

가변 감쇠 파라미터를 이용한 Levenberg-Marquardt 알고리즘의 학습 속도 향상

곽 영 태*

Accelerating Levenberg-Marquardt Algorithm using Variable Damping Parameter

Young-tae Kwak *

요 약

Levenberg-Marquardt 알고리즘에서 감쇠 파라미터는 오류역전과 학습과 Gauss-Newton 학습의 스위치 역할을 하며 학습 속도에 영향을 준다. 이런 감쇠 파라미터를 고정시키는 것은 오차 함수의 진동을 유발하고 학습 속도를 감소시킨다. 따라서 본 논문은 오차 함수의 변화 과정을 참조하여 감쇠 파라미터를 가변적으로 적용하는 방법을 제안한다. 제안된 방법은 오차의 변화량이 크면 감쇠 파라미터를 크게, 오차의 변화량이 작으면 감쇠 파라미터를 작게 조정한다. 이것은 모멘텀과 유사한 역할을 하여 학습 속도를 향상시킨다. 제안된 방법의 검증은 위한 실험으로는 iris 분류 문제와 wine 분류 문제를 사용하였다. 제안된 방법은 iris 분류 문제에서는 67% 학습에서, wine 분류 문제에서는 78% 학습에서 학습 속도가 향상되었으며 기존 방법과 비교하여 오차의 진동도 적은 것을 확인할 수 있었다.

Abstract

The damping parameter of Levenberg-Marquardt algorithm switches between error backpropagation and Gauss-Newton learning and affects learning speed. Fixing the damping parameter induces some oscillation of error and decreases learning speed. Therefore, we propose the way of a variable damping parameter with referring to the alternation of error. The proposed method makes the damping parameter increase if error rate is large and makes it decrease if error rate is small. This method so plays the role of momentum that it can improve learning speed. We tested both iris recognition and wine recognition for this paper. We found out that this method improved learning speed in 67% cases on iris recognition and in 78% cases on wine recognition. It was also showed that the oscillation of error by the proposed way was less than those of other algorithms.

▶ Keyword : 오류역전파학습, Levenberg-Marquardt 학습, Damping Parameter, Variable Decay Rate

• 제1저자 : 곽영태
• 투고일 : 2010. 03. 10, 심사일 : 2010. 03. 15, 게재확정일 : 2010. 04. 01.
* 전북대학교 IT정보공학부 부교수

I. 서론

함수 근사화, 패턴 인식, 최적화 등에 응용되고 있는 다층 퍼셉트론은 입력층, 은닉층, 출력층으로 구성된 전방향 신경 회로망이다. 이런 다층퍼셉트론의 학습을 위해 여러 가지 학습 방법이 연구되었고 그중 가장 간단한 방법이 오차 함수의 기울기를 이용한 즉, 오차 함수의 1차 미분을 이용한 오류역전파(Error Backpropagation) 학습이다[1,2]. 오류역전파 학습은 기울기 벡터를 사용하므로 학습 속도가 느리고 지역 최소점에 빠질 수 있는 단점이 있다.

오류역전파 학습의 단점을 극복하기 위해서 활성화 함수를 RBF(Radial Basis Function)을 사용하는 방법[3], 오차 함수의 2차 미분을 이용하는 방법, 기타 경험적인 방법 [4,5,6]들이 사용되어 왔다. 그러나 경험적인 방법은 추가적인 파라미터들을 요구하고 문제의 복잡도나 파라미터의 선택에 따라 성능이 좌우되는 경우가 발생한다.

오차 함수의 2차 미분을 이용한 방법으로는 Conjugate Gradient 알고리즘[7,8], Newton 방법과 유사 Newton 방법인 LM(Levenberg-Marquardt) 방법[9,10], Quasi-Newton 방법[10,11]등이 있다. 이런 방법 모두 2차 도함수인 Hessian 행렬을 직접 구하는 대신 근사적인 방법으로 2차 도함수를 구하여 학습에 적용함으로써 빠른 학습 속도를 보인다.

LM 학습은 Hessian 행렬을 직접 구하는 대신 Jacobian 행렬을 이용하여 근사적으로 2차 도함수를 구하고 있으며 다층 퍼셉트론의 크기가 아주 크지 않은 경우 다른 학습 방법에 비해 빠르다고 평가되고 있다[9]. LM 학습의 특징은 오류역전파 학습(학습 속도 느림)의 수렴 안정성과 Newton 방법(수렴 불안정성)의 빠른 학습 속도이다. 여기서, 오류역전파 학습과 Gauss-Newton 방법의 스위치 역할을 하며 LM 학습의 속도를 결정하는 것이 감쇠 파라미터(damping parameter: μ)이다.

감쇠 파라미터를 이용하여 LM 학습 속도를 개선하는 기존 방법에서 Chen[12]은 오차 함수의 로그값에 대한 변화로 가변 감쇠 파라미터를 제안하였으나 초기 학습 속도가 느린 단점이 있다. 또한, Amir[13]은 오차의 제곱을 감쇠 파라미터에 적용하였으나 학습시 오차 함수가 진동하는 경우가 자주 발생한다. 그리고 광영태[14]는 Jacobian 행렬의 주부분 행렬(principal submatrix)을 생성하고 주부분 행렬의 고유값을 이용하여 학습 속도를 개선하였다. 여기서는 주부분 행렬 계산을 위한 추가적인 저장 공간이 필요하다.

따라서 본 논문은 LM 학습에서 Hagan[9]이 제안한 것처럼 감쇠 파라미터를 고정하는 것이 아니라 오차 값에 따라 변

하는 가변 감쇠 파라미터 방법을 제안한다. 제안된 방법은 오차의 변화 과정을 조사하여 감쇠 파라미터를 조정하는 방법으로 오차의 변화량이 크면 감쇠 파라미터를 크게, 오차의 변화량이 작으면 감쇠 파라미터를 작게 변화시키는 모멘텀과 유사한 역할을 한다. 제안된 방법은 패턴 분류에서 많이 사용되는 iris 데이터와 wine 데이터[15]를 대상으로 한 실험에서 기존 방법보다 학습 속도가 상당히 개선되었으며 오차의 수렴 부분에서 오차 함수의 진동이 적게 나타났다.

논문의 구성은 2장에서 LM 알고리즘과 기존 감쇠 파라미터 방법을 설명하고 제안된 LM 학습 알고리즘을 기술한다. 3장에서는 iris 데이터와 wine 데이터의 분류 문제를 통하여 제안한 방법의 적합성을 확인하고 4장에서 결론을 맺는다.

II. 본론

2장에서는 LM 학습과 기존 감쇠 파라미터를 이용한 LM 학습을 기술한다.

2.1 Levenberg-Marquardt 알고리즘

LM 학습은 오차 함수를 테일러 급수로 전개하고 2차 미분을 통하여 근사 최소점을 찾는 방법인데, 2차 미분을 구하기 위하여 Hessian 행렬을 계산해야 한다. 그러나 Hessian 행렬을 구하기 위하여 많은 저장 공간과 계산 시간이 필요하다. 따라서 LM 학습은 Hessian 행렬의 근사화를 위해 Jacobian 행렬을 이용한다.

LM 학습에서 오차 함수는 목표 패턴과 다층퍼셉트론 출력 사이의 오차에 대한 제곱 합으로 수식(2.1)처럼 정의된다. 여기서, $\mathbf{w} = [w_1, w_2, \dots, w_n]^T$ 는 다층퍼셉트론의 가중치(바이어스 포함)이고 \mathbf{e} 는 학습 패턴에 대한 오차 벡터이다.

$$P(\mathbf{w}) = \mathbf{e}^T \mathbf{e} \dots\dots\dots (2.1)$$

LM 학습의 가중치 조정은 수식(2.2)와 같고 \mathbf{I} 는 단위 행렬이며 \mathbf{J} 는 Jacobian 행렬이다.

$$\Delta \mathbf{w} = - [\mathbf{J}^T \mathbf{J} + \mu \mathbf{I}]^{-1} \mathbf{J}^T \mathbf{e} \dots\dots\dots (2.2)$$

수식(2.2)에서 감쇠 파라미터 μ 는 $\mu > 0$ 이고, $P(\mathbf{w})$ 가 증가하면 μ 는 수식(2.3)과 같이 $\beta (0 < \beta < 1)$ 로 나누고, $P(\mathbf{w})$ 가 감소하면 β 를 곱한다. 따라서 큰 μ 값은 오류역전파 학습을 수행하며, 작은 μ 값은 Gauss-Newton 방법을 수행한다.

$$\mu = \begin{cases} \mu/\beta & \text{if } P(\mathbf{w}_{i+1}) \geq P(\mathbf{w}_i) \\ \mu \cdot \beta & \text{if } P(\mathbf{w}_{i+1}) < P(\mathbf{w}_i) \end{cases} \dots\dots\dots (23)$$

일반적인 LM 학습 과정을 요약하면 다음과 같으며 Hagan은 μ 값을 초기에는 작은 값(0.01)으로 설정한 후 학습이 진행됨에 따라 $\beta(=0.1)$ 을 오차 함수의 변화에 따라 곱하거나, 나누는 과정을 반복한다. 이런 고정된 β 값은 학습 초기와 종료 시점에 오차의 진동을 유발한다.

```

1. Initialize the weights and parameter  $\mu(\mu = 0.01)$ 
2. Compute the sum of squared errors over all inputs  $P(\mathbf{w})$ 
3. Compute the Jacobian matrix  $\mathbf{J}$ 
4. Solve equation(2.2) to obtain the increment of weights  $\Delta\mathbf{w}$ 
5. Recompute the sum of squared errors  $P(\mathbf{w})$  using  $\mathbf{w} + \Delta\mathbf{w}$  as the trial  $P(\mathbf{w})$ , and judge
   IF (trial  $P(\mathbf{w})$ ) < ( $P(\mathbf{w})$  in step 2) THEN
        $\mathbf{w} = \mathbf{w} + \Delta\mathbf{w}$ 
        $\mu = \mu \cdot \beta$  ( $\beta = 0.1$ )
       go back to step 2
   ELSE
        $\mu = \mu/\beta$ 
       go back to step 4
    
```

따라서 본 논문에서는 오차의 변화 과정을 조사하여 학습 상태에 따라 가변적으로 변화하는 가변 감쇠 파라미터 방법을 제시한다. 우선 학습 속도를 향상시키기 위해 제안된 기존의 감쇠 파라미터 방법을 알아보자.

2.2. 기존 감쇠 파라미터 방법

Chen은 Hagan의 알고리즘에서 처럼 β 값이 고정되어 있는 경우 학습 말기에 즉, 오차 함수가 목표로 하고 있는 오차 값에 근접하는 경우 오차가 진동하는 것을 방지하기 위해 수식(2.4)와 같은 가변 감쇠 파라미터를 제안하였다. 여기서, P 는 오차 함수, P_0 는 초기 오차 함수, P_{\min} 은 목표 오차 값이다.

$$\beta = \frac{\log(P) - \log(P_{\min})}{\log(P_0) - \log(P_{\min})} \times 0.8 + 0.1 \dots\dots\dots (24)$$

Chen의 단점으로는 초기 학습 속도가 느리다. LM 학습은 학습 초기에 오류역전파 학습처럼 학습한다. 즉 μ 값이 증가하므로 β 값은 더 작아야 한다. 또한, 학습 말기에는 Gauss-Newton 방법처럼 학습하기 위해 μ 값은 감소해야 하고 β 값도 작아야 한다. 그러나 학습 초기 $P = P_0$ 인 경우 $\beta = 0.9$ 로써 Hagan의 $\beta = 0.1$ 보다 크기 때문에 초기 학습 속도가 느리고 이런 결과는 실험에서도 확인할 수 있다.

Amir는 μ 값이 학습의 진행 상태를 나타낸다는 점을 고려하여 수식(2.5)와 같은 방법을 고안했다. 학습이 많이 진행되지 않는 초기에는 $\mathbf{e}^T\mathbf{e}$ 값이 크므로 μ 값을 크게 한다. 그리고 학습이 거의 완료되는 말기에는 $\mathbf{e}^T\mathbf{e}$ 값이 작아지므로 μ 값을 작게하여 오차 함수를 최소화시킨다. 그러나 Amir방법은 적용하고자 하는 대상에 따라 $\mathbf{e}^T\mathbf{e}$ 값의 변화가 크고 이것은 불규칙적인 μ 값을 유도한다. 또한 μ 값의 최대/최소 경계값을 설정하기 어려운 문제점을 가지고 있다.

$$\mu = \mu \mathbf{e}^T \mathbf{e} \dots\dots\dots (25)$$

2.3. 가변 감쇠 파라미터를 이용한 LM 알고리즘

우리가 학습하고자 하는 문제는 대부분이 선형 분리가 되지 않는 비선형적인 문제이다. 이런 비선형적인 문제의 오차 함수는 정확히 2차 함수(quadratic function)와 일치하지 않는다. 그러므로 Newton 유사 방법(Gauss-Newton 방법)을 직접 사용할 수 없어 1차 도함수를 이용한 최급 강하법(steepest gradient descent)과 혼성시킨 것이 LM 학습이다. LM 학습은 오류역전파 학습과 Gauss-Newton 방법의 특징을 모두 가지고 있다. 학습 초기시 LM 학습은 오류역전파 학습의 특징을 나타내며 학습이 진행되어 오차가 감소할수록 Gauss-Newton 방법의 특징을 나타낸다. 이때 μ 값은 학습 초기에 증가하다가 학습이 진행됨에 따라 감소한다.

일반적으로 μ 는 임의의 작은 값(0.01)으로 초기화 한다. 그러나 LM 학습이 오류역전파 학습과 유사해질 때까지 μ 값은 증가하며 이런 현상은 실험에서도 확인할 수 있다. 이런 μ 값의 증가를 Hagan은 고정값 $\beta(=0.1)$ 로 나누어 10배씩 증가시키지만 본 논문에서는 다음 그림 1과 같이 μ 값을 조정한다.

그림 1은 학습시 epoch의 변화에 따라 오차 함수 P 가 변화하는 것을 나타내며 $P(i)$ 는 i epoch의 현재 오차이고 $trial_P(i)$ 는 수식(2.2)를 사용하여 기중치를 조정할 경우의 새로운 오차이다. $trial_P(i)$ 가 현재 오차 $P(i)$ 보다 커서 μ 를 가변적으로 증가시키는 방법은 수식(2.6)과 같다.

$$\beta = \beta \left| \frac{trial_P(i) - P(i)}{trial_P(i-1) - P(i-1)} \right| \dots\dots\dots (26)$$

$$\mu = \mu/\beta$$

$trial_P(i)$ 가 계속 증가하는 경우 $trial_P(i-1)$ 의 변화를 관찰하여 현재 μ 값의 변경에 반영한다. 그림 1에서 ①의 경우 $trial_P(i)$ 가 $trial_P(i-1)$ 보다 증가하면 수식(2.6)은 Hagan의 고정된 $\beta(=0.1)$ 보다 β 값을 많이 증가시킨다.

그러나 ②의 경우 현재의 $trial_P(i)$ 가 $trial_P(i-1)$ 보다 변화가 작으므로 고정된 β 보다 β 값을 작게 증가시킨다. 이런 μ 의 변화는 과거의 $trial_P(i-1)$ 의 변화를 참조하므로 모멘트 역할을 한다.

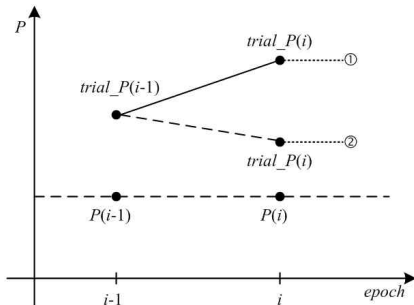


그림 1. μ 가 증가하는 경우
Fig 1. Case of increasing μ

반대로 μ 가 감소하는 경우 다시 말해, $trial_P(i)$ 가 $P(i)$ 보다 작아 가중치를 조정하고 μ 를 감소시키는 경우는 수식(2.7)이며 이에 대한 그림 2가 있다.

$$\beta = \beta \left| \frac{P(i) - P(i-1)}{trial_P(i) - P(i)} \right| \dots\dots\dots (2.7)$$

$$\mu = \mu\beta$$

이 경우는 μ 가 증가하는 경우와 다르게 $trial_P(i-1)$ 의 값 대신 $P(i)$ 을 사용한다. 이것은 오차값의 관계식이 $P(i) = trial_P(i-1)$ 이기 때문이다. 수식(2.7) 또한 모멘트 역할을 하는데, 현재의 오차 변화량 $|trial_P(i) - P(i)|$ 가 과거 오차값의 변화량 $|P(i) - P(i-1)|$ 보다 크면 β 를 0.1보다 더 작게 만들고 만약에 그 반대이면 β 를 $0.1 < \beta < 1$ 사이의 값으로 만들어 μ 값을 감소시킨다.

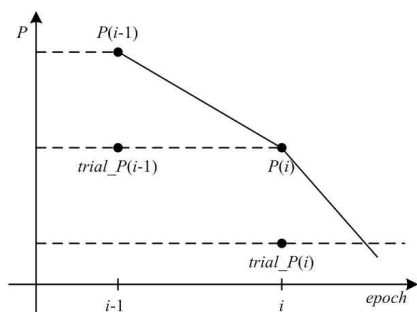


그림 2. μ 가 감소하는 경우
Fig 2. Case of decreasing μ

수식(2.6)과 수식(2.7)은 모두 과거의 $trial_P(i-1)$ 나

$P(i-1)$ 를 참조하고 있다. 그러므로 $i = 1$ 인 초기 설정 문제와 μ 가 증가하다가 감소하는 경우와 반대로 감소하다가 증가하는 문제를 해결해야 한다. 초기 설정 문제는 $i-1$ epoch를 계산할 수 없으며 μ 의 증가 및 감소에 대한 수식(2.6)과 수식(2.7)이 서로 상이하므로 논문은 위 문제의 해결 방법으로 수식(2.3)의 Hagan의 방법을 그대로 사용한다. 또한 LM 학습 동안 μ 가 계속 증가 상황인지 또는 감소 상황인지를 구분하는 up_flag 와 $down_flag$ 를 사용한다. $up_flag = 1$ 이면 $trial_P(i)$ 가 계속 증가하는 상황이며 $down_flag = 1$ 이면 $trial_P(i)$ 가 계속 감소하는 상황이다. 제안한 알고리즘의 전체적인 구조는 다음과 같다.

```

1. Initialize the weights and parameter
    $\mu(0.01), \beta(0.1), up\_flag = 0, down\_flag = 0$ 
2. Compute the sum of squared errors over all inputs
    $P(\mathbf{w})$ 
3. Compute the Jacobian matrix  $\mathbf{J}$ 
4. Solve equation(2.2) to obtain the increment of
   weights  $\Delta\mathbf{w}$ 
5. Recompute the sum of squared errors  $P(\mathbf{w})$  using  $\mathbf{w} + \Delta\mathbf{w}$ 
   as the trial  $P(\mathbf{w})$ , and judge
   IF  $(trial\ P(\mathbf{w})) < (P(\mathbf{w})\ \text{in\ step\ 2})$  THEN
      $up\_flag = 0$ 
      $\mathbf{w} = \mathbf{w} + \Delta\mathbf{w}$ 
     IF  $down\_flag = 0$ 
        $\mu = \mu \cdot \beta$ 
        $down\_flag = 1$ 
     ELSE
        $\beta = \beta \left| \frac{P(i) - P(i-1)}{trial\_P(i) - P(i)} \right|$ 
        $\mu = \mu\beta$ 
     END
     go back to step 2
   ELSE
      $down\_flag = 0$ 
     IF  $up\_flag = 0$ 
        $\mu = \mu/\beta$ 
        $up\_flag = 1$ 
     ELSE
        $\beta = \beta \left| \frac{trial\_P(i) - P(i)}{trial\_P(i-1) - P(i-1)} \right|$ 
        $\mu = \mu/\beta$ 
     END
     go back to step 4

```

우선 학습전에 가중치와 파라미터들을 초기화한다. 그리고 2-4번 과정은 기존 LM 학습을 그대로 이용한다. 오차가 감소하는 $trial_P(i) < P(i)$ 인 경우는 $up_flag = 0$ 으로 초기화하여 오차가 증가하는 상황이 종료되고 오차가 감소하는

상황임을 설정한다. 그리고 오차가 처음 감소하면 μ 값을 Hagan의 방법으로 변경한 후 $down_flag = 1$ 로 설정하여 다음의 오차 감소가 연속적인 오차 감소임을 나타내고 수식(2.7)과 같이 μ 값을 변경한다.

이와 유사하게 $trial_P(i) \geq P(i)$ 으로 오차가 증가하는 경우는 $down_flag = 0$ 으로 오차가 감소하는 상황이 아님을 나타내고 처음 오차가 증가하면 Hagan의 방법으로, 연속적으로 오차가 증가하면 수식(2.6)과 같이 μ 값을 변경한다.

III. 실험결과

가변 감쇠 파라미터를 이용한 LM 학습 속도 향상을 위하여 본 논문에서는 두 가지 실험을 하였다. 패턴 인식 문제에서 널리 사용되고 있는 iris 문제와 wine 분류 문제를 대상으로 실험하였다. 여기서 사용한 오차함수는 수식(2.8)과 같은 Mean Squared Error함수이며 N 은 총 학습 패턴의 수이다. LM 학습의 구현은 Matlab을 이용하여 구현하였다.

$$P = \frac{1}{N} \mathbf{e}^T \mathbf{e} \dots\dots\dots (2.8)$$

Iris 문제는 꽃받침의 길이와 넓이, 꽃잎의 길이와 넓이, 모두 4개를 입력하여 3종류의 iris로 분류하는 문제이다. 다층퍼셉트론의 구성은 입력 노드 4개, 출력 노드 3개, 은닉 노드는 5, 8, 10개로 변경하면서 3번씩 학습하였다. 학습에 사용된 패턴은 각 iris 종류별로 50개씩 총 150개가 사용되었고 최대 epoch가 100이 되거나 MSE가 0.01이하가 될 때까지 학습하였다. 또한, 기존 방법들과 비교하기 위하여 초기 가중치 설정을 동일하게 적용하였다.

표 1. Iris 분류 결과
Table 1. Result of Iris Classification

비교방법 \ 노드수	5			8			10		
제안 방법	39	63	50	45	51	29	36	36	47
Hagan 방법	50	52	69	52	40	56	43	37	69
Chen 방법	x	x	97	92	59	94	x	84	84

표 1은 iris 분류 문제의 학습결과를 나타낸다. 표 1의 단위는 epoch이고 x는 최대 epoch까지 수렴하지 못한 것을 나타낸다. 실험된 3가지의 방법중 Chen의 방법이 가장 학습 속도가 느리다. 이것은 학습 초기 β 값이 작아져야 하지만,

$\beta = 0.9$ 를 유지함으로써 학습 속도를 느리게 만들고 있다. 전체 실험중 2/3의 실험에서 제안된 방법이 Hagan의 방법보다 학습 속도의 향상을 가져왔다. 또한, 은닉 노드의 수(10)가 많은 경우보다 적은 경우에 학습 속도의 향상이 더 뚜렷했다. 이것은 제안한 방법이 기존 Hagan의 방법을 포함하고 오차의 변화량에 따라 감쇠 파라미터를 확대/축소하는 작용을 하고 있기 때문이다.

그림 3은 은닉 노드 8개의 첫 번째 실험에서 나타나는 각 방법의 오차 변화를 나타낸다. 모든 그래프에서 x축은 선형으로 epoch를 나타내고 y축은 대수적으로 오차 P 를 나타내어 가독성을 높였다. 여기서 Our는 논문에서 제안한 방법이다. Chen의 방법은 감쇠 파라미터를 계산하기 위해 오차에 log를 취하고 있기 때문에 다른 방법에 비해 오차의 진동이 덜하다. 그러나 앞서 지적한 단점처럼 초기 학습 속도가 느리다. 제안 방법과 Hagan의 방법은 오차의 증감에 따라 감쇠 파라미터를 선형적으로 변형시키기 때문에 오차의 진동을 피할 수 없다. 그러나 제안 방법은 오차의 변화에 따라 가변적으로 감쇠 파라미터를 변화시키기 때문에 Hagan의 방법보다 오차의 진동 폭이 작고 학습 속도를 향상시킬 수 있었다.

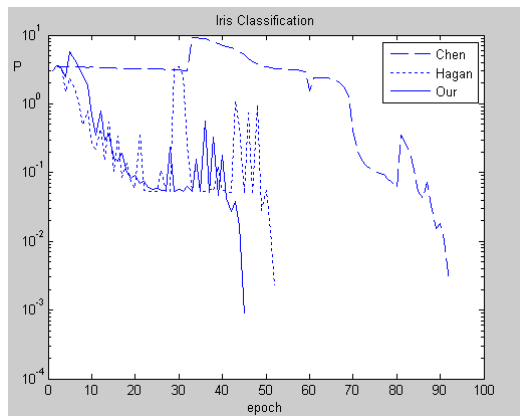


그림 3. Iris 분류의 학습 결과
Fig 3. Result of Iris Classification

두 번째 실험은 wine 분류 문제로서 13개의 화학 성분을 입력받아 3종류의 wine으로 분류하는 문제이다. 따라서 다층 퍼셉트론의 구성은 입력 노드 13개, 출력 노드 3개, 은닉 노드 4, 6, 9개로 변경하면서 3번씩 학습하였다. 학습에 사용된 패턴은 178개이며 학습 조건은 iris 문제와 동일하게 하였다. 표 2는 wine 분류 문제의 학습 결과이며 그림 4는 은닉 노드 4개의 두 번째 실험에서 오차의 변화를 나타낸다.

표 2. Wine 분류 결과
Table 2. Result of Wine Classification

비교방법 \ 노드수	4			6			9		
	제안 방법	Hagan 방법	Chen 방법	제안 방법	Hagan 방법	Chen 방법	제안 방법	Hagan 방법	Chen 방법
제안 방법	13	13	13	21	13	10	14	20	15
Hagan 방법	14	17	13	16	14	14	16	13	19
Chen 방법	60	53	48	61	60	58	57	59	59

Wine 분류 문제는 iris 분류 문제보다 입력 성분이 많아 복잡성이 증가할 것으로 기대되었으나 iris 문제보다 단순한 문제였다. 따라서 전체적으로 학습 시간이 iris 문제보다 짧았고 Chen 방법도 최대 epoch내에서 학습할 수 있었다. 제안한 방법은 Chen의 방법보다 우수할 뿐만 아니라 Hagan 방법과 비교하여 두 번의 실험을 제외하고 모두 우수하였다.

그림 4에서처럼 Chen의 방법은 오차의 진동은 작지만 학습 속도가 느리고 제안한 방법과 Hagan의 방법은 오차의 진동은 있지만 학습 속도가 빠르다. 그리고 제안한 방법이 Hagan의 방법보다 오차의 진동폭이 작음을 알 수 있다.

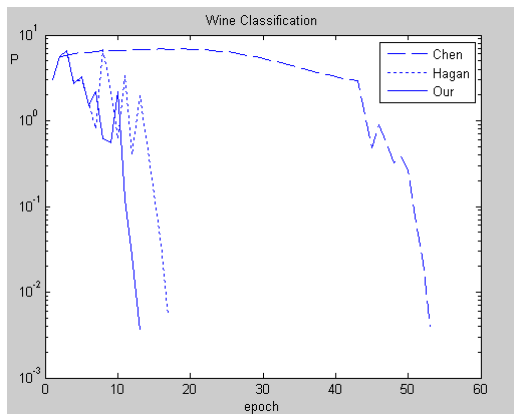


그림 4. Wine 분류의 학습 결과
Fig 4. Result of Wine Classification

Hagan의 방법이 두 번의 wine 분류 실험과 1/3의 iris 실험에서 우수하였는데 이것은 초기 가중치 설정이 학습 속도에 영향을 준다는 것을 의미한다. 그러나 임의의 초기 가중치 설정에서 제안한 방법은 wine 분류 문제에 대해 78% 학습에서 학습 속도 향상과 iris 문제에 대해 67% 학습에서 학습 속도 향상을 가져왔다.

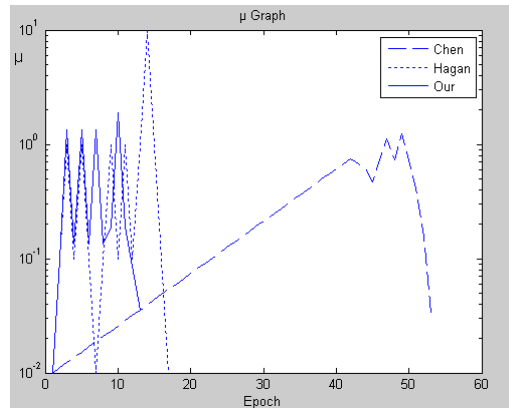


그림 5. μ 그래프
Fig 5. μ Graph

그림 5는 그림4의 실험에서 발생하는 μ 값의 변화를 나타낸 그래프이다. 그림 5에서처럼 모든 μ 는 작은 값으로 초기화되지만 학습이 진행됨에 따라 증가한다. 이때 LM 학습은 오류역전과 학습처럼 작용하고 학습 말기에 μ 는 감소하여 Gauss-Newton 방법처럼 오차를 수렴한다. 제안한 방법과 Hagan의 방법에서 μ 값의 진동을 확인할 수 있지만 Hagan의 μ 가 진동폭이 크다. 이에 반해 제안한 방법은 μ 의 진동폭이 작다. 또한 Chen의 μ 값의 진동은 없지만 μ 값의 증가에 시간이 소요되므로 학습 속도가 느려지는 결과를 얻었다.

IV. 결론

LM 학습은 오류역전과 학습과 Gauss-Newton 학습의 특징을 모두 가지고 있으며 이들 사이에서 스위치 역할을 하는 것이 감쇠 파라미터이다. LM 학습을 다층퍼셉트론에 적용한 Hagan은 감쇠 파라미터를 계산하는 β 값을 고정하였고 이것은 오차 함수의 진동을 유발하고 학습 속도를 지연시킨다.

따라서 본 논문은 Hagan의 고정 감쇠 파라미터를 개선하는 새로운 알고리즘을 제시하였다. 제안된 방법은 오차의 진동을 적게 하고 오차의 변화 과정을 조사하여 감쇠 파라미터를 가변적으로 조정하는 방법이다. 이런 방법은 오차의 변화량이 크면 감쇠 파라미터를 크게, 오차의 변화량이 작으면 작게 변화시키는 모멘텀과 유사한 역할을 하기 때문에 학습 속도를 향상할 수 있다.

Iris 문제와 wine 분류 문제의 실험에서 제안된 방법은 가장 우수한 학습 속도의 향상을 보였으며, 오차 함수의 진동에서는 Chen의 방법보다는 많지만 Hagan의 방법보다는 적어 오차의 진동폭을 줄일 수 있었다.

참고문헌

- [1] D. E. Rumelhart and J. L. McClelland, "Parallel Distributed Processing," MIT Press, Cambridge, MA, pp. 318-362, 1986.
- [2] 김광백, 조재현, "퍼지 신경망을 이용한 자동차 번호판 인식 시스템," 한국컴퓨터정보학회 논문지, 제 12권 제 5호, 313-319쪽, 2007년 11월.
- [3] John Moody and Christian J. Darken, "Fast Learning in Networks of Locally-Tuned Processing Units," Neural Computation, Vol. 1, pp. 281-294, 1989.
- [4] T. P. Vogal, J. K. Mangis, A. K. Zigler, W. T. Zink and D. L. Alkon, "Accelerating the convergence of the backpropagation method," Biological Cybernetics, Vol. 59, pp. 256-264, Sept. 1988.
- [5] T. Tollaere, "SuperSAB: Fast adaptive back propagation with good scaling properties," Neural Networks, Vol. 3, No. 5, pp. 561-573, 1990.
- [6] M. Kordos and W. Duch, "Variable step search algorithm for feedforward networks," Neurocomputing, Vol. 71, pp. 2470-2480, 2008.
- [7] C. Charalambous, "Conjugate gradient algorithm for efficient training of artificial neural networks," IEEE Proceedings, Vol. 139, No. 3, pp. 301-310, 1992.
- [8] Wei Chu, Chong Jin Ong and Keerthi S.S., "An improved conjugate gradient scheme to the solution of least squares SVM," IEEE Transactions on Neural Networks, Vol. 16, No. 2, pp. 498-501, 2005.
- [9] M. T. Hagan and M. Menhaj, "Training feedforward networks with the Marquardt algorithm," IEEE Transactions on Neural Networks, Vol. 5, No. 6, pp. 989-993, 1994.
- [10] Jorge Nocedal and Stephen Wright, *Numerical Optimization*, Springer, 2006.
- [11] Ronald E. Miller, "Optimization," John Wiley & Son, INC. pp. 358-362, 2000.
- [12] Chen T., Han DJ., Au FTK. and Tham LG., "Acceleration of Levenberg-Marquardt training of neural networks with variable decay rate," International Joint Conference on Neural Networks Proceedings, Portland, Oregon, Vol. 3, pp. 1873-1878, July 2003.
- [13] Amir Abolfazl Suratgar, Mohammad Bagher Tavakoli and Abbas Hseinabad, "Modified Levenberg-Marquardt Method for Neural Networks Training," Proceedings of world academy of science, engineering and technology, Vol. 6, pp. 46-48, June 2005.
- [14] 박영태, 신정훈, "Jacobian 행렬의 주부분 행렬을 이용한 Levenberg-Marquardt 알고리즘의 개선," 한국컴퓨터정보학회 논문지, 제 14권, 제 8호, 11-18쪽, 2009년 8월.
- [15] Machine Learning Repository
<http://archive.ics.uci.edu/ml/>

저자 소개



박영태

1993년 : 충남대학교 컴퓨터공학과
공학사

1995년 : 충남대학교컴퓨터공학과
공학석사

2001년 : 충남대학교컴퓨터공학과
공학박사

2001년 - 2002년 :

한국전자통신연구원선임연구원

2002년 3월 - 현재 :

전북대학교 IT정보공학부 부교수

관심분야 : 신경회로망, 패턴인식, 영
상처리