

오류 역전도 알고리즘의 학습속도 향상기법

심 범 식[†] · 정 의 용^{††} · 윤 충 화^{†††} · 강 경 식^{††††}

요 약

다층신경회로망의 학습방법인 오류역전도 알고리즘은 연관기억장치, 음성인식, 패턴인식, 로봇틱스등과 같은 다양한 응용분야에 널리 사용되고 있다. 그럼에도 불구하고 계속 많은 논문들이 역전도 알고리즘에 대해 발표되고 있는 실정이다. 이러한 연구 동향의 주된 이유는, 뉴런 갯수와 학습 패턴의 갯수가 큰 경우에 역전도 알고리즘의 학습속도가 상당히 느리다는 사실때문이다.

본 연구에서는 가변학습율, 가변모멘텀율, 그리고 시그모이드 함수의 가변기울기를 이용한 새로운 학습속도 가속기법을 개발하였다. 학습이 수행되는 도중에, 이러한 파라미터들은 전체 오류의 변화량에 따라 연속적으로 조정되며, 제안된 기법은 기존의 역전도 알고리즘에 비해 획기적으로 학습시간을 단축시키는 결과를 보였다. 제안된 기법의 효율성을 입증하기 위하여, 처음에는 난수발생기로 생성한 이진 데이터를 이용하여 에포크(epoch) 횟수를 비교할 때 훌륭한 속도 향상을 보였으며, 또한, 기계학습(machine learning)의 벤치마크 학습자료로 많이 사용되는 이진 Monk's data, 4, 5, 6, 7비트 패리티 검사 문제와 실수 Iris data에도 적용하였다.

An Enhancement of Learning Speed of the Error-Backpropagation Algorithm

Bum Sik Shim[†] · Eui Yong Jung^{††} · Chung Hwa Yoon^{†††} · Kyung Sik Kang^{††††}

ABSTRACT

The Error BackPropagation (EBP) algorithm for multi-layered neural networks is widely used in various areas such as associative memory, speech recognition, pattern recognition and robotics, etc. Nevertheless, many researchers have continuously published papers about improvements over the original EBP algorithm. The main reason for this research activity is that EBP is exceeding slow when the number of neurons and the size of training set is large.

In this study, we developed new learning speed acceleration methods using variable learning rate, variable momentum rate and variable slope for the sigmoid function. During the learning process, these parameters should be adjusted continuously according to the total error of network, and it has been shown that these methods significantly reduced learning time over the original EBP. In order to show the efficiency of the proposed methods, first we have used binary data which are made by random number generator and showed the vast improvements in terms of epoch. Also, we have applied our methods to the binary-valued Monk's data, 4, 5, 6, 7-bit parity checker and real-valued Iris data which are famous benchmark training sets for machine learning.

※ 본 연구는 한국과학재단 1995년도 핵심연구과제 지원 (과제 번호: 951-0906-086-1)에 의하여 수행되었음.

† 종신회원: 명지대학교 컴퓨터공학과

†† 정 회 원: 명지대학교 컴퓨터공학과

††† 정 회 원: 명지대학교 컴퓨터공학과

†††† 정 회 원: 명지대학교 산업공학과

논문접수: 1997년 2월 18일, 심사완료: 1997년 6월 23일

1. 서론

신경회로망(neural networks)의 이론은 폰노이만(Von Neumann)형 컴퓨터와 같이 1940년대에 발표되었으나 민스키(Minsky)와 페퍼트(Papert)가 퍼셉트론의 한계를 지적함으로써 침체되었다가 호프필드(Hopfield)의 연상메모리 모델의 제시로 신경회로망에 대한 연구가 활기를 띠기 시작했다. 또한 다층 신경회로망은 퍼셉트론의 한계를 극복하기 위하여 제안되었고 특히, PDP(Parallel Distributed Processing)그룹에 의해 폭넓은 연구가 진행되었다.

신경회로망의 구조나 적용 분야에 따라 여러 가지 학습 알고리즘이 연구되었으나 일반적으로 역전도 알고리즘을 이용한 다층 퍼셉트론 모델이 가장 널리 사용되고 있다. 역전도 알고리즘은 연관기억장치(Associative Memory), 음성인식, 패턴인식, 로보틱스등 여러 응용분야에 다양하게 사용되고 있다. Minsky와 Papert에 의해 지적되었던 비선형분리문제(Non-linearly Separable Problem)를 은닉뉴런층을 이용하여 해결할 수 있다는 장점에에도, 알고리즘이 간결하고 안정성(Stability)이 수학적으로 증명되어 있다는 점들을 수있다. 그럼에도 불구하고 계속 많은 논문들이 역전도알고리즘에 대해 발표되는 주된 이유는, 입력층의 뉴런 갯수와 저장하려는 패턴의 갯수가 증가할 수록 학습시간이 많이 소요된다는 사실때문이다.

80년대 중반이후 발표된 역전도 알고리즘의 학습 속도를 향상시키려는 연구를 살펴 보면, 비선형 활성화함수의 기울기를 최적으로 구하는 연구[1], 오류표면분석에 의한 경험적 지식(heuristics)을 이용한 학습율의 조정[2, 3], 오류의 2차 도함수를 이용한 기법[4], 비학습(Unlearning)개념을 이용한 연구[5], 학습과정을 기하학적으로 분석한 출력 중심벡터를 이용한 방법[6], eigenvalue를 이용하여 활성화함수의 기울기를 조정하는 방법[7], 학습하는 도중에 발생하는 오류의 진동 현상을 최소화 하는 기법[8], 오류를 수정할 때 선형 갱신함수를 이용하는 기법[9] 등이 있다. 위에 언급된 기법들은 파라미터들을 결정하기 위한 계산 부하가 과도하거나, 기존 역전도 알고리즘 자체를 수정하는 시도인데 반하여, 본 연구에서 제안한 기법은 전체 오류의 변화량을 기준으로 하여 파라미터들을 연속적으로 조정하므로 계산 부하가 걸리지 않는 간

편한 기법이다.

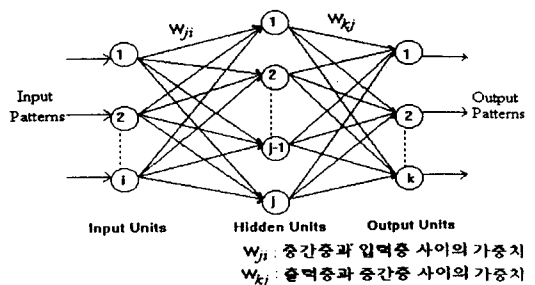
본 논문의 구성은 다음과 같다. 2절에서는 역전도 알고리즘과 가변 학습율, 가변 모멘텀율, 가변 기울기 알고리즘을 설명하였으며, 3절에서는 성능 평가로써 임의의 이진자료를 이용한 실험과 기계학습의 벤치마크 자료인 Monk's data, 4-7 비트 패리티 검사 문제와 Iris data를 이용한 실험 결과를 분석하여 제안된 기법의 효율성을 입증하였다. 마지막으로 4절에서는 결론과 향후 연구 과제를 제시하였다.

2. 역전도 알고리즘의 학습시간 단축기법

2.1 역전도 알고리즘

다층 신경회로망의 구조는 입력층과 출력층 그리고 그 사이에 하나 이상의 중간층(은닉층)으로 구성된다. 각 층은 뉴런으로 구성되고 각 층에 있는 뉴런은 인접층의 뉴런들과 연결되며 각 연결은 가중치를 갖는다.

가중치에 대한 오류의 최급강하법(gradient descent method)을 사용하여 반복적 계산으로 오류를 최소화 하면서 학습하는 델타 규칙을 다층신경망에 일반화 시킨 오류 역전도 학습 알고리즘(error backpropagation) [10]은 각 노드에 제시된 패턴에 대하여 활성화 함수(activation function)를 이용하여 출력(O_j)을 구하고, 요구되는 출력(T_j)과 실제 출력(O_j)과의 차이를 계산하여 이 차이를 출력층에서 입력층으로 역전도시키면서 층과 층 사이의 가중치(weight)를 갱신한다. (그림 1)은 다층 구조로 된 신경회로망을 표현하고 있다.



(그림 1) 다층 신경회로망의 구조

역전도 알고리즘에 사용되는 활성화 함수는 다음

과 같은 시그모이드(sigmoid)함수를 사용한다.

$$f_j(\text{net}_{pj}) = \frac{1}{1 + e^{-(\text{net}_{pj} + \theta_j)}} \quad (1)$$

$$\text{net}_{pj} = \sum_j W_{ji} O_{pi} \quad (2)$$

$$O_{pj} = f_j(\text{net}_{pj}) \quad (3)$$

여기서 W_{ji} 는 이전층의 i 번째 뉴런과 다음층의 j 번째 뉴런간의 연결 가중치(weight)를 나타낸다. O_{pj} 는 p 번째 패턴의 j 번째 뉴런의 출력이고, θ_j 는 임계치(threshold)이다. 출력층에서 계산되는 실제 출력값과 요구되는 출력값 사이의 오류는 다음과 같이 구할 수 있다.

$$E_p = 0.5 * \sum_j (T_{pj} - O_{pj})^2 \quad (4)$$

$$E = \sum E_p \quad (5)$$

식 4는 학습되는 출력층에서 발생하는 p 번째 패턴의 오류값을 나타내고, 식 5는 각 패턴의 오류를 합산한 전체 오류이다. T_{pj} 는 출력층에서 p 번째 패턴의 j 번째 뉴런의 목표값이다. 계산된 전체 오류가 정의된 오류, 예를 들어 10^{-4} 이나 10^{-5} , 보다 작을 때까지 학습을 반복하게 된다.

출력층과 은닉층 사이의 가중치 변화량 δ_{pj} 는 다음과 같다.

$$\delta_{pj} = (T_{pj} - O_{pj}) \cdot O_{pj} \cdot (1 - O_{pj}) \quad (6)$$

은닉층과 입력층 사이의 가중치 변화량 δ_{pi} 는 다음과 같다.

$$\delta_{pi} = O_{pj} \cdot (1 - O_{pj}) \cdot \sum_k \delta_{pk} \cdot W_{kj} \quad (7)$$

식 6과 식 7을 이용하여 수정되는 가중치의 변화량은 다음과 같다.

$$\Delta_p W_{ji}(n+1) = \eta \delta_{pj} O_{pi} + \alpha \Delta_p W_{ji}(n) \quad (8)$$

여기서 η 는 학습율(learning rate)를 나타내고, α 는

모멘텀율(momentum rate)를 나타낸다.

기본적으로 역전도 알고리즘은 기울기 감소(gradient descent) 기법을 이용하여 식 5의 전체 오류를 최소화하는 학습기법이다. 이때, 가중치는 식 8에 의하여 변화되며, 아주 작은 양만큼씩 변화하면서 오류평면의 전역 최소치로 수렴해 가게 된다. 이 변화량을 제어하는 매개변수의 역할을 학습율이 담당한다. 실제로 학습율은 클수록 빠른 학습이 가능하나, 너무 크게 되면 전역 최소치를 사이에 두고 진동(oscillation)이 일어나는 경우를 관찰할 수 있다. 모멘텀율은 학습속도를 향상시키기 위하여 사용되는데, 이전에 가중치가 변화하였던 양(history of weight change)을 금번 가중치 수정시, 얼마만큼 고려할 것인가를 결정하는 매개변수이다. 보편적으로 학습율과 모멘텀율은 0과 1사이의 실수를 사용한다.

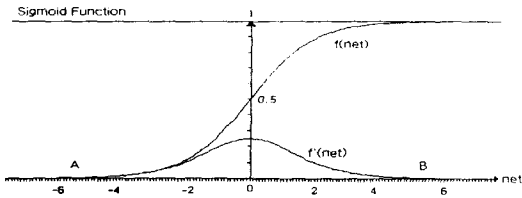
전체적인 학습 과정은 다음과 같다.

- ① 초기 가중치를 임의로 선정한다.
- ② 입력패턴에 대해서 입력층에서 출력층까지 식 1, 2, 3에 의해 각 뉴런의 출력값을 구한다.
- ③ 식 4, 5를 이용하여 전체 오류를 구하여 정의된 오류보다 작으면 학습을 마친다.
- ④ 식 6, 7을 이용하여 층과 층사이의 가중치 변화량을 계산하고, 식 8을 적용하여 새로운 가중치를 구한다.
- ⑤ ②로 가서 반복 수행한다.

2.2 가변 학습율(variable learning rate)

기존 역전도 알고리즘의 가중치 변화량을 나타내는 식 8의 η 는 0과 1사이의 실수값을 갖는 고정 상수이다. 그렇기 때문에 가중치의 변화량 또한 출력값과 오류에 비례하여 일정한 비율로 변화한다. 문제점은 학습과정중 고정된 학습율을 사용하면 (그림 2)의 A와 B 영역에서, 즉 목표값에 거의 근접한 부분에서 대부분의 시간을 소비한다는 점이다.

이는 학습이 진행됨에 따라 0 또는 1로 수렴해 가는 데, A와 B 영역에서 f' (식 6과 7에서)의 값이 거의 0이므로 층과 층 사이의 가중치 변화량(δ)이 작아진다. 따라서 가중치가 거의 변하지 않고, 이로 인해 시그모이드 함수의 출력도 이전의 출력에 비하여 크게 변하지 못한다. 결국, 0과 1에 수렴해 갈수록 오류는 아주 조금씩 변하게 되므로, A와 B 영역에서 많은 시간



(그림 2) 시그모이드 함수

을 소비하게 된다.

가변 학습을 방법은 η 를 가변적으로 증가시킴으로써 A와 B 부분에서 학습을 빠르게 하는 기법이다. 초기에는 학습율을 작게 하였다가 오류가 수렴상태로 접어드는 적당한 시점을 설정하여 식 9를 이용하여 학습률을 가변적으로 증가 시켜준다.

$$\eta = (\log(E))^2 \cdot k \tag{9}$$

식 9에서 k 는 학습에서 최초로 사용했던 학습율이고 <표 1>은 역전도 학습 알고리즘에서 오류에 따른 학습율의 변화를 보인 것이다. 초기 학습시 전체 오류가 0.1이 될 때까지의 학습율은 0.6으로 고정시켜 수행하다가 오류가 0.1미만으로 될 때부터 식 9에 의해서 학습율이 커지기 시작하며 학습이 끝날 무렵에는 기존의 역전도 알고리즘에서 학습율의 최대치인 1.0보다 훨씬 큰 값으로 변화한다.

표 1. 오류의 변화에 따른 학습율의 변화

오류의 변화	학습율의 변화
~ 0.1	0.6
0.1 ~ 0.01	0.6 ~ 2.4
0.01 ~ 0.001	2.4 ~ 5.4
0.001 ~ 0.0001	5.4 ~ 9.6

2.3 가변 모멘텀율(variable momentum rate)

기존 역전도 알고리즘의 가중치 변화량을 나타내는 식 8의 α 는 0.0과 1.0사이의 값을 갖는 고정 상수이다. 학습율과 유사하게 가중치의 변화량을 크게 해주기는 하지만 모멘텀은 새로운 가중치의 변화량을 구할 때 전번의 변화량을 얼마나 고려할 것인지를 결정해 주는 상수이다. 따라서 학습율의 경우와 같이 오

류표면을 변화시키지는 않지만 모멘텀을 또한 초기에 너무 큰 값을 주면 가중치의 변화가 크게 되어 오류의 최소치를 경계로 새로 구해지는 가중치 값의 부호를 바꾸게 되어 진동을 초래하게 된다. 그러므로 학습의 초기 단계에서는 오류 표면을 가로지르는 진동을 방지하기 위하여 0.6를 초기치로 하여 학습을 하다가 오류가 수렴상태로 접어드는 적당한 시점을 설정하여 식 10을 이용하여 모멘텀율을 가변적으로 증가 시켜준다.

$$\alpha = 1 - \frac{(1-k)}{((\log(E))^2 + 1)} \tag{10}$$

식 10에서 k 는 학습에서 최초로 사용했던 모멘텀을 이고 <표 2>는 역전도 학습 알고리즘에서 전체 오류에 따른 모멘텀율의 변화를 보인 것이다. 초기 학습시 오류가 0.01이 될 때까지의 모멘텀율은 0.6으로 유지하다가 오류가 0.1미만으로 될 때부터 식 10에 의해서 모멘텀율이 커지기 시작하며 학습이 끝날 무렵에는 모멘텀율의 최대치인 1.0에 가깝게 된다.

표 2. 오류의 변화에 따른 모멘텀율의 변화

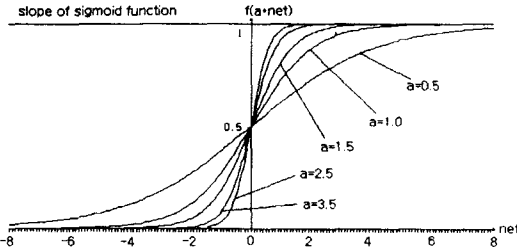
오류의 변화	모멘텀율의 변화
~ 0.1	0.6
0.1 ~ 0.01	0.6 ~ 0.92
0.01 ~ 0.001	0.88 ~ 0.96
0.001 ~ 0.0001	0.94 ~ 0.97647
0.0001 ~ 0.00001	0.96471 ~ 0.98462

2.4 가변 기울기(variable slope)

앞에서 설명한 바와같이 역전도 알고리즘의 단점 중의 하나는 학습이 종료되는 데 많은 시간이 걸린다는 것이다. 이와같은 문제점을 해결하기 위하여 시그모이드(sigmoid) 함수의 기울기를 발생하는 전체 오류에 따라 적절하게 변화시킴으로써 학습 수행 시간을 최소화한다.

역전도 알고리즘에서, 학습이 진행됨에 따라 출력층의 각 뉴런 출력값이 0 또는 1에 점점 수렴해 가기 때문에 시그모이드 함수의 기울기를 점점 크게 한다면 출력층에서 각 뉴런의 출력이 요구되는 출력(0 또는 1)에 더욱 가깝게 되어 수렴하는 속도가 빨라질 것

이다.



(그림 3) 시그모이드 함수의 기울기

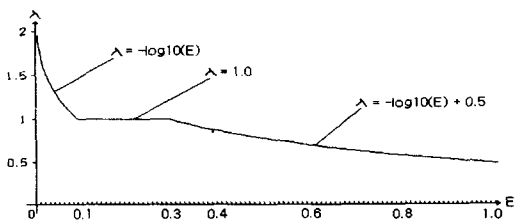
그러나, 기울기를 단순히 크게 한다고 해서 학습이 빨리 끝나는 것이 아니다. 왜냐하면 가중치의 계속적인 변화, 패턴들 간의 관계, 각 층의 뉴런 갯수 등이 복합적으로 관계하고 있기 때문이다. 따라서, 학습이 빨리 종료되면서 다른 알고리즘보다 효율이 떨어지지 않도록 하기 위하여 다음의 수식을 이용한다. 식 11와 식 12은 학습이 수행되는 과정에서 시그모이드 함수의 기울기를 전체 오류에 따라 변화되도록 하였다.

$$\begin{cases} \lambda = -\log_{10}(E), & E \leq 0.1 \text{ 일때} & (11) \\ \lambda = -\log_{10}(E) + 0.5, & 0.3 < E < 0.1 \text{ 일때} & (12) \end{cases}$$

식 12에서 0.5를 더해주는 이유는 오류가 0.3에서 1.0 사이에 있을 때 $-\log(E)$ 의 범위가 0.507에서 0.0046이 되기 때문에 시그모이드 함수의 출력값 변화폭이 아주 미세하게 되어 학습 효율을 저하시킨다. 따라서 적당한 기울기를 정해주기 위해 0.5을 더해주었다.

(그림 4)는 전체 오류값과 λ 사이의 관계를 그래프로 표현한 것이다.

식 11, 12에 따라 시그모이드 함수에 λ 를 추가하면 다음의 식과 같이 된다.



(그림 4) 전체 오류에 따른 시그모이드 함수의 기울기 변화

$$f_j(\text{net}_{pj}) = \frac{1}{1 + e^{-\lambda \cdot \text{net}_{pj}}} \quad (13)$$

λ 의 추가로 역전도 알고리즘에서 바뀌는 수식은 다음과 같다. 식 6의 수식은 식 6'로 수정되며,

$$\delta_{pj} = (T_{pj} - O_{pj}) \cdot \lambda \cdot O_{pj} \cdot (1 - O_{pj}) \quad (6')$$

식 7은 식 7'로 바뀌게 된다.

$$\delta_{pj} = \lambda \cdot O_{pj} \cdot (1 - O_{pj}) \cdot \sum_k \delta_{pk} \cdot W_{kj} \quad (7')$$

학습이 진행됨에 따라 시그모이드 함수의 기울기 (λ)는 점점 커진다. 결국, f' 가 0에 가까워지더라도 λ 값이 식 6'와 식 7'처럼 곱해지기 때문에 층과 층 사이의 가중치 변화량이 커지게 된다.

3. 성능 평가

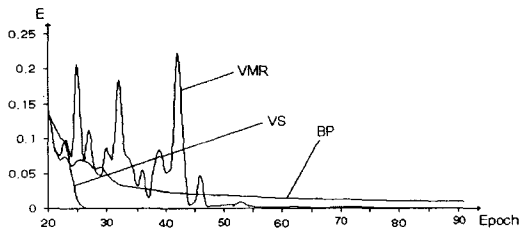
제안된 기법의 성능을 평가하기 위하여 3가지의 실험을 수행하였다. 첫째로 난수발생기로 생성한 임의 이진자료를 사용하였으며, 둘째로 기계학습의 벤치마크로 많이 사용되는 Monk's data[11], 4-7비트 패리티 자료와 Iris data를 이용하여 제안된 학습속도 향상기법의 효율성을 검증하였다. 실험 환경은 Pentium PC와 SUN Sparcstation 20으로 구성되었으며, 프로그램은 C언어로 작성하였다.

3.1 임의 자료를 이용한 실험

실험에 사용한 신경회로망의 구조는 은닉층이 1개로, 각 층의 뉴런 갯수는 동일한 모델을 이용하였으며, 역전도 알고리즘의 학습율은 0.7, 모멘텀율은 0.8, 그리고 전체 오류의 허용 한계는 0.0001로 정의하였다. <표 3>부터 <표 6>에 나타난 M*N의 의미는 다음과 같다: M은 단일 패턴의 크기(뉴런 갯수), N은 전체 학습패턴의 갯수를 나타낸다. 예를 들어 7*10의 경우, 단일 패턴이 가진 특징의 갯수가 7개인 이진 패턴 10개를 난수발생기를 이용하여 생성한 후, 이를 신경회로망에 학습시킨다. 그 다음에 모든 가능한 이진 패턴의 갯수인 128개(=2⁷)로 일반화 성능을 검사하였다.

이러한 실험을 50번 반복하여 그 평균 수치를 표에

포함시켰으며, epoch는 학습의 횟수를 의미하는 용어로써, 이는 모든 학습 패턴이 알고리즘에 의해 순방향과 역방향 처리과정을 거친 것을 1회로 간주한다. 비교 분석된 기법으로는, BP는 기존의 역전도 알고리즘, VLR은 가변 학습을 알고리즘, VMR은 가변 모멘텀을 알고리즘, 그리고 VS는 가변 기울기 기법을 대상으로 하였다. 인식 성능을 검사하기 위하여 가능한 모든 패턴인 2^M 개를 이용하였으며, 임의의 패턴을 입력시켜 얻은 출력 패턴을 구한 뒤, 저장된 학습 패턴과의 해밍 거리(Hamming distance)를 구하였다. 해밍 거리가 가장 최소인 패턴으로 수렴한 경우는 올바른 답(Correct answer)으로, 저장은 되었으나 거리가 최소가 아닌 경우는 틀린 답(Wrong answer)로 분류하였으며, 그외에 본래 저장하지 않았던 패턴으로 수렴한 경우는 모호한 답(Spurious answer)로 간주하였다. 또한 출력 값이 0.4와 0.6 구간에 속해서 이진화하기 어려운 경우, 그 입력 패턴은 미결정(Undetermined)으로 구분하였다. 학습이 진행되면서, 학습 횟수에 따



(그림 5) 학습 횟수에 따른 전체 오류의 변화

<표 3>

M*N : 7*10

	Epoch	Correct	Wrong	Spurious	Undeter
BP	6990.0	39.9	2.5	85.6	28.6
VLR	256.4	40.1	3.3	84.6	27.3
VMR	62.4	43.5	4.6	80.0	23.4
VS	45.6	49.5	9.3	69.2	12.1

<표 4>

M*N : 8*10

	Epoch	Correct	Wrong	Spurious	Undeter
BP	9143.8	65.1	3.3	187.6	73.5
VLR	284.2	62.3	4.0	189.8	75.1
VMR	60.4	65.7	4.5	185.8	66.2
VS	34.2	76.2	13.8	165.9	31.2

라 전체 오류가 변화하는 상황은 다음 그림에 도시되었다.

<표 5>

M*N : 9*10

	Epoch	Correct	Wrong	Spurious	Undeter
BP	10102.5	129.5	7.7	374.8	172.5
VLR	290.3	126.4	5.7	379.9	178.1
VMR	61.7	127.3	10.0	374.8	159.4
VS	34.7	144.6	21.8	345.6	70.8

<표 6>

M*N : 10*10

	Epoch	Correct	Wrong	Spurious	Undeter
BP	11330.6	222.9	14.5	786.5	400.7
VLR	280.4	210.5	13.6	799.9	389.5
VMR	68.4	224.8	18.0	781.2	366.8
VS	33.3	245.1	35.2	742.9	165.6

3.2 Monk's data를 이용한 실험

기계학습의 벤치마크로 많이 사용되는 Monk's data [11]는 제반 기계학습 기법의 평가를 위해 인위적으로 만든 이진수 형태의 자료로써, 특징값에 의해 2개의 출력 클래스로 분류되는 로봇의 분류 문제이다. Monk's data는 다음과 같은 특징값을 갖는 로봇의 외관을 표현한다.

- x_1 : head_shape {round, square, octagon}
- x_2 : body_shape {round, square, octagon}
- x_3 : is_smiling {yes, no}
- x_4 : holding {sword, balloon, flag}
- x_5 : jacket_color {red, yellow, green, blue}
- x_6 : has_tie {yes, no}

각 패턴의 출력 클래스는 0 또는 1이며, 1이면 로봇임을 나타내게 된다. Monk's의 문제란 주어진 기준을 가지고 출력 클래스를 부여한 후, 432개의 전체 패턴 중 일부를 임의로 추출하여 학습을 수행한 다음, 학습시키지 않은 나머지 패턴들로 분류 성능을 평가하는 문제이다. 학습 패턴의 갯수 124개는 Monk's 자료를 만든 저자[11] 등이 추천한 수치이며, 결과를 보면 일관되게 90개의 경우보다 인식성능이 우수한 것을 볼 수 있다.

아래의 표에 나타난 결과는 동일한 실험을 5번씩 반복한 후, 그 결과를 평균낸 수치이다. 또한 은닉층 뉴런 갯수가 학습 시간과 인식 성능에 미치는 영향을 보기 위하여, 은닉층 뉴런의 갯수를 12개와 15개로 설정하여 실험하였다. 초기 학습율은 0.8, 모멘텀율은 0.7을 사용하였으며, 학습은 전체 오류가 0.0001보다 작아지면 종료하였다.

문제 1: 다음의 조건을 만족하는 패턴은 클래스가 1이며, 아니면 0이 된다.

(head_shape=body_shape) or (jacket_color = red)

신경망의 구조: 입력뉴런 = 6개, 은닉뉴런 = 12개, 출력뉴런 = 2개

<표 7>

학습 기법	학습 횟수 (Epoch)		인식 성능 (%)	
	학습 패턴의 갯수		학습 패턴의 갯수	
	90	124	90	124
역전도 알고리즘	13,920.7	13,219.2	93.0%	100%
가변 학습율	1,515.0	377.0	90.4%	99.0%
가변 모멘텀율	759.7	368.6	90.3%	98.5%
가변 기울기	409.7	80.0	86.8%	98.0%

신경망의 구조: 입력뉴런 = 6개, 은닉뉴런 = 15개, 출력뉴런 = 2개

<표 8>

학습 기법	학습 횟수 (Epoch)		인식 성능 (%)	
	학습 패턴의 갯수		학습 패턴의 갯수	
	90	124	90	124
역전도 알고리즘	13,322.4	13,244.6	92.3%	98.8%
가변 학습율	388.2	354.2	92.7%	99.4%
가변 모멘텀율	385.2	356.8	92.9%	98.9%
가변 기울기	96.4	76.8	87.5%	98.5%

위의 표에서 나타난 대로, 학습 속도는 가변 기울기가 가장 빨랐으며, 학습 패턴의 갯수가 많을수록 인식율이 높아지는 것을 볼 수 있었다. 또한 은닉층의 뉴런 갯수 12개와 15개 모델을 비교하면, 전반적으로 15개인 모델의 학습 속도와 인식 성능이 나은 것을 볼 수 있다.

문제 2: 다음의 조건을 만족하는 패턴은 클래스가 1이

며, 아니면 0이 된다.

Exactly two of the six attributes have their first value.

신경망의 구조: 입력뉴런 = 6개, 은닉뉴런 = 12개, 출력뉴런 = 2개

<표 9>

학습 기법	학습 횟수 (Epoch)		인식 성능 (%)	
	학습 패턴의 갯수		학습 패턴의 갯수	
	90	124	90	124
역전도 알고리즘	21,138.0	19,499.2	75.0%	98.3%
가변 학습율	4433.3	745.2	75.2%	98.0%
가변 모멘텀율	884.3	590.6	71.9%	99.1%
가변 기울기	230.6	140.0	76.4%	96.5%

신경망의 구조: 입력뉴런 = 6개, 은닉뉴런 = 15개, 출력뉴런 = 2개

<표 10>

학습 기법	학습 횟수 (Epoch)		인식 성능 (%)	
	학습 패턴의 갯수		학습 패턴의 갯수	
	90	124	90	124
역전도 알고리즘	29,639.0	19,378.7	71.2%	99.0%
가변 학습율	1,816.0	1639.0	71.5%	92.2%
가변 모멘텀율	1048.6	1019.0	74.4%	94.4%
가변 기울기	206.2	123.7	73.7%	94.2%

문제 3: 다음의 조건을 만족하는 패턴은 클래스가 1이며, 아니면 0가 된다.

(jacket_color is green & holding a sword) or (jacket_color is not blue & body_shape is not octagon)

신경망의 구조: 입력뉴런 = 6개, 은닉뉴런 = 12개, 출력뉴런 = 2개

<표 11>

학습 기법	학습 횟수 (Epoch)		인식 성능 (%)	
	학습 패턴의 갯수		학습 패턴의 갯수	
	90	124	90	124
역전도 알고리즘	19,368.5	14,336.5	83.3%	88.6%
가변 학습율	2,001.0	619.5	71.6%	88.5%
가변 모멘텀율	1,179.0	560.5	89.5%	89.7%
가변 기울기	1,021.5	96.0	87.7%	88.2%

신경망의 구조: 입력뉴런 = 6개, 은닉뉴런 = 15개, 출력뉴런 = 2개

<표 12>

학습 기법	학습 횟수 (Epoch)		인식 성능 (%)	
	학습 패턴의 갯수		학습 패턴의 갯수	
	90	124	90	124
역전도 알고리즘	12,972.0	10,442.8	90.0%	90.2%
가변 학습율	1,606.3	842.3	90.2%	90.3%
가변 모멘텀을	1569.3	454.3	89.1%	89.3%
가변 기울기	517.8	128.8	86.4%	89.2%

위의 실험 결과를 보면, 가변 기울기의 학습 기법이 학습 속도면에서 가장 우수하였으며, 인식 성능의

측면을 보면 별다른 차이가 없었다. 다만 가변 기울기의 경우, 실험 도중에 지역 최소치에 빠지는 경우가 간혹 있었으며, 이를 감지하여 그것으로부터 빠져 나오는 기법의 개발이 시급하다고 사료된다.

3.3 패리티 검사 문제에 적용한 실험

패리티 검사 문제는 이진 패턴에 패리티 비트를 추가하여 1의 갯수가 항상 짝수 또는 홀수가 되도록 하여 전송상의 오류를 발견하는 문제이다. 실험에 사용한 자료는 4, 5, 6, 7 비트 패턴으로 구성되며, 표에 나타난 수치는 동일한 실험을 10번씩 반복하여 그 결과를 평균낸 수치이다.

신경망의 구조: 입력뉴런 = N개 (N=4, 5, 6, 7), 은닉뉴런 = N*2개, 출력뉴런 = 1개
 임의 추출한 학습 패턴의 갯수: 4-bit:10개, 5-bit:20개, 6-bit:40개, 7-bit:80개

<표 13>

학습 기법	학습 횟수 (Epoch)				인식 성능 (%)			
	4-bit	5-bit	6-bit	7-bit	4-bit	5-bit	6-bit	7-bit
역전도 알고리즘	10,143.8	12,683.2	7,908.4	7,572.4	13.33%	25.0%	55.0%	54.58%
가변 학습율	777.2	564.6	674.6	423.2	13.33%	36.67%	54.17%	61.25%
가변 모멘텀을	1,053.4	968.6	932.0	684.0	10.0%	31.67%	47.50%	64.17%
가변 기울기	38.8	108.2	116.6	130.4	13.33%	38.33%	53.33%	57.5%

신경망의 구조: 입력뉴런 = N개 (N=4, 5, 6, 7), 은닉뉴런 = N*3개, 출력뉴런 = 1개
 임의 추출한 학습 패턴의 갯수: 4-bit:10개, 5-bit:20개, 6-bit:40개, 7-bit:80개

<표 14>

학습 기법	학습 횟수 (Epoch)				인식 성능 (%)			
	4-bit	5-bit	6-bit	7-bit	4-bit	5-bit	6-bit	7-bit
역전도 알고리즘	8,901.2	9,056.0	6,775.0	4,884.4	16.67%	23.33%	40.0%	64.17%
가변 학습율	514.2	561.6	55.83	348.8	13.33%	20.0%	40.66%	67.92%
가변 모멘텀을	1,014.8	777.8	655.0	385.8	10.0%	25.0%	54.17%	55.42%
가변 기울기	37.0	85.2	93.2	157.6	13.33%	28.33%	58.33%	70.0%

위의 실험에서는 학습 패턴을 임의로 추출하였으나, 4-bit 문제의 경우, 전체 16개 패턴중에서 하나를 제외한 나머지 15개 패턴으로 학습을 수행한 후, 제외 하였던 패턴으로 인식 성능을 검사하는 “leaving one out” 기법으로 다음의 실험을 수행하였다. 이러한 실험 기법은 패리티 검사 문제에 많이 사용되는 방법이다. 난수 발생기로 초기화하는 연결 가중치의 영향을 완화하기 위하여 동일한 실험을 5번씩 반복하여 평균 하였으므로, 표에 나타난 수치는 80번 실험의 결과를 평균한 결과이다.

<표 15>

학습 기법	학습 횟수 (Epoch)		인식 성능 (%)	
	중간 노드의 갯수		중간 노드의 갯수	
	8	12	8	12
역전도 알고리즘	18,262.49	12,111.14	18.75%	18.75%
가변 학습율	1,015.26	718.55	16.25%	13.75%
가변 모멘텀을	1,822.9	1,396.53	22.5%	22.5%
가변 기울기	441.93	84.4	21.25%	18.75%

3.4 Iris data를 이용한 실험

Iris data는 꽃잎에서 추출한 실수 형태의 자료로써, 하나의 패턴은 꽃받침과 꽃잎에 대한 길이와 폭 등 4개의 특징으로 구성되며, 출력 클래스는 setosa, versicolor, virginica의 3개로 주어진다. 각 클래스는 50개의 패턴으로 구성되며, setosa는 다른 클래스로 부터 잘 분리되어 있는 반면에, versicolor와 virginica는 3개의 중복된 패턴을 갖는다.

아래의 표에 나타난 결과는 동일한 실험을 5번씩 반복한 후, 그 결과를 평균낸 수치이다. 또한 은닉층 뉴런 갯수가 학습 시간과 인식 성능에 미치는 영향을 보기 위하여, 은닉층 뉴런의 갯수를 8개와 12개로 설정하여 실험하였다. 초기 학습율은 0.3, 모멘텀율은 0.7을 사용하였으며, 학습은 전체 오류가 0.001보다 작아지면 종료하였다.

신경망의 구조: 입력뉴런=4개, 은닉뉴런=8개, 출력뉴런=3개

<표 16>

학습 기법	학습 횟수 (Epoch)		인식 성능 (%)	
	학습 패턴의 갯수		학습 패턴의 갯수	
	75	100	75	100
역전도 알고리즘	1,586.5	1,533.2	93.3%	90.0%
가변 학습율	1,509.2	1,359.6	93.1%	92.8%
가변 모멘텀을	1,359.6	1,205.6	94.9%	94.0%
가변 기울기	920.4	943.0	93.6%	92.4%

신경망의 구조: 입력뉴런=4개, 은닉뉴런=12개, 출력뉴런=3개

<표 17>

학습 기법	학습 횟수 (Epoch)		인식 성능 (%)	
	학습 패턴의 갯수		학습 패턴의 갯수	
	75	100	75	100
역전도 알고리즘	1,616.0	1,537.8	93.3%	90.0%
가변 학습율	1,472.4	1,431.0	94.7%	90.0%
가변 모멘텀을	1,256.2	1,357.6	94.7%	94.0%
가변 기울기	1,126.6	949.0	93.1%	91.6%

위의 실험 결과를 보면, 실수 자료인 Iris data의 경우, 제안된 기법의 효율성이 이전의 이진 자료에 비해 월등히 저하된 결과가 주목된다. 그러므로 제안된 기법은 실수 형태의 자료보다는 이진 형태의 자료에 적합한 기법이라고 할 수 있다. 또한 실험 도중 발견된 현상중 하나를 살펴 보면, 학습 초기에 임의로 설정되는 가중치들의 초기값에 의해 학습 속도가 좌우되는 경우를 간혹 보았다. 이는 기존의 역전도 알고리즘에도 동일하게 적용되는 문제점이며, 이에 대한 연구, 즉 초기 연결 가중치의 효율적인 설정 기법에 대한 연구도 수행되어야 할 것이다.

4. 결 론

본 연구에서는 학습속도를 향상시키기 위하여 전체 오류의 변화량에 따라 가변적으로 변화하는 학습율, 모멘텀을, 그리고 시그모이드 함수의 기울기를 이용하여 기존의 역전도 알고리즘에 비해 획기적으로 학습시간을 단축시키는 기법을 개발하였고, 가변 기울기를 사용하는 기법의 성능이 가장 우수하였다. 개발된 기법의 효율성을 입증하기 위하여 처음에는 난수발생기를 이용한 임의의 데이터로 학습 속도의 향상

을 보였으며, 패턴인식 분야의 벤치마크 학습자료로 사용되는 Monk's data와 패리티 검사 문제에 적용하여 훌륭한 학습속도의 향상을 입증하였다. 이 결과는 이진 형태의 자료에 국한되며, 실수 자료인 Iris data의 경우, 제안된 기법의 효율성은 그다지 훌륭하지 못한 결과가 나왔다. 또한 제안된 방법은 경험적 지식(Heuristics)을 이용한 기법으로 분류될 수 있으므로, 수학적인 배경이 없다는 점이 단점으로 지적될 수 있다.

향후 수행해야 할 문제점으로는 역전도 알고리즘이 수행될 때, 난수 발생기를 이용하여 연결 가중치의 초기치를 부여하는데, 이 초기치에 따라 학습의 속도가 상당히 좌우되는 경우가 간혹 있었으며, 이에 대한 연구가 수행되어야 할 것이다. 또한 특히 Monk's data에 대한 실험중 지역 최소치에 빠져서 오류가 줄어들지 않고 학습이 지연되는 현상도 발생하였는데, 이러한 현상을 감지하여 탈출하는 기법에 대한 연구도 중요하다고 사료된다.

참 고 문 헌

[1] A. Rezgui, N. Tepedelenlioglu, "The Effect of the Slope of the Activation Function on the Backpropagation Algorithm", Proc. IJCNN, Vol. I, pp. 707-710, Jan. 1990.

[2] R. A. Jacobs, "Increased Rates of Convergence through Learning Rate Adaptation", Neural Networks, Vol. I, pp. 295-308, 1988.

[3] U. Bhattacharya, S. K. Parui, "Self-adaptive Learning Rates in Backpropagation Algorithm Improve Its Function Approximation Performance", Proc. ICNN, Vol. 5, pp. 2784-2788, Nov. 1995.

[4] D. B. Parker, "Optimal Algorithms for Adaptive Networks: Second Order Back Propagation, Second Order Direct Propagation, and Second Order Hebbian Learning", Proc. ICNN, Vol. II, pp. 593-600, June 1987.

[5] M. Hagiwara, "Accelerated Back Propagation using Unlearning based on Hebb Rule", Proc. IJCNN, Vol. I, pp. 617-620, Jan. 1990.

[6] 강태원, 황종선, "학습속도 향상을 위한 수정 오류

역전파 알고리즘", 정보과학회 논문지(B), 제23권 7호 pp. 721-733, 1996년 7월.

[7] B. Nassersharif, P. Sadhu, "Enhanced Eigenvalue Acceleration Technique(EEAT) for Backpropagation Learning", Proc. WCNN, Vol. 1, pp. 216-222, Sep. 1996.

[8] C. H. Kwen, C. H. Park, H. S. Lee, "Efficient Strategies for Error Updating to Improve Performance Backpropagation Learning", Proc. ICNN, Vol. 5, pp. 2825-2830, Nov. 1995.

[9] W. E. Deredy, N. M. Branston, "An Update Function that Speeds up Backpropagation Learning", Proc. ICNN, Vol. I, pp. 477-482, Jun. 1994.

[10] D. E. Rumelhart, J. L. McClelland, Parallel Distributed Processing, Vol. 1, pp. 318-330, MIT Press, Cambridge, MA, 1986.

[11] S. B. Thrun, T. Mitchell, J. Cheng, "The MONK's Problems: A Performance Comparison of Different Learning Algorithms", CMU-CS-91-197, 1991.



심 범 식

1985년 명지대학교 전자계산학과 학사
 1987년 명지대학교 대학원 전자계산학과 석사
 1994년 명지대학교 대학원 컴퓨터공학과 박사과정 수료
 1991년~현재 마산전문대 사무

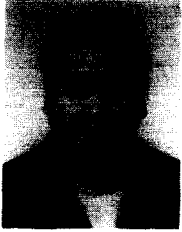
자동화과 조교수
 관심분야: 신경망, 인공지능



정 의 용

1991년 명지대학교 전자계산학과 학사
 1993년 명지대학교 대학원 전자계산학과 석사
 1996년~현재 명지대학교 대학원 컴퓨터공학과 박사과정 재학중

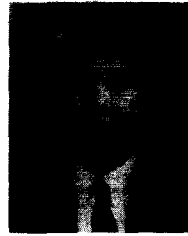
관심분야: 신경망, 멀티미디어



윤 충 화

1979년 서울대학교 수학과 학사
1984년 미국 텍사스대 전자계산
학과 석사
1989년 미국 루이지아나 주립대
전자계산학과 박사
1990년~현재 명지대학교 컴퓨
터공학과 부교수

관심분야: 신경회로망, 전문가시스템, 지능형 소프트
웨어 에이전트, 프로그래밍언어



강 경 식

1974년 인하대학교 산업공학과
학사
1976년 연세대학교 공업경영학
과 석사
1977년 인하대학교 산업공학과
석사
1987년 경희대학교 산업공학과

박사
1976년~현재 명지대학교 산업공학과 교수
관심분야: 전문가시스템, 생산관리