

# 퍼지 논리 시스템을 이용한 학습률 자동 조정 방법에 관한 연구

주영호\* · 김태경\*\* · 김광백\*\*

## A Study on Auto-Tuning Method of Learning Rate by Using Fuzzy Logic System

Young-Ho Ju\* · Tae-Kyung Kim\*\* · Kwang-Baek Kim\*\*

### 요 약

본 논문에서는 역전파 알고리즘의 성능 개선을 위해 퍼지 논리 시스템을 이용한 학습률 자동 조정 방법을 제안한다. 제안된 방법은 목표값과 출력값의 차이에 대한 절대값이  $\epsilon$  보다 적거나 같으면 정확성으로 분류하고 크면 부정확성으로 분류한다. 정확성의 총 개수를 퍼지 논리 시스템에 적용하여 학습률과 모멘텀을 동적으로 조정한다.

제안된 방법을 XOR 문제와 숫자패턴 문제에 적용하여 실험한 결과, 기존의 역전파 알고리즘, 모멘텀 방식, Jacob의 delta-bar-delta 방식보다 성능이 개선됨을 확인하였다.

**Key words :** 역전파 알고리즘, 퍼지 논리 시스템, 학습률, 모멘텀, delta-bar-delta 방식

### 1. 서론

하나 이상의 은닉층을 가진 다층 신경망은 단층 퍼셉트론을 확장한 형태로서 공학이나 과학 등 사회 전 분야에 걸쳐 응용되고 있다. 다층 신경회로망에서 일반적으로 사용하는 학습 알고리즘은 역전파 알고리즘(BP: backpropagation algorithm)으로 이는 목표 출력과 실제 출력의 차의 자승 합을 목적함수로 하여 그 목적함수의 최소치를 스텝 폭 고정 및 최급강하법(steepest descent method)으로 구하는 방식이다. 역전파 알고리즘은 훈련 시간이 많이 소요되고, 초기 연결 강도에 따른 모델의 불안정성(instability), 그리고 지역 최소화(local minimum)에 빠지는 단점이 있다[1]. 이러한 문제점을 개선한 연구 방법으로는 크게 경험적 접근 방법과 수치 최적화 방법으로 분류된다[2].

경험적 접근 방법으로는 학습률을 가변화하는 방법, 모멘텀법, 변수들의 재 스케일링, 다층 신경회로망의 은닉층 수 및 각 층의 뉴런 수를 조절하는 방법 등이 있다. 학습률을 가변화하는 방법에는 학습률을 선형적으로 증가시키다가 지수적으로 감소시키는 Jacobs등에 의한 delta-bar-delta법칙 그리고 학습 반복수와 방향에 의해 학습률을 계산하는 방

법 등이 있다[3,4]. 모멘텀법은 이전의 연결강도 변화량을 고려하여 최종 연결강도 변화량을 결정하는 방법으로 알고리즘이 간단하고 학습이 고속으로 이루어지는 장점이 있지만 적용하는 문제에 따라 적절한 파라미터가 서로 다르므로 시행착오적인 파라미터의 설정이 필요하다[5]. 수치 최적화 기술로는 conjugate gradient algorithm, inverse gradient method, 변형된 오차함수를 이용하는 방법 및 확률적 방식을 이용하는 방법 등이 있다[6,7].

본 논문에서는 역전파 알고리즘의 성능 개선을 위해서 퍼지 논리 시스템을 이용한 학습률 자동 조정 방법을 제안한다. 제안한 방식은 목표값과 출력값의 차이의 절대값이  $\epsilon$  보다 적거나 같으면 정확성으로 분류하고 크면 부정확성으로 분류한다. 정확성의 총 개수를 퍼지 논리 시스템에 적용하여 학습률과 모멘텀을 동적으로 조정한다. 제안한 방식의 유용성을 확인하기 위하여 XOR 문제, 숫자 패턴 분류를 대상으로 기존의 역전파 알고리즘, 모멘텀 방식, delta-bar-delta 방식과 시뮬레이션을 통해서 성능을 비교 분석한다.

### 2. 관련 연구

#### 2.1 오류 역전파 알고리즘

\* 신라대학교 정보통신공학과

\*\*신라대학교 컴퓨터공학과

Rumelhart에 의하여 제안된 오류 역전과 학습 알고리즘은 입력 벡터에 대한 출력층의 출력값을 원하는 출력값(교사신호)과 비교하고 그 오차를 각 층의 가중치로 전파하는 방식이다. 이러한 일련의 과정들은 순방향 과정과 역방향 과정으로 구분된다. 순방향 과정에서 각층의 뉴런에 대한 출력값은 동일한 방식으로 계산한다. 그리고 오차 급강하법은 출력층의 오차에 대한 각층의 가중치가 chain rule에 의하여 계산된다. 오류 역전과 알고리즘은 학습 속도가 느리고 최적의 은닉층 뉴런 수를 결정할 수 없는 문제점이 있다[8]. 학습 속도의 느림은 역전과 학습 방식이 기울기 감소 방법을 사용하기 때문이며, 이는 기울기 감소 방법을 사용하는 알고리즘의 공통된 문제이다. 그리고 은닉층의 뉴런 수가 적으면 학습이 정체되거나 국부극소점에 빠져 학습이 잘 이루어 질 수 없다. 따라서 오류 역전과 알고리즘은 지역 최소화에 빠질 수 있으며, 학습 시간이 많이 소요되는 단점이 있다. 이러한 문제점을 개선하기 위한 방법으로는 연결 가중치의 수정 식에 오차의 2차항을 사용하는 적응최소 2차방법(adaptive least square method)과 학습률(learning rate)을 조정하는 방법, 은닉층의 뉴런 수를 동적으로 조정하는 방법 등이 있다[6,7,9].

## 2.2 Delta-bar-delta 알고리즘

오류 역전과 알고리즘의 성능 개선을 위한 방법 중에는 Jacobs 등에 의해 제안된 delta-bar-delta 알고리즘이 있다[3].

각층 신경회로망이 입력층  $i$ , 은닉층  $j$ , 출력층  $k$ 로 구성될 때, delta-bar-delta 알고리즘은 식(1)과 같은 연결강도(weight)의 변화를 갖는다.

$$\begin{aligned} w_{ki}(t+1) &= w_{ki}(t) - a_{kj}(t+1) \frac{\partial E}{\partial w_{ki}} \\ &= w_{ki}(t) + a_{kj}(t+1) \delta_k z_j \end{aligned} \quad (1)$$

식(1)에서  $a_{kj}$ 는 학습률,  $\delta_k$ 는 출력층 오차이며,  $z_j$ 는 은닉층 뉴런의 출력이다. 각 출력층 뉴런에서  $\delta_k$ 는 식(2)와 같으며, 각 은닉층 뉴런에서  $\delta_k$ 는 식(3)과 같다.

$$\delta_k = \frac{\partial E}{\partial w_{ki}} = -\delta_k z_j \quad (2)$$

$$\delta_j = \frac{\partial E}{\partial w_{ji}} = -\delta_j x_i \quad (3)$$

delta-bar-delta 규칙은 각 출력 뉴런에 대하여 현재의 정보를 식(4)와 같이 delta-bar를 형성하기 위하여 파생된 과거의 정보를 조합하여 이용한다.

$$\bar{\Delta}_{kj}(t) = (1-\beta)\Delta_{kj}(t) + \beta\bar{\Delta}_{kj}(t-1) \quad (4)$$

식(4)에서 파라미터  $\beta$ 의 값은  $0 < \beta < 1$  사이의 고정된 상수이다. delta와 delta-bar 변화에 대한 학습률의 변화는 다음과 같다. 만약 연결강도의 변화

가 연속된 학습과정에서 같은 방향으로 이루어진다면 학습률은 증가한다. 이것은  $\bar{\Delta}_{kj}(t-1)$ 과  $\Delta_{kj}(t)$ 가 같은 부호에서 발생하며, 이 때 학습률은 증가해야 한다. 또한 만약  $\bar{\Delta}_{kj}(t-1)$ 과  $\Delta_{kj}(t)$ 의 부호가 반대 부호를 가지면 학습률은 현재값의  $(1-\gamma)$ 의 비율로 감소한다. 이러한 가변 학습률은 각 층에 대하여 다음과 같은 식으로 표현할 수 있다[3].

$$\begin{aligned} \alpha_{kj}(t+1) &= \alpha_{kj}(t) + k && \text{if } \bar{\Delta}_{kj}(t-1) \cdot \Delta_{kj}(t) > 0 \\ &= (1-\gamma)\alpha_{kj}(t) && \text{if } \bar{\Delta}_{kj}(t-1) \cdot \Delta_{kj}(t) < 0 \\ &= \alpha_{kj}(t) && \text{if } \bar{\Delta}_{kj}(t-1) \cdot \Delta_{kj}(t) = 0 \end{aligned} \quad (5)$$

$$\begin{aligned} \alpha_{ji}(t+1) &= \alpha_{ji}(t) + k && \text{if } \bar{\Delta}_{ji}(t-1) \cdot \Delta_{ji}(t) > 0 \\ &= (1-\gamma)\alpha_{ji}(t) && \text{if } \bar{\Delta}_{ji}(t-1) \cdot \Delta_{ji}(t) < 0 \\ &= \alpha_{ji}(t) && \text{if } \bar{\Delta}_{ji}(t-1) \cdot \Delta_{ji}(t) = 0 \end{aligned} \quad (6)$$

## 3. 제안된 학습률 조정 방법

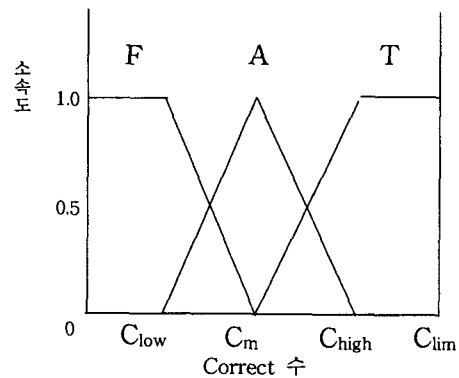
본 논문에서는 역전과 알고리즘의 성능을 개선하기 위해서 퍼지 논리 시스템을 이용해 학습률과 모멘텀을 자동으로 조절하는 방식을 제안한다. 각 패턴에 대한 출력층의 실제 출력값과 목표값의 차이의 절대값이  $\epsilon$  보다 적거나 같은 경우에는 정확성으로 분류하고 큰 경우에는 부정확성으로 분류하여 퍼지 논리 시스템의 입력으로 사용한다. 퍼지 논리 시스템을 이용한 학습률의 조정은 식(7)과 같다.

$$\alpha = \text{Fuzzy}(\text{Correct}, \text{Incorrect}) \quad (7)$$

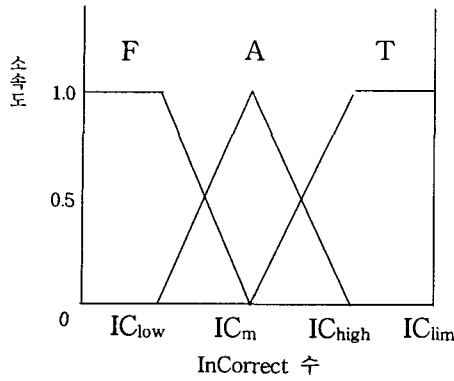
모든 패턴의 출력층 출력값이 정확성으로 분류되면 학습을 종료한다. 모멘텀은 퍼지 논리 시스템에서 출력된 학습률을  $\zeta$ 에 감하여 구한다. 식(8)은 모멘텀을 동적으로 조정하는 식이다.

$$\mu = \zeta - \alpha \quad (8)$$

제안된 방법에서 정확성의 소속 함수는 그림 1과 같고 부정확성의 소속 함수는 그림 2와 같다.



[그림 1] 정확성의 소속함수



[그림 2] 부정확성의 소속함수

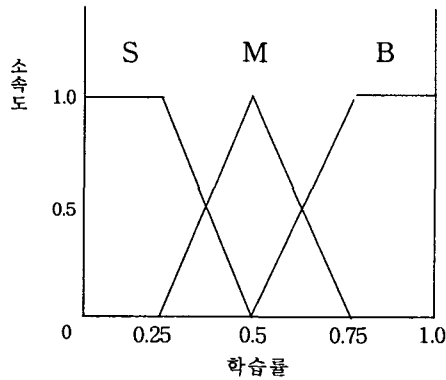
그림 1에서 정확성의 소속 함수 구간  $[C_{low}, C_{high}]$  는 다음과 같이 계산한다.

$$C_{low} = \log_2(\text{입력노드수} + \text{패턴수}) \quad (9)$$

$$C_{high} = C_{lim} - C_{low} \quad (10)$$

여기서  $C_{lim}$ 는 정확성의 한계치를 의미한다.

학습에 적용될 학습률을 계산하는 출력 소속함수는 그림 3과 같다.



[그림 3] 학습률 출력 소속함수

제안된 방법에서 퍼지 제어 규칙을 if ~ then 형식의 퍼지 관계로 표현하면 다음과 같다.

- $R_1 : \text{If correct is } F, \text{ incorrect } F \text{ Then } \alpha \text{ is } B$
- $R_2 : \text{If correct is } F, \text{ incorrect } A \text{ Then } \alpha \text{ is } B$
- $R_3 : \text{If correct is } F, \text{ incorrect } T \text{ Then } \alpha \text{ is } B$
- $R_4 : \text{If correct is } A, \text{ incorrect } F \text{ Then } \alpha \text{ is } M$
- $R_5 : \text{If correct is } A, \text{ incorrect } A \text{ Then } \alpha \text{ is } M$
- $R_6 : \text{If correct is } A, \text{ incorrect } T \text{ Then } \alpha \text{ is } M$
- $R_7 : \text{If correct is } T, \text{ incorrect } F \text{ Then } \alpha \text{ is } S$

$R_8 : \text{If correct is } T, \text{ incorrect } A \text{ Then } \alpha \text{ is } S$

$R_9 : \text{If correct is } T, \text{ incorrect } T \text{ Then } \alpha \text{ is } S$

정확성과 부정확성이 퍼지 논리 시스템의 입력값으로 정해지면 각 소속 함수에 대한 정확성과 부정확성의 소속도를 구한다. 각 소속 함수에 대하여 소속도가 구해지면 퍼지 제어 규칙을 적용하고 Max\_Min 방법으로 추론한다. 퍼지 추론 후에는 비퍼지화 하여 학습에 사용될 학습률을 구한다. 비퍼지화는 무게 중심법을 사용하는데, 식(11)과 같다.

$$\alpha = \frac{\sum \mu(y)y}{\sum y} \quad (11)$$

## 4. 실험 및 결과 분석

제안된 방법의 성능을 분석하기 위하여 Intel PentiumIII-866MHz CPU 와 128MB RAM이 장착된 IBM 호환 PC상에서 VC++로 실험하였다. 제안된 방법의 성능을 평가하기 위하여 XOR 문제와 숫자 패턴 분류 문제에 적용하여 학습 반복 횟수와 수렴성을 기존의 역전파 알고리즘, 모멘텀 방식, delta-bar-delta 방식과 비교 분석하였다.

### 4.1 XOR 문제

신경 회로망에서 XOR 문제가 벤치마크로 사용되는 이유는 벤치마크의 문제가 은닉층 노드들을 요구하는 고전적인 문제이고 다른 많은 어려운 문제들이 부분 문제로 XOR문제를 이용하기 때문이다. XOR 문제에서는 1 개의 은닉층에 2개의 은닉층 뉴런과 2개의 입력 뉴런, 1개의 출력 뉴런으로 신경회로망을 구성하였다. 학습 판정은 3000번의 Epoch의 수행에서 모든 패턴들의 실제 출력값과 목표값 간의 차이의 절대값이  $\epsilon (\epsilon \leq 0.1)$ 보다 적거나 같으면 정확성으로 분류되는 것으로 하여 10회의 실험을 수행하였다. XOR 문제에서 입력값과 목표값을 표1로 나타내었다.

<표 1> XOR 문제의 입력값과 목표값

입력값	0,0	0,1	1,0	1,1
목표값	0	1	1	0

XOR 실험에서 사용된 각 학습방법의 파라미터는 표 2와 같고, 실험 결과는 표 3과 같다. 여기서  $\epsilon$ 는 0.1 로 설정하였다.

<표 2> XOR 문제에 사용된 파라미터

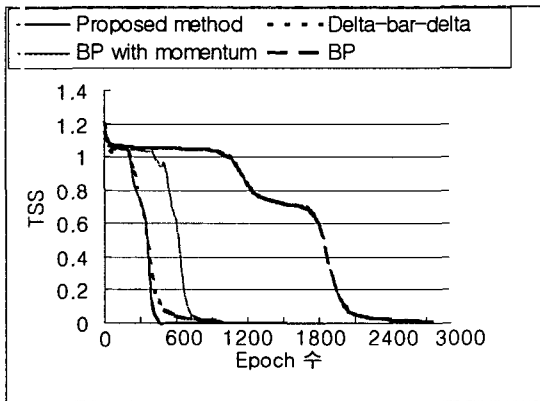
학습 방법 \ 파라미터			$k$			
BP	0.7					
BP with momentum	0.5	0.7				
Delta-bar-Delta	0.8		0.33	0.03	0.7	
Proposed method						1.5

여기서  $\alpha$ 는 학습율,  $\mu$ 는 모멘텀 계수,  $k, \gamma, \beta$ 는 delta-bar-delta 상수,  $\epsilon$ 는 본 논문에서 제안한 정확성을 분류하는 기준 파라미터이고,  $\zeta$ 는 모멘텀을 구하는 파라미터이다.

<표 3> XOR 문제에서 각 학습 방법간의 학습 결과

학습 방식	실험 횟수	학습 성공	평균 Epoch 수
BP	10	8	2411
EP with momentum	10	9	982
Delta-bar-Delta	10	10	632
Proposed method	10	10	432

XOR 문제에서 제안된 방법이 기존의 방법들 보다 수렴성과 학습 속도가 개선된 것을 표 3에서 확인할 수 있다. 제안된 방법과 기존 방법들 간의 오차 제곱 합 곡선을 그림 4로 나타내었다. 그림 4에서와 같이 제안된 방법이 기존 방법들 보다 초기의 수렴 속도가 빠르고 오차 제곱 합이 적은 것을 알 수 있다.



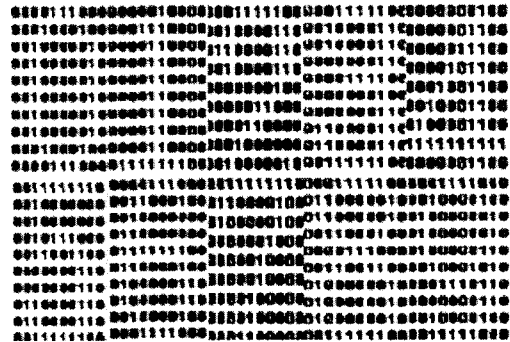
[그림 4] 오차 제곱 합 곡선

#### 4.2 숫자 패턴 분류

숫자 패턴 분류는  $10 \times 10$  크기의 아라비아 숫자 0에서 9까지를 입력값으로 사용하였고, 0과 1로 구성된  $100 \times 1$ 의 열 벡터로 변환하여 학습에 적용하였다. 실험에 적용된 아라비아 숫자는 그림 5와

같다.

숫자 패턴 분류에서는 하나의 은닉층에 10개의 은닉층 뉴런 그리고 100개의 입력 뉴런, 4개의 출력 뉴런으로 신경회로망을 구성하였다. 학습 판정은 3000번의 Epoch의 수행에서 모든 패턴들의 실제 출력값과 목표값의 차이의 절대값이  $\epsilon$  ( $\epsilon \leq 0.1$ ) 보다 적거나 같으면 정확성으로 분류되는 것으로 하여 10회의 실험을 수행하였다. 숫자 패턴 분류에서 목표값은 표 4와 같이 설정하였다.



[그림 5] 아라비아 숫자 패턴(10 × 10)

<표 4> 숫자 패턴 분류에서의 목표값

숫자	0	1	2	3	4	5	6	7	8	9
목표값	0	0	0	0	0	0	0	0	1	1
	0	0	0	0	1	1	1	1	0	0
	0	0	1	1	0	0	1	1	0	0
	0	1	0	1	0	1	0	1	0	1

숫자 패턴 분류 실험에서 사용된 각 학습방법의 파라미터는 표 5와 같고, 학습 결과는 표 6과 같다. 여기서  $\epsilon$ 는 0.1로 설정하였다.

<표 5> 숫자 패턴 분류에서의 각 파라미터

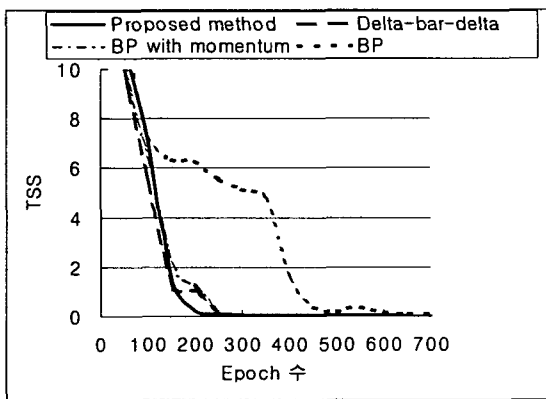
학습방식 \ 파라미터			$k$			
BP	0.5					
BP with momentum	0.5	0.7				
Delta-bar-Delta	0.8		0.15	0.3	0.7	
Proposed method						1.0

표 6과 같이 제안된 방법이 기존의 방법들보다 학습 시간이 적게 소요되고 학습 성능도 개선되었다.

<표 6> 숫자 패턴 분류에서의 학습 결과

학습 방식	실험 횟수	학습 성공	평균 Epoch 수
BP	10	10	812
BP with momentum	10	10	473
Delta-bar-Delta	10	10	325
Proposed method	10	10	232

그림 6의 오차 제곱 합 곡선에서도 제안된 방법이 기존의 방법들 보다 초기의 수렴 속도가 빠르고 오차 제곱 합도 적은 것을 확인할 수 있다.



[그림 6] 오차 제곱 합 곡선

오류 역전파 알고리즘은 은닉층의 불충분한 노드 수로 인한 지역 최소화에 빠질 가능성이 있고 은닉층의 노드 수가 필요 이상으로 많으면 판별 성능에 영향을 거의 미치지 않는 부가적인 노드가 존재하여 학습 시간이 많이 소요된다. 따라서 은닉층의 노드 수에 따른 수렴성을 확인하기 위해 은닉층 노드 수를 3개부터 5개까지 설정하여 실험한 결과는 표 7과 같고 실험에 설정된 파라미터는 표 8과 같다.

<표 7> 은닉층 노드 수에 따른 실험 결과

학습 방식	은닉층 노드수	실험 횟수	학습 성공	평균 Epoch 수
BP	3	10	6	2362
	4	10	8	1932
	5	10	10	1331
BP with momentum	3	10	3	1762
	4	10	9	1021
	5	10	9	562
delta-bar-delta	3	10	9	1932
	4	10	10	639
	5	10	10	442
proposed method	3	10	9	994
	4	10	10	519
	5	10	10	420

표 7과 같이 BP 와 BP with momentum은 은닉층 노드 수가 3개 또는 4개일 경우에는 지역 최소화 빠지는 경우가 발생하였으나 제안된 방법의 경우에는 기존의 방법보다 초기 은닉층의 노드 수에 민감하게 반응하지 않았으며 지역 최소화에 빠진 횟수도 적었다. 그러므로 제안된 방법이 기존의 방법들 보다 학습의 수렴성이 개선되었고 학습 시간도 단축된 것을 확인하였다.

<표 8> 학습에 사용된 파라미터

학습 방법 \ 파라미터	$\eta$	$\alpha$	$k$	$\epsilon$	$\delta$
BP	0.7				
BP with momentum	0.5	0.5			
Delta-bar-Delta	0.8		0.15	0.3	0.7
Proposed method					1.0

XOR 문제와 숫자 패턴 분류의 학습 성능 평가에서 다른 방법들에 비해 제안한 방법이 수렴성과 학습 시간이 개선된 것을 확인하였다.

## 5. 결론

본 논문에서는 역전파 알고리즘의 성능을 개선시키기 위해서 퍼지 논리 시스템을 이용하여 학습률을 동적으로 조정하는 방법을 제안하였다. 출력값과 목표값의 차이의 절대값이  $\epsilon$  보다 적거나 같으면 정확하게 분류하였고, 그렇지 않은 경우에는 부정확성으로 분류하였다. 정확성과 부정확성의 수를 퍼지 논리 시스템에 적용하여 학습률을 동적으로 조정하였고 동적으로 조정된 학습률을 이용하여 모멘텀을 조정하여 고속으로 수렴할 수 있도록 하였다. 제안된 방법의 학습 성능을 평가하기 위하여 XOR 문제와 숫자 패턴 분류 문제에 적용한 결과, 기존의 방법들 보다 제안한 방법이 학습 횟수가 현저히 감소하였다. 또한 은닉층의 노드 수에 따른 실험에서도 제안된 방법이 기존의 방법들에 비해 지역 최소화 빠질 가능성도 적었고 오차 제곱 합도 다른 방식에 비해 현저히 감소하여 학습 속도와 수렴성이 개선된 것을 확인하였다.

## 참고문헌

- [1] R. Hecht-Nielsen, "Theory of Backpropagation Neural Networks," Proceedings of IJCNN, Vol.1, pp.593-605, 1989.
- [2] Peiman G. Maghami and Dean W. Sparks, "Design of Neural Networks for Fast Convergence and Accuracy: Dynamics and Control", IEEE Transactions on Neural Networks, vol. 11, No. 11, pp. 113~123, 2000.
- [3] R. A. Jacobs, "Increased rates of convergence

- through learning rate adaptation," *Neural Networks*, Vol. 1, No. 4, pp. 295~308, 1988.
- [4] A.K.Rigler, J.M. Irvine and T.P. Vogl, "Rescaling of Neural Networks, Vol. 3, No. 5, pp. 61~573, 1990.
  - [5] Cheung, et al, " Relative Effectiveness of Training Set Patterns for Backpropagation," *IJCNN*, Vol.1, pp.673-678, 1990.
  - [6] M. T. Hagan and M. Menhaj, "Training Feedforward Networks with the Marquardt algorithm," *IEEE Tran. on Neural Networks*, Vol. 5, No. 6, 1994.
  - [7] C.Charalambous, "Conjugate gradient algorithm for efficient training of artificial neural networks," *IEEE Proceedings*, Vol. 139, No. 3, pp. 301~310, 1992.
  - [8] Y. Hirose, K. Yamashita, S. Hijiya," Back-Propagation Algorithm Which Varies the Number of Hidden Units", *Neural Networks*, Vol.4, pp.61-66, 1991.
  - [9] K. B. Kim, M. H. Kang and E. Y. Cha, "Fuzzy Competitive Backpropagation using Nervous System," *Proceedings of WCSS*, pp.188-193, 1997.