중 박테리아 생존(Bacterial Foraging)알고리즘의 수정된 구조를 이용하여 ELM의 초기 입력 가중치들과 은닉층 바이어스들을 선택하는 개선된 방법을 제안하였다. 실험을 통하여 제안된 알고리즘이 많은 입력 데이터를 가지는 문제들에 대하여 성능이 우수함을 보였다.

키워드 : 전방향 신경회로망, Extreme Learning Machine, Moore-Penrose 역행렬, 박테리아 생존 알고리즘.

Abstract

Recently, Extreme learning machine(ELM), a novel learning algorithm which is much faster than conventional gradient-based learning algorithm, was proposed for single-hidden-layer feedforward neural networks. The initial input weights and hidden biases of ELM are usually randomly chosen, and the output weights are analytically determined by using Moore-Penrose(MP) generalized inverse. But it has the difficulties to choose initial input weights and hidden biases. In this paper, an advanced method using the bacterial foraging algorithm to adjust the input weights and hidden biases is proposed. Experimental results show that this method can achieve better performance for problems having higher dimension than others.

Key Words : Single-hidden-layer feedforward neural networks, Exterme Learning Machine, Moore Penrose generalized inverse, Bacterial foraging algorithm.

1. 서 론

신경회로망은 인간두뇌의 신경세포를 모방한 개념으로 과거에 수집된 데이터로부터 반복적인 학습과정을 통하여 데이터 내에 내재되어 있는 패턴을 찾아내는 모델링 기법으로, 예측, 군집, 분류 문제 등 다양한 분야에 적용되고 있다 [1]. 신경회로망은 학습방법에 따라서 지도학습(Supervised learning)과 비지도학습(Unsupervised learning)으로 나누어지며,

구조 및 학습법칙에 따라 다양한 방법들이 있으며, 현재까지도 성능이 우수한 알고리즘들이 제안되어 적용되고 있다. 이외에도 인간의 의사판단능력을 모사한 퍼지개념을 도입하여 신경회로망과 퍼지시스템이 지닌 장점을 융합한 뉴로-퍼지 시스템도 주목받고 있다[2-4]. 이렇듯 여러 종류의 신경회로망 중에서도 학습 알고리즘이 용이하고 구현이 쉬운 장점을 지닌 다층 신경회로망(MLPNs:Multi-layer Neural Networks)이 널리 적용되고 있으며, 패턴 인식뿐만 아니라 함수 근사화 문제에 널리 적용되어지고 있다. David Rumelhart에 의해 제안된 다층신경망은 출력층의 목표값과 실제 출력값 사이의 오차를 계산하여 가중치를 조정하는 오류 역전파 알고리즘(BP:Back- Propagation)에 의해 학습이 수행된다 [5]. 현재에도 오류역전파와 같은 고전적인 경사기반 학습은 다층전방향신경망의 학습에 광범위하게 이용되고 있다 [6][7]. 그러나 경사기반 학습방법들은 복잡한 논리나 학습이 가능하지만 수

+ : 교신저자

접수일자 : 2006년 9월 18일

완료일자 : 2007년 11월 22일

감사의 글 : 본 연구는 산업자원부의 지원에 의하여 기초전력연구원(R-2007-2-046) 주관으로 수행된 과제임.

학적으로 전역 최적해에 수렴한다는 보장이 없고, 신경회로망의 활성화 함수가 미분가능해야 한다는 단점을 가지고 있다. 또한, 좋은 성능을 갖는 신경회로망을 구현하기 위해서는 반복적인 실험에 의해 최적의 파라미터를 찾아야 하기 때문에 수렴속도 면에서 다른 학습 알고리즘에 비해 느리다는 단점이 있다.

이러한 다층신경망의 문제점을 해결하기 위하여 Huang은 학습시간이 월등히 향상된 ELM(Extreme Learning Machine)을 제안하였다 [8][9]. 제안된 ELM은 단일 은닉층을 갖는 전방향 신경회로망의 구조로 되어 있으며, 기존의 신경회로망과 같이 입력층, 은닉층, 출력층으로 이루어져 있다. ELM의 학습 방법은 입력층과 은닉층 사이에 연결된 가중치를 임의로 선택한 후, 은닉층과 출력층 사이의 가중치는 모어-페로스의 일반화된 역행렬(Moore-Penrose generalized inverse matrix)을 이용하여 최적의 값을 찾는다. 이 점이 기존 방식과의 주된 차이점으로서, 기존의 오류역전파 알고리즘은 전방향과 후방향으로 최적의 해를 찾기 위해 반복적인 연산을 수행하지만, ELM은 후방향을 고려하는 것 없이 한번의 전방향 단계에서 학습이 종료됨으로 학습시간은 다층신경망에 비해 월등히 우수하다. 수렴속도 뿐만 아니라 ELM은 오류역전파 알고리즘이 미리 정해야 할 변수들인 학습의 반복횟수, 학습률, 학습 종료 조건등과 같은 변수들을 고려하지 않아도 된다 [9].

그러나 ELM 방식도 기존의 신경회로망과 동일하게 은닉층의 뉴런의 수, 입력과 은닉층에 연결된 가중치의 초기값을 설정해 주어야 하며, 설정된 값에 따라 시스템 성능에 차이를 보임으로 최적의 뉴런 수 및 가중치를 구하는 문제는 매우 중요하다. 이러한 문제를 해결하기 위해 Zhu는 차분 진화(differential evolution)기법을 ELM에 적용하여 입력층과 은닉층간의 최적의 가중치를 찾는 접근법을 시도하였다 [10]. 최적의 가중치를 찾는 문제는 단순히 ELM의 문제뿐만 아니라 다층 신경망 측면에서도 해결해야 할 중요한 문제이다. 다층 신경망에서는 자연선택과 진화과정의 개념을 가진 병렬적이고 전역적인 탐색 알고리즘인 유전 알고리즘을 적용하여 해결하고자 하는 시도가 진행되고 있다 [11][12]. 그러나 유전자 알고리즘의 탐색은 임의의 초기집단에서 초우량 개체가 출현하면 근사최적해로 조기 수렴하는 단점이 있기 때문에 방향성 탐색이 가능한 새로운 알고리즘들이 개발되어지고 있다[13][14].

최적화의 한 방법인 박테리아 생존 알고리즘은 박테리아의 생존을 위한 먹이찾기 행동을 모방한 알고리즘으로서 최근 최적화 분야에서 많이 연구되어지고 있다. 박테리아를 모방한 최적화 방법은 Bremermann에 의해 제안된 박테리아의 주화성 반응(chemotaxis)을 이용한 방법과 최근 Passino에 의해 제안된 박테리아 생존 알고리즘이 있다 [13][14]. Bremermann에 의해 제안된 방법은 박테리아의 주화성 반응을 기반으로 하고 Passino가 제안한 방법은 박테리아의 주화성 반응, 재생산, 제거-분산, 군집(Swarming) 반응들을 포함한다.

본 논문에서는 ELM의 초기 입력 가중치들과 은닉층 바이어스들을 동조하기 위하여 박테리아 생존 알고리즘의 적용방법 및 유용성을 다루고자 한다. 제안된 방법은 수정된 박테리아의 구조를 이용하여 ELM의 초기 가중치 및 바이어스를 박테리아의 초기 위치로 생성한 후 임의의 방향으로 탐색을 시도하여 최적의 가중치와 바이어스들을 찾는다. 제안방법을 함수 근사화 문제와 패턴 분류 문제에 적용하여 이전의 논문과 비교함으로써 타당성과 유용성을 보이고자 한다.

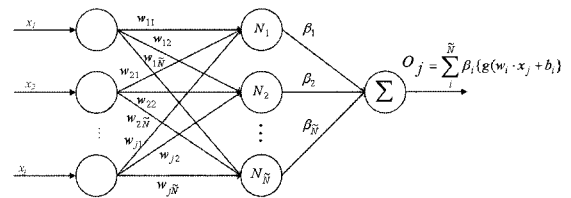
논문의 구성은 2장에서 ELM 알고리즘을 설명하고, 3장에

서는 박테리아 생존 알고리즘을 설명한다. 4장에서는 제안된 알고리즘과 관련한 실험 및 고찰을 설명하고, 마지막 5장에서 결론을 맺는다.

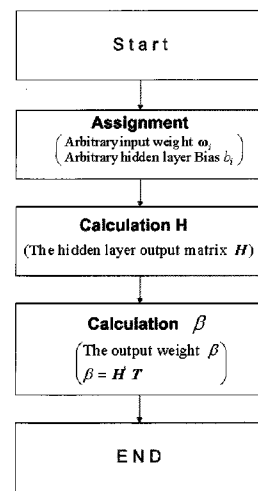
2. Extreme Learning Machine 알고리즘

다층 신경회로망에 사용되는 경사 기반 학습 방법은 기본적으로 입력과 은닉층, 은닉층과 출력층에 관련된 모든 가중치들과 바이어스들의 학습이 필요하며, 신경회로망의 성능을 높이기 위하여 파라미터들의 많은 반복 학습이 필요하다. 또한 더 좋은 일반화 성능을 위해서는 가중치들의 값들을 작게 조절할 필요가 있다[11]. 이런 과정들은 최적해가 아닌 지역해에 수렴할 우려가 높으며, 많은 학습 시간이 필요함으로 실제 시스템에 적용 하는 데는 문제가 있다. ELM 알고리즘은 고전적 학습 방법들의 단점과 빠른 학습시간을 위하여 입력 가중치와 뉴런들의 바이어스들의 학습을 하지 않고 오직 출력 가중치들을 놔 최소화해(norm least-squares solution)와 모어-페로스의 일반화된 역행렬을 이용하여 구한다.

그림 1에서는 ELM의 구조와 학습과정을 나타냈다. 그림 1에서 보는 바와 같이 ELM은 한 개의 은닉층을 갖는 SLFN(Single hidden layer feedforward networks)의 구조로 되어 있으며, 출력층에 사용되는 활성화 함수는 선형함수를 사용한다.



(a) ELM의 구조



(b) ELM의 학습 순서

그림 1. ELM의 구조 및 학습

Fig. 1 Structure and learning process in ELM

ELM 알고리즘의 주요 장점을 열거하면 다음과 같다. ① 그림 1에서 보는 바와 같이 반복학습을 하지 않기 때문에 다른 학습 방법들에 비해 속도가 매우 빠르다. ② 경사 기반

학습 방법들은 오직 출력 오차를 최소화 하기 때문에 가중치들의 크기를 고려하지 않지만 ELM 알고리즘은 가장 작은 출력 오차와 노름최소자승 방법을 이용하여 가장 작은 가중치들까지 구하기 때문에 일반화 성능이 우수하다. ③ 경사 기반 학습 방법은 오직 미분 가능한 활성화 함수를 사용하지만 ELM 알고리즘은 미분가능하지 않은 활성화 함수를 가지는 한 개의 은닉층으로 구성된 신경회로망을 학습할 수 있다. ④ 경사 기반 학습 방법이 국부최소, 부적절한 학습률의 효과 그리고 over-fitting과 같은 문제점이 있는 반면 ELM은 한번의 학습으로 해에 도달하는 경향이 있다.

ELM의 학습과정을 설명하기 위해서 N 개의 학습 데이터 중에서 임의의 입력력 데이터 (x_i, t_i) 를 고려하자. 여기서, 입력 데이터 $x_i = [x_{i1}, \dots, x_{in}]^T \in \mathbb{R}^n$ 이고, 목표 출력값 $t_i = [t_{i1}, \dots, t_{im}]^T \in \mathbb{R}^m$ 이다. 은닉층의 뉴런의 개수가 \tilde{N} 개이고 활성화 함수가 $g(x)$ 인 단일 은닉층을 갖는 신경회로망의 실제 출력값 o_j 는 다음과 같다.

$$\sum_{i=1}^{\tilde{N}} \beta_i \{ g(w_i \cdot x_j + b_i) \} = o_j, \quad j = 1, \dots, N \quad (1)$$

여기서, $w_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$ 은 입력 뉴런들과 i 번째 은닉 뉴런 사이에 연결된 가중치 벡터, $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$ 은 출력 뉴런들과 i 번째 은닉 뉴런 사이에 연결된 가중치 벡터 이이며, b_i 는 i 번째 은닉 뉴런의 바이어스 값을 나타낸다. 은닉 뉴런 \tilde{N} 을 가지는 단일 은닉층을 갖는 신경회로망은 학습을 통하여 N 개의 모든 학습 데이터에 대하여 오차가 제로인 다음 식 (2)를 만족할 경우 식 (3)를 만족하는 β_i, w_i, b_i 가 존재한다.

$$\sum_{j=1}^{\tilde{N}} \| o_j - t_j \| = 0 \quad (2)$$

$$\sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) = t_j \quad j = 1, \dots, N \quad (3)$$

N 개의 방정식으로 표현되는 식 (4)를 아래와 같이 간단히 표현하자.

$$H\beta = T \quad (4)$$

여기서,

$$H(w_1, \dots, w_{\tilde{N}}, b_1, \dots, b_{\tilde{N}}, x_1, \dots, x_{\tilde{N}}) = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_{\tilde{N}} \cdot x_1 + b_{\tilde{N}}) \\ \vdots & \dots & \vdots \\ g(w_1 \cdot x_N + b_1) & \dots & g(w_{\tilde{N}} \cdot x_N + b_{\tilde{N}}) \end{bmatrix} \quad (5)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \quad t = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m} \quad (6)$$

H 는 신경망의 은닉층의 출력 행렬값을 나타낸다. H 의 i 번째 열은 입력 x_1, x_2, \dots, x_N 의 각각에 대한 i 번째 은닉 뉴런의 출력 벡터이다. 여기서 β 의 극소노름최소자승해 $\hat{\beta}$ 는 아래와 같이 구해질 수 있다.

$$\hat{\beta} = H^+ T \quad (7)$$

3. 박테리아 생존 알고리즘 및 제안 방법

자연 선택은 동물들이 세대 번식을 하는데 있어서 성공적인 전략과 천적들의 공격으로부터 살아남는 것들만이 존재하도록 선택되어지고 유지되어진다. 이러한 여러 가지 동물들의 생존전략들은 기본적으로 먹이를 찾는 것으로부터 시작된다. 먹이 찾기는 동물들이 동물 자신들의 생존을 유지하는데 있어서 가장 기본적이고 근본적인 생존전략 중에 하나이다 [15]. 많은 세대가 지난 후에 다양한 생존전략들 중 우수한 전략을 가진 동물들만이 번식에 성공하고 진화하여 자신의 후손들을 유지하게 된다. 이러한 이론들은 공학에 있어 최적화 분야에 새로운 방법들을 제시하는데 바탕이 되어져 왔다 [14]. 박테리아 생존 알고리즘은 박테리아의 생존을 위한 행동들을 모방하여 최적화 문제에 적용한 것으로서 방향성을 가지는 탐색 전략과 이동 단위 거리의 조정으로 최적해를 탐색하는 특징을 가지고 있다.

3.1 박테리아의 생존을 위한 최적화

박테리아는 생존을 위하여 최적의 환경과 먹이를 찾아 항상 무작위로 이동하는 주화성 반응을 한다. 이동 후 그 위치가 생존에 유리한 환경인지 아닌지를 평가한 후 좋은 환경이면 그 곳에서 멈춰 생활을 하고 아니면 다시 임의의 방향으로 이동하게 된다. 좋은 환경에서의 박테리아들은 세포분열, 형질전환, 집합, 형질도입과 같은 재생산과 돌연변이를 하게 된다. 그러나 유리한 환경을 찾지 못한 박테리아들은 제거된다.

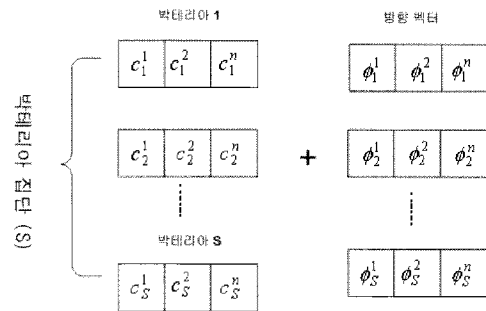


그림 2. 박테리아 집단의 초기 구조.

Fig. 2. Initial Structure of bacteria population.

이러한 박테리아의 생존을 위한 행동들은 기존의 다른 진화 알고리즘들이 최적화 문제에서 평가함수를 이용하여 어떤 임의의 해공간의 값들을 평가하는 과정들과 매우 유사하다. 만약 어떤 평가함수 $J(\theta)$ 를 최소화 하는 문제에 박테리아 생존 알고리즘들을 적용한다면 θ 는 박테리아의 위치(최적화 문제에서는 임의의 후보해)로 생각할 수 있고, $J(\theta)$ 가 작을수록 박테리아에 좋은 환경(최적화 문제에서는 최적해), $J(\theta)$ 가 커질수록 좋지 않은 환경으로 생각할 수 있다.

3.1 박테리아 생존 알고리즘의 최적화 문제 적용

박테리아 생존 알고리즘을 최적화 문제에 적용하기 위하여 아래와 같이 정의하기로 한다. j 는 주화성 단계를 표시하고 k 는 재생산 단계, 그리고 l 을 제거-분산 동작으로 정의하면 박테리아의 행동은 아래의 식(8)로 표현될 수 있다.

$$P(j, k, l) = \{ \theta^i(j, k, l) \mid i = 1, 2, \dots, S \} \quad (8)$$

식 (8)은 박테리아 집단 S 에서 l 번째 제거-분산 동작, k 번째 재생산, j 번째 주화성 단계를 가지는 i 번째 박테리아의 위치(θ)를 말한다. 여기서 우리는 i 번째 박테리아의 위치 $\theta^i(j, k, l)$ 에서의 평가된 값(예를 들어 환경의 최적성)을 $J(i, j, k, l)$ 으로 표현 할 수 있다. 최적화 문제에서는 비용함수로 생각할 수 있다. $C(i) > 0, i = 1, 2, \dots, S$ 는 주화성 단계에서 한 번 이동할 때의 이동 길이로 정의한다. $\phi(j)$ 는 주화성 단계 동안 박테리아가 방향 전환을 할 때 선택되어지는 무작위적인 방향을 나타낸다. 위의 기호들을 사용해서 아래에 식 (9)처럼 다음 단계의 박테리아의 행동을 표현한다. 식 (9)는 어떤 임의의 박테리아가 전 주화성 단계에서(j)에서 다음 주화성 단계($j+1$)에는 무작위 방향에서 $C(i)$ 만큼 이동 된다는 것을 의미한다.

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i)\phi(j) \quad (9)$$

만약 $\theta^i(j+1, k, l)$ 에서 $J(i, j+1, k, l)$ 가 $\theta^i(j, k, l)$ 에서 보다 더 우수한 값을 가지면 같은 방향에서 $C(i)$ 크기의 길이를 작게 하여 더 정밀한 탐색을 할 수 있고, 더 나쁜 값을 가진다면 $C(i)$ 의 길이를 크게 하여 좀 더 광범위한 탐색을 할 수 있도록 조절할 수 있다. 이러한 주화성 반응은 정해진 주화성 단계에 도달하거나 사용자가 원하는 목적을 달성하면 주화성 반응은 멈추게 된다. 박테리아 알고리즘의 수행 순서는 다음과 같다.

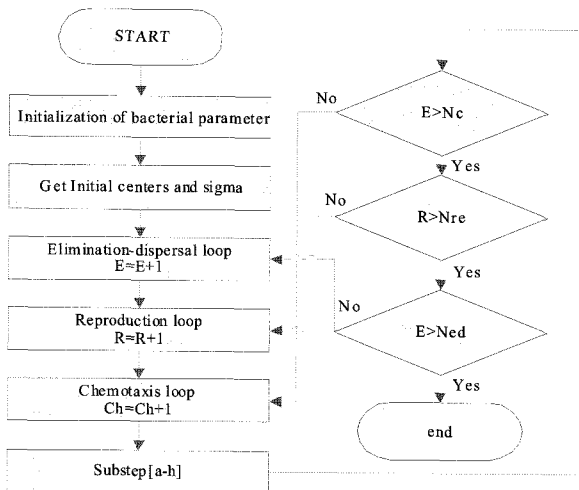


그림. 3 박테리아 먹이찾기 알고리즘의 순서도.
Fig. 3. Flowchart of bacterial foraging algorithm.

[step 1] BF 알고리즘의 초기의 파라미터들($n, S, N_c, N_{re}, N_{ed}, P_{ed}, C(i), \phi^i$)을 정의한다.

- 여기서, n : 문제 해공간의 차원. S : Bacteria의 집단 수.
- N_c : 주화성 단계의 반복 횟수.
- N_{re} : 재생산 단계의 반복 횟수.
- N_{ed} : 제거-분산 반응의 반복 횟수.

P_{ed} : 제거-분산의 비율, ϕ^i : 박테리아의 이동 방향.
 $C(i)$: 주화성 반응에서의 단위 이동 크기.

- [Step 2] 제거-분산 loop: $l = l + 1$
- [Step 3] 재생산 loop: $k = k + 1$
- [Step 4] 주화성 loop: $j = j + 1$
- [Step 4-1] 목적함수의 계산.
- [Step 4-2] 목적함수의 비교 및 무작위 방향 재설정. (전

단계의 목적함수와 비교하여 방향을 유지할 것인지 새로운 무작위적 방향을 선택할 것인지를 선택한다)

[Step 5] If $j < N_c$, go to step 4. (설계자에 의해 정해진 주화성 횟수나 목적함수 값을 만족하면 반복을 멈추고 그렇지 않으면 step 4로 가서 주화성 반응을 반복한다)

[Step 6] If $k < N_{re}$, go to step 3. (설계자에 의해 정해진 재생산 횟수만큼 반복)

[Step 7] If $l < N_{ed}$, then go to step 2 (설계자에 의해 정해진 제거-분산 횟수만큼 반복)

End

3.3 박테리아 알고리즘을 이용한 ELM 알고리즘의 초기 파라미터 최적화

ELM 알고리즘은 초기입력 가중치의 선택에 따라 그 성능이 큰 차이를 보이고, 가중치 행렬은 많은 입력과 은닉층 뉴런 수에 따라 차원이 커지기 때문에 일반적인 박테리아의 구조를 적용하기에는 문제점이 있다. 따라서 본 논문에서는 박테리아 알고리즘을 ELM 알고리즘에 적용하기 위하여 수정된 박테리아 구조를 제안하였다. ELM 알고리즘의 초기 입력 가중치는 입력의 수가 n 개이고 설계자에 의해 정의된 은닉 뉴런 수가 m 개 일 때 $m \times n$ 개의 초기 가중치 행렬을 가진다. 그러나 일반적인 박테리아 알고리즘에서의 각각의 박테리아 구조는 그림 2와 같이 $m \times 1$ 셀을 가지는 n 개의 박테리아로 구성되어 있기 때문에 방향성을 가지고 해공간을 탐색하는 박테리아 구조를 그대로 사용하기에는 문제점이 있다. 수정된 박테리아 구조에서는 그림 4와 같이 한 개의 박테리아가 $m \times n$ 개의 셀을 가지도록 구성하고 n 셀에 방향 벡터 $\phi(j)$ 를 더하여 주화성 반응 후의 박테리아의 위치를 생성하였다. 또한 $m \times n$ 차원을 가지는 하나의 박테리아만을 사용하였다.

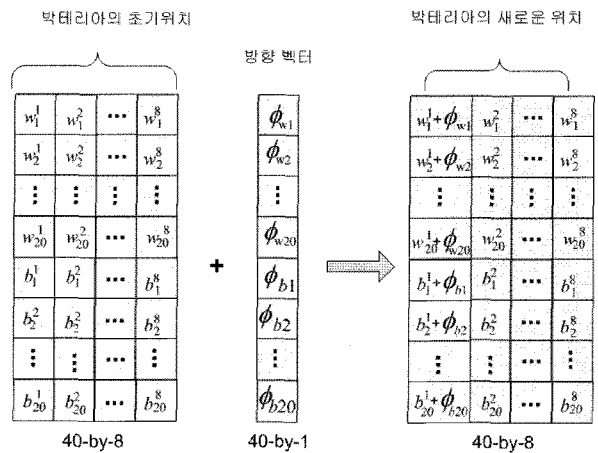


그림 4. 수정된 박테리아 집단의 초기 구조.
Fig. 4. Initial structure of modified bacteria population.

본 논문에서 박테리아의 차원 n 은 ELM의 초기가중치 행렬의 크기이고, 박테리아의 초기 위치 값은 ELM의 초기 가중치 행렬과 은닉층 바이어스 값, $C(i)$ 는 가중치들의 단위 변화량, ϕ^i 는 가중치 행렬의 방향벡터이다. 박테리아 알고리즘의 평가함수는 ELM 출력 오차의 RMS(Root mean-square)값을 이용하여 아래의 식 (10)과 같이 정의하였다.

$$\text{평가함수} = \frac{1}{1+S} \quad (10)$$

먼저 박테리아의 초기 위치(초기 가중치 행렬과 은닉층 바이어스)를 임의로 결정하고 주화성 반응 단계에서 박테리아의 평가함수를 평가한다. 그 후 방향 벡터를 이용하여 위치를 변경하면서 최적 값을 탐색하게 된다. (입력 가중치와 은닉층 바이어스들을 변경하면서 최적성능을 가지는 파라미터들을 찾는다). 또한, 일반적인 박테리아 알고리즘에서는 재생산과 제거-분산 단계가 있으나 본 논문에서는 변형된 박테리아 구조와 연산시간을 고려하여 적용하지 않았다.

4. 시뮬레이션 및 결과 고찰

4.1 함수 근사화 문제의 적용

함수 근사화 문제에 제안된 방법의 성능을 비교하기 위해 California Housing 데이터를 사용하였다 [8]. 사용된 데이터는 California에서 주택 가격을 예측하기 위한 20,640개의 관측 데이터들로 구성되어 있다. 주택 가격 변화들의 정보는 1990년 인구조사를 통하여 California에 모든 블록을 이용하여 수집되어졌다. 이 데이터에서 조사된 각각의 블록 그룹들은 평균적으로 복잡한 지역에 거주하는 1425.5 세대들이 포함되어져 있고, 각 블록 그룹들의 중심 사이의 거리는 위도와 경도를 이용하여 측정되어졌다.

최종적으로 California Housing Data는 9개의 변수를 가지는 20,640개의 데이터를 포함하고 있다. 연속적인 8개의 입력들은(중간 소득(median income), 주거 중위 연령(housing median age), 총 방수(total rooms), 총 침실 수(total bedrooms), 인구(population), 가구(households), 위도(latitude), 경도(longitude))이고 1개의 연속적인 출력은 중간 주택 가격(median house value)이다.

본 실험에서는 데이터의 순서대로 각각 8,000개의 학습데이터와 12,640개의 검증데이터로 구분하여 실험하였다. 제안된 알고리즘의 초기 파라미터들은 표 1과 같다.

표 1. 제안된 방법의 초기 파라미터.
Table 1. Initial parameters of proposed method.

Parameter	Value
S : Bacteria의 집단 수	1
N_c : 주화성 단계의 반복 횟수	8
N_{re} : 재생산 반응의 반복 횟수	1
N_{ed} : 제거-분산 반응의 반복 횟수	1
P_{ed} : 제거-분산의 비율,	0.5
C : 주화성 반응에서의 단위 이동 크기	0.01
\tilde{N} : hidden Neuron의 수	20
활성함수 : $F(x)=1/(1+\exp(-x))$	

모든 데이터들은 [0,1] 사이로 일반화된 값을 사용하였다. 표 2는 California housing 데이터에 대한 각각의 알고리즘들의 성능을 비교한 것이다. 표 2에서 알 수 있듯이 학습속도에서는 SVM과 비슷한 반면 ELM과 BP 방법보다 느리고, 학습 RMS는 SVM의 학습오차보다 0.11 더 우수하였으며, 검증 RMS에서는 평균적으로 0.1 정도 향상된 성능을 보였다.

표 2. California housing 데이터를 이용한 성능비교.
Table 2. Performance comparison using California housing data.

Method	Time		Training RMS	Testing RMS	No of SVs/Neurons
	Training	Testing			
ELM[8]	0.27	0.14	0.1358	0.1365	20
BP[8]	295.23	0.28	0.1369	0.1426	20
SVM[8]	558.41	20.97	0.1267	0.1275	2534
제안된 방법	30.85		0.0173	0.0267	20

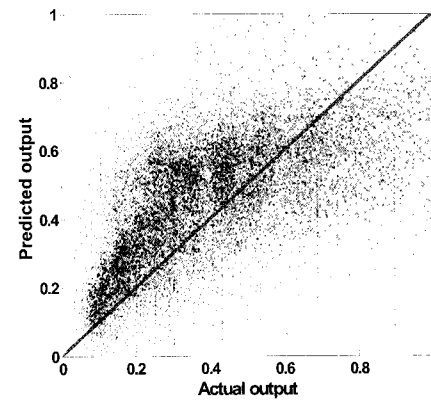
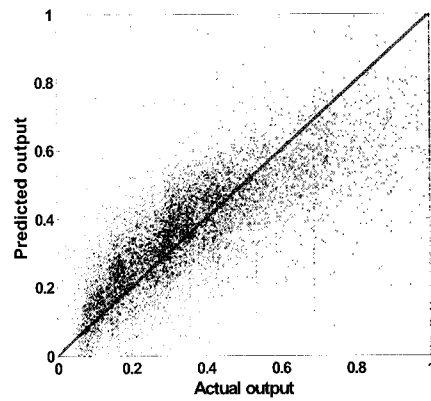


그림 5. 제안된 방법의 결과.
Fig. 5. Result of proposed method.

4.2 패턴 분류 문제의 적용

제안된 알고리즘의 성능을 평가하기 위하여 1988년 존 홉킨스 대학의 피마족 인디언의 당뇨병 데이터(PIDD:Pima Indians Diabetes Data)를 이용하여 기존 연구방법과 비교하였다. 당뇨병 데이터는 768개의 데이터로 구성되어 있으며, 비교를 위해 기존 방법과 마찬가지로 입력데이터를 [0, 1] 사이로 정규화하여 사용하였다. 총 데이터 768개 중 75%인 577개를 학습데이터로 사용하였고, 나머지 25%인 191개의 검증데이터를 이용하여 제안방법의 성능을 분석하였다 [6]. 표 3에서는 PIDD 데이터에 대하여 제안된 방법과 기존 알고리즘들에 대한 성능비교를 나타내었다. 표 3에서 보이는 바와 같이 학습데이터에서는 BP 방법보다 성능이 우수하지 않

지만 다른 방법보다 우수한 성능을 보이고 검증데이터에서는 다른 방법들보다 우수한 성능을 보이고 있다.

표 3. 피마족 인디언의 당뇨병 데이터를 이용한 성능비교
Table 3. Performance comparison using Pima indians diabetes data

Method	Training Time (Seconds)	Success Rate		No of SVs/Neurons
		Training	Testing	
ELM[8]	0.015	78.71	76.54	20
BP[8]	16.196	92.86	63.45	20
SVM[8]	0.1860	78.76	77.31	317.16
제안된 방법	2.57	79.69	79.17	20

5. 결 론

본 논문에서는 최적화 알고리즘 중 수정된 BF알고리즘을 이용하여 ELM의 초기 입력 가중치들과 은닉층 바이어스들을 동조하는 방법을 제안하였다. 제안된 방법은 학습 속도가 빠른 ELM의 초기 파라미터들을 동조하기 위하여 입력 가중치들과 은닉층 바이어스들을 초기 박테리아로 선택한 후 BF 알고리즘의 주화성 반응, 재생산 과정, 제거-분산 과정을 거쳐 최적에 가깝도록 동조하였다. 제안된 알고리즘의 성능을 평가하기 위하여 California Housing data를 이용한 함수 근사화 문제와 실제의 내과 진단 데이터의 분류 문제에 적용하였다. 함수 근사화 문제에서 기존의 방법들 보다 RMS 수치가 평균 0.1 향상된 결과를 보였으며, 분류 문제에서는 기존의 ELM 방법에 비해 학습데이터 0.2% 와 검증데이터 3.5%의 더 우수한 성능을 보였다.

참 고 문 헌

[1] Richard O. Duda, Peter E. Hart, David G. Stork, Pattern Classification, John Wiley & Sons, 2001.
 [2] J-S. R. Jang, C. T. Sun, and E. Mizutani, Neuro-Fuzzy and Soft Computing : A Computational Approach to Learning and Machine Intelligence, Prentice Hall, 1997.
 [3] J. S. R. Jang, "ANFIS : Adaptive Network-based Fuzzy Inference System," *IEEE Trans on System, Man, and Cybernetics*, Vol. 23, No. 3, pp. 665- 685, 1993.
 [4] Mohammad Fazle Azeem, Madasu Hanmandlu, and Nesar Ahmad, "Structure Identification of Generalized Adaptive Neuro-Fuzzy Inference Systems," *IEEE Trans on Fuzzy Systems*, Vol. 11, No. 5, pp. 666-681. 2003.
 [5] D. Rumelhart, and J. McClelland, "Parallel distributed processing", Vol. 1, MIT Press, 1986.
 [6] S. Suresh, S.N. Omkar, and V. Mani, "Parallel implementation of back-propagation algorithm in

networks of workstations", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 16, pp. 24-34, 2005.
 [7] Hsu, C.-T. Kang, M.-S. and Chen, C.-S, "Design of adaptive load shedding by artificial neural networks", *IEE Proceed., Generation, Transmission and Distribution*, Vol. 152, pp. 415-421, 2005.
 [8] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks", *International Joint Conference on Neural Networks (IJCNN2004)*, pp. 25-29, 2004.
 [9] M.-B. Li, G.-B. Huang, P. Saratchandran, N. Sundararajan, "Fully complex extreme learning machine", *Neurocomputing*, vol. 68, pp. 306-314, 2005.
 [10] Qin-Yu Zhu, A.K. Qin, P.N. Suganthan, Guang-Bin Huang, "Evolutionary extreme learning machine", *Pattern Recognition*, Vol. 38, pp. 1759 - 1763, 2005.
 [11] 한승수, "유전알고리즘을 이용한 신경망 구조 및 파라미터 최적화", *퍼지 및 지능시스템학회 논문집*, Vol. 11, No. 3, pp. 215-222, 2001.
 [12] G.W. Game and C. D. James, "The application of genetic algorithms to the optimal selection of parameter values in neural networks for attitude control system", *IEEE Colloquium High Accuracy Platform Control Space*, 1993.
 [13] H. J. Bremermann, "Chemotaxis and optimization," *J. Franklin Inst.*, Vol. 297, pp. 397-404, 1974.
 [14] PASSINO, K. M., "Biomimicry of Bacterial foraging for Distributed Optimization", University Press, Princeton, New Jersey, 2001.
 [15] W. John O'brien, Howard I. Browman, and Barbara I. Evans, "Search Strategies of Foraging Animals," *American Scientist*, Vol. 78 pp. 152-160, 1990. R. M. Tong, "The Evaluation of Fuzzy Models Derived from Experimental Data," *Fuzzy Sets and Systems*, Vol. 4, pp. 1-12, 1980.

저 자 소 개

조재훈(Jae Hoon Cho)

2006년 제16권 3호 참조

이대종(Dae Jong Lee)

2007년 제17권 5호 참조

전명근(Myung Geun Chun)

2007년 제17권 5호 참조