

## 퍼지를 이용한 신경망에서의 파라미터의 수정

\*이광원, \*고재호, \*\*배영철, \*임화영

\* 광운대학교 제어계측공학과, \*\* 산업 기술 정보원

### Parameter Adaptation in Neural Network Using Fuzzy

\*Lee Kwang-Won, \*Ko Jae-Ho, \*\*Bae Young-Chul, \*Yim Wha-Young

\* Dept. of Control and Instrumentation Eng, Kwang Woon Univ., \*\* KINITI

#### Abstract

Back-propagation is one of the efficient algorithms used to nonlinear optimizations or controls. In spite of its structural simplicity and learning ability, learning time is very long or bad case converge local minimum on complicate input patterns. In order to improve these matters varying learning rate and momentums were proposed. In this paper, to improve its performance fuzzy is adjusted in parameters, learning rate and momentums. Parameters are adjusted by errors and change of errors adaptively. In order to evaluate proposed method simulated with MATLAB on inverted pendulum.

#### 1. 서 론

신경망은 비선형 최적화 알고리즘의 하나로 복잡한 수식 계산 과정이 필요없이 단순한 반복적 연산에 의해 수행되므로 비선형 문제에 쉽게 적용될 수 있다. 여러 가지 신경망 가운데 오차 역전파 알고리즘은 (back-propagation, 이후 BP로 표기) 제어나 최적화를 목적으로 많이 사용되고 있다. 학습된 신경망은 입력 패턴에 의한 기대되는 출력 이외에도 학습되지 않은 입력에 대해서도 적절한 출력을 낼 수 있는 학습과 연상이 가능한 장점이 있으나 복잡한 입력 패턴의 경우에는 수렴속도가 느리거나 더 나쁜 경우에는 지역 최소점으로 수렴하여 학습이 진전되지 않는 경우가 발생할 수 있다. 따라서 원하는 오차까지의 수렴성과 수렴 속도가 가장 큰 문제로 대두된다. 이에 대한 대안으로 학습률을 적응적으로 변화시키는 방법과 모멘텀을 적용하여 과거의 요소를 이용하는 방법이 제시되었다.[4] 또한 BP 구조가 다층의 이용하는 구조이므로 각 층의 노드 수를 능동적으로 가감하여 최적의 학습 구조를 찾아 내는 방법도 제시 되었다.[1] 이러한 결과로 BP 알고리즘의 성능은 알고리즘에 의해 출력과

목표 출력과의 오차 제곱의 합이 일정한 범위 내에 이를때까지 반복하는 학습의 반복 횟수로 평가될 수 있다. 본 논문에서는 신경망의 구조를 변화시키지 않고 신경망 내부의 파라미터 즉 학습률과 모멘텀에 퍼지를 적용하고 오차의 경사값과 그 변화율에 따라 적응적으로 파라미터를 변경함으로써 성능을 향상시켰다. 또한 그 결과를 도립진자에 대해 MATLAB을 이용하여 시뮬레이션하고 그 결과를 평가하였다.

#### 2. BP 알고리즘의 학습 방법과 파라미터 갱신

BP 알고리즘이란 일반화  $\delta$  rule이라고 불리우는 supervised 학습 방식의 알고리즘으로 다층 신경망을 통해서 얻고자하는 목표 출력과 실제 출력과의 오차를 이용해서 이 오차를 줄이는 방향으로 각 층의 가중치를 갱신하는 방법을 사용한다.  $\delta$ 라는 오차를 포함하는 함수를 정의하고 최상층부터  $\delta$ 를 계산해서 아래층으로 이전층의  $\delta$ 를 포함하는 새로운  $\delta$ 를 정의 함으로써 오차를 아래 층까지 역 전파시켜 network를 학습 시키게 된다. 오차를 감소시키기 위해 많은 알고리즘들이 제시되어 왔는데 BP 알고리즘에서는 경사 descent method를 사용하게 된다. 여러 가지 활성 함수중 여기서는 sigmoid 함수를 이용 하겠다.

BP 알고리즘에서 오차는 다음과 같이 정의된다.

$$E = \frac{1}{2} (d - y)^2 = \frac{1}{2} e^2 \quad (1)$$

$d$ 는 목표 출력,  $y$ 는 network에 의한 실제 출력이다.  $n$ 개의 입력 패턴을 가진  $p$ 개의 입력이 있다고 가정하면 오차는 다음과 같이 정의된다.

$$E_p = \frac{1}{2} \sum_{j=1}^n (d_{jp} - y_{jp})^2 = \frac{1}{2} \sum_{j=1}^n e_{jp}^2 \quad (2)$$

$$E = \sum_p E_p = \frac{1}{2} \sum_p \sum_j (d_{jp} - y_{jp})^2 \quad (3)$$

오차를 이용한 가중치의 갱신은 다음과 같다.

$$W(k+1) = W(k) + \Delta w(k+1) \quad (4)$$

$$\Delta w = -\eta \frac{\partial E}{\partial w} x, \quad \eta > 0 \quad (5)$$

$\eta$ 는 학습률이고,  $x$ 는 입력값이다. 기본적인 BP 알고리즘의 경우 입력 패턴이 복잡한 경우 지역 최소값으로 수렴하여 학습이 더 이상 진행되지 않는 경우가 발생하는 문제점이 있다. 이러한 문제를 해결하기 위해서 Jacobs에 의해 변동 학습률을 적용한 delta-bar-delta 알고리즘과 Toll-enaere등에 의한 super SAB, Riedmiller등에 의한 RPROP 알고리즘 등이 제시 되었다.[4] 이들의 공통된 특징은 현재와 과거의 오차의 경사값을 이용하여 학습률을 적응적으로 변화 시킨다는 것이다. 오차의 경사값을 이용하여 변동 학습률을 적용하면 학습률  $\eta$ 는 다음과 같이 정의될 수 있다.

$$\eta(k+1) = \begin{cases} 1.04\eta(k) & \nabla E(k) > 1.04\nabla E(k-1) \\ 0.7\eta(k) & \nabla E(k) < \nabla E(k-1) \\ \eta(k) & \text{그 외의 경우} \end{cases} \quad (6)$$

여기에 과거의 변화량을 이용하는 모멘텀을 적용하면 다음과 같이 가중치의 변화량이 정의 된다.

$$\Delta w(k+1) = \alpha \Delta w(k+1) + (1-\alpha) \Delta w(k) \quad (7)$$

### 3. 퍼지이론의 파라미터 갱신에의 적용

변동 학습률과 모멘텀의 적용은 학습의 효과를 향상시킨다. 그러나 제시된 학습률의 가변적 적용은 오차 경사값의 부호에만 의존하여 고정된 몇 가지의 상수값을 이용하는 방법을 사용하며, 모멘텀은 오차와 무관하게 일정한 상수 값으로 정해지게 된다. BP 알고리즘의 성능을 개선하기 위해 오차의 경사 값과 그 변화율을 이용하여 학습률과 모멘텀에 대한 퍼지 룰 베이스를 작성하고 기존의 파라미터 적용 방법을 오차의 경사값과 그 변화율이 작은 곳에서 민감하게 변화시켜 학습의 효과를 증대 시키고자 하였다. 학습률은 일반적으로 오차가 감소하는 경우 증가시키고, 반대의 경우 줄여주는 방법을 이용한다. 대개의 경우 학습률의 변화 범위를  $0.7\eta$ ,  $1.05\eta$ 를 하한과 상한으로 잡는다. 여기에서는 오차의 경사값의 부호가 바뀌는 곳을 민감하게 처리했으며 학습률 변동의 폭을 하한  $0.55\eta$ , 상한  $1.20\eta$ 으로 잡아 변동의 폭을 크게 설정했다. 모멘텀의 경우 오차의 변화가 작은 경우 값을 증가시키고 변화가 큰 경우 감소시킨다. 실험적으로  $\alpha=0.95$ 의 경우가 가장 좋은 성능을 내는 것으로 알려져 있다. 여기서의 하한  $\alpha=0.8$ , 상한  $\alpha=0.97$ 로 설정했다.

### 3.1 룰 베이스의 설정

학습률 갱신의 경우 표1과 그림1과 같이, 모멘텀의 경우는 표2와 그림2와 같이 설정했다. 그림 1,2는 룰 베이스들을 그래프로 나타낸 것이다.

$\eta$	$\nabla E'$							
	NVB	NB	NM	NS	PS	PM	PB	PVB
NVB	1.200	1.175	1.150	1.125	0.625	0.600	0.575	0.550
NB	1.175	1.150	1.125	1.100	0.650	0.625	0.600	0.575
NM	1.150	1.125	1.100	1.075	0.675	0.650	0.625	0.600
NS	1.125	1.100	1.075	1.100	0.910	0.675	0.650	0.625
PS	0.625	0.650	0.675	0.910	1.100	1.075	1.100	1.125
PM	0.600	0.625	0.650	0.675	1.075	1.100	1.125	1.150
PB	0.575	0.600	0.625	0.650	1.100	1.125	1.150	1.175
PVB	0.550	0.575	0.600	0.625	1.125	1.150	1.175	1.200

표 1. 학습률을 위한 룰 베이스

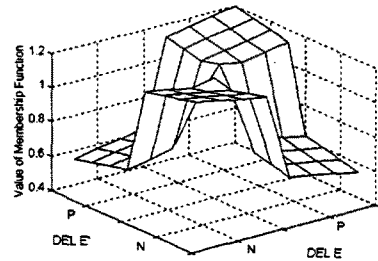


그림 1. 학습률의 룰 베이스

$\alpha$	$\nabla E'$							
	NVB	NB	NM	NS	PS	PM	PB	PVB
NVB	0.800	0.800	0.800	0.800	0.800	0.800	0.800	0.800
NB	0.850	0.850	0.850	0.850	0.850	0.850	0.850	0.850
NM	0.910	0.910	0.910	0.910	0.910	0.910	0.910	0.910
NS	0.970	0.970	0.970	0.970	0.970	0.970	0.970	0.970
PS	0.970	0.970	0.970	0.970	0.970	0.970	0.970	0.970
PM	0.910	0.910	0.910	0.910	0.910	0.910	0.910	0.910
PB	0.850	0.850	0.850	0.850	0.850	0.850	0.850	0.850
PVB	0.800	0.800	0.800	0.800	0.800	0.800	0.800	0.800

표2. 모멘텀을 위한 룰 베이스

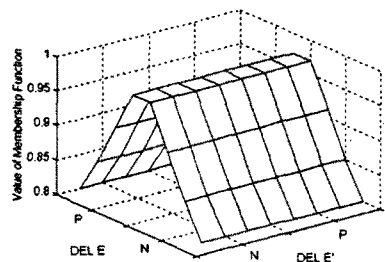


그림2. 모멘텀의 룰 베이스

Singleton fuzzyfier, product inference engine, center average defuzzyfier를 사용하여 구성했다.

#### 4. 시뮬레이션

BP 알고리즘의 성능을 평가하기 위해서 2개의 입력과 1개의 출력을 갖고, 2개의 은닉층을 갖는 다음과 같은 신경망을 고려했다.

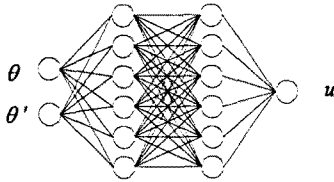


그림 3. 신경망의 구조

제시된 시스템을 도립 진자의 제어에 응용하기 위해서 각도와 각가속도를 두 입력으로, 도립 진자에 인가 되는 힘을 출력으로 설정했다. 입출력 패턴은 25개로 다음의 표와 같다.

입/출력	$\theta$					
	-1	-0.5	0	0.5	1	
$\theta'$	-1	3	2	1	0.5	0
	-0.5	2	1	0.5	0	-0.5
	0	1	0.5	0	-0.5	-1
	0.5	0.5	0	0.5	-1	-2
	1	0	-0.5	-1	-2	-3

표3. 학습에 사용되는 입출력 패턴

기본적인 BP 알고리즘, 변동 학습률 적용, 모멘텀의 적용, 변동 학습률과 모멘텀의 적용, 학습률에 퍼지를 적용한 경우, 학습률과 모멘텀에 모두 퍼지를 적용한 경우로 나누어서 오차가  $10^{-2}$  이내로 감소될 때까지 MATLAB을 이용하여 시뮬레이션 하였다.

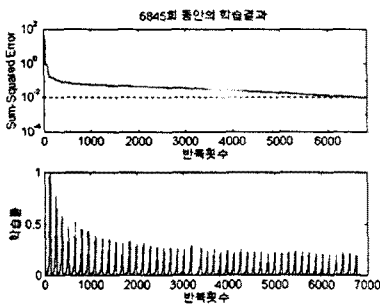


그림 4. 변동 학습률과 모멘텀을 적용한 경우 (6845회 반복 후 수렴)

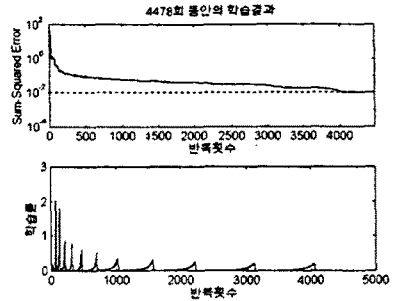


그림 5. 퍼지 변동 학습률과 모멘텀을 적용한 경우 (4478회 반복 후 수렴)

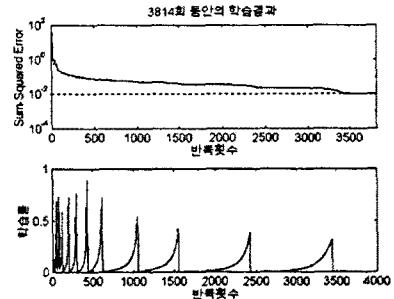


그림 6. 퍼지 변동 학습률과 퍼지 모멘텀을 적용한 경우 (3814회 반복 후 수렴)

#### 5. 결 론

퍼지를 적용한 경우 학습 횟수가 같은 입력에 대해서 반 정도로 감소했다. 그러나 퍼지를 이용하므로 파라미터 계산에 시간이 다소 걸리는 문제점과, 더 작은 오차를 고려한 경우 수렴하지 않는 신경망의 일반적인 문제가 발생했다. 거리까지 고려한 실제의 플랜트에 적용하는 문제와 오차를 더 줄이는 문제가 이후의 과제로 남는다.

#### 참고 문헌

- [1] Y. Hirose, K. Yamashita, S. Hijiya, "Back-Propagation Algorithm Which Varies the Number of Hidden Units", Neural Networks, Vol. 4, pp. 61-66, 1991
- [2] H. Demuth, M. Beal, "Neural Network Toolbox User's Guide", the Math Work, 1993
- [3] A. Cichocki, R. Unbehauen, "Neural Network for Optimization and Signal Processing", Wiley, 1993
- [4] P. D. Wasserman, "Advanced Method in Neural Computing", ITP, 1993
- [5] P. Arabshahi, J. J. Choi, R. J. Mark II, T. P. Caudell, "Fuzzy Parameter Adaption in Optimization: Some Neural Net Training Example", IEEE Computational Science & Engineering, pp.57-65, 1996
- [6] L. X. Wang, "A Course in Fuzzy System and Control", Prentice Hall, 1997