

# Kernel Relaxation과 동적 모멘트를 조합한 Support Vector Machine의 학습 성능 향상

김 은 미<sup>†</sup> · 이 배 호<sup>††</sup>

## 요 약

본 논문에서는 커널완화법과 동적모멘트를 이용한 support vector machines의 학습성능 개선을 제안하였다. 제안된 학습 방법은 기존의 정적모멘트와는 달리 수렴 정도에 따라 현재의 학습에 과거의 학습 속성을 반영하는 동적모멘트 방법이다. 기존의 정적 상수로 정의된 모멘트가 전체 학습에 동등하게 영향을 주는 반면 제안된 동적모멘트를 이용한 학습방법은 학습 수행에 따라 동적으로 모멘트를 변경함으로써 수렴속도와 학습 성능을 효과적으로 제어할 수 있다. 제안된 학습법을 support vector machine의 새로운 순차 학습 방법인 커널완화법에 적용하였다. 신경망 분류기 표준 평가 데이터인 SONAR 데이터를 이용하여 실험한 결과 동적모멘트를 이용한 방법이 수렴속도와 학습 성능면에서 기존의 커널완화법과 정적모멘트를 이용한 학습법에 비해 향상된 성능을 보이는 것을 확인하였다.

## Improving Learning Performance of Support Vector Machine using the Kernel Relaxation and the Dynamic Momentum

Eun-Mi Kim<sup>†</sup> · Bae-Ho Lee<sup>††</sup>

## ABSTRACT

This paper proposes learning performance improvement of support vector machine using the kernel relaxation and the dynamic momentum. The dynamic momentum is reflected to different momentum according to current state. While static momentum is equally influenced on the whole, the proposed dynamic momentum algorithm can control to the convergence rate and performance according to the change of the dynamic momentum by training. The proposed algorithm has been applied to the kernel relaxation as the new sequential learning method of support vector machine presented recently. The proposed algorithm has been applied to the SONAR data which is used to the standard classification problems for evaluating neural network. The simulation results of proposed algorithm have better the convergence rate and performance than those using kernel relaxation and static momentum, respectively.

**Keyword :** Support Vector Machine, 커널 완화(Kernel Relaxation), 동적 모멘트(Dynamic Momentum)

### 1. 서 론

Support vector machine(SVM)은 Vapnik에 의해 제안된 방법으로 분류 문제에 있어 두 부류사이의 거리를 최대로 하는 초평면(hyperplane)을 찾는다[1]. 이때 SVM의 해는 quadratic programming(QP)를 계산함으로써 구하게 된다. 하지만 QP는 입력 패턴의 양이 커질수록 복잡한 연산과 과도한 메모리 요구로 인해 시간과 공간에 있어 최대의 취약점을 가지며 그 결과 QP의 문제점은 SVM의 일반화에 많은 제약이 된다[2-6]. 이러한 문제점을 해결하기 위해 많은 연구가 있

었으며 문제해결의 접근방법에 따라 재구성(decomposition) 방법과 순차 학습 방법(sequential learning method)으로 나눌 수 있다. 이중 재구성 방법은 QP가 가지는 메모리의 한계를 벗어나기 위한 방법들로 SVM에 대한 직접적인 접근 방법은 아니다.

이에 반해 순차 학습 방법인 kernel adatron(KA), 커널완화법(KR)등은 기존의 SVM을 수정 보완한 학습법[7]으로 보다 직접적인 접근 방법으로 볼 수 있다. KA의 학습 방법은 SVM의 비선형 특징 공간에 기존의 신경망 학습 방법을 접목한 방식으로 기울기 강하/상승(gradient descent/ascent)기법을 이용하여 가중치를 학습함으로써 해를 찾는 방법이다[12]. 그러나 이에대한 체계적인 이론적 전개는 최근 발표된 KR에 의해 완성되었다[4]. KR은 KA와 동등한 학습 성능을 가지며 support vectors(SVs)의 수를 효과적으로 줄이는 저

\* 본 연구는 한국과학재단 지정 전남대학교 고품질전자전자부품 및 시스템 연구센터의 연구비 지원에 의해 연구되었음.

† 준 회 원 : 전남대학교 대학원 컴퓨터공학과

†† 정 회 원 : 전남대학교 전자컴퓨터정보통신공학부, 정보통신연구소 교수  
논문접수 : 2002년 4월 24일, 심사완료 : 2002년 8월 21일

밀도 학습법으로 기존의 선형 판별함수의 학습법을 커널 공간(kernel space)으로 확장한 것으로 볼 수 있다[4]. 이때 선형 판별 함수의 학습법에서 나타나는 수렴시의 발진 현상은 커널 공간에서도 발생하게 된다. 이러한 발진 현상을 줄여 수렴 속도를 향상하는 방법으로는 기존의 선형 판별 함수의 학습법이나 신경망에서 제안된 모멘트를 이용할 수 있다. 모멘트를 이용한 학습법은 과거의 학습을 현재의 학습에 반영하는 것으로 이때 모멘트는 상수이며 사용자에게 의해 정의된다. 커널 공간에서 모멘트를 이용한 방법은 모멘트를 이용한 KA의 학습 성능 향상 기법으로 소개되었다[8].

이에 본 논문에서는 기존의 상수 모멘트를 정적(static)모멘트로 정의하며 본 논문에서 제안한 동적(dynamic)모멘트와 구분한다.

정적 모멘트는 전체 학습에 있어 수렴과 학습정도에 관계없이 동일한 모멘트 값의 영향을 받는 것으로 학습의 횟수나 성능에 관계없이 일정하게 영향력을 가진다. 따라서 학습 성능과 학습 횟수에 따라 다른 크기를 갖는 모멘트가 필요하다.

본 논문에서는 순차 학습법에서의 수렴시 발진억제를 위해 과거의 속성값을 반영하는 모멘트 항을 추가하여 KR의 수렴 속도와 학습 성능을 향상시키는 동적모멘트를 이용한 학습 방법을 제안한다. 기존의 모멘트는 상수로 선언되어 학습에 있어 수렴 정도에 관계없이 전체 학습에 균등하게 영향을 주는 것에 비해 제안된 동적모멘트를 이용한 학습 방법은 수렴 정도에 따라 모멘트의 크기를 조절하는 방법이다. 제안된 학습 방법을 신경망 분류기 표준 평가 데이터인 SONAR 분류 문제를 가지고 실험하였다[9].

## 2. Support Vector Machine

분리 가능한 문제(separable problem)인 경우 입력값  $\mathbf{x}_i$ 과 목표치  $d_i$ 에 대한 선형 판별 함수는 다음 조건을 만족한다[3].

$$\begin{aligned} g(\mathbf{x}_i) &> 0, d_i = 1 \\ g(\mathbf{x}_i) &< 0, d_i = -1 \end{aligned} \quad (2.1)$$

분리 불가능한 문제(nonseparable problem)인 경우 선형 판별 함수는 여유  $b$ 에 오차  $\xi$ 를 도입하여 표현한다.

$$\begin{aligned} d_i g(\mathbf{x}_i) &\geq b - \xi_i, \\ i &= 1, 2, \dots, N \text{ with } \xi_i \geq 0 \end{aligned} \quad (2.2)$$

여기서  $\xi = 0$ 일때 패턴으로부터 분리 초평면까지의 최대여유  $r$ 를 갖는다. 가중치  $\mathbf{w}$ 와 오차  $\xi$ 를 동시에 최소화하는 비용함수  $\phi$ 는 아래와 같다[13].

$$\phi(\mathbf{w}, \xi) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \quad (2.3)$$

여기서  $C$ 는 정규화 인수이다.

라그랑지안(Lagrangian) 승수법을 이용하여 라그랑지안 함수를 구성하면 다음과 같다.

$$\begin{aligned} L(\mathbf{w}, w_0, \xi_i; \alpha, \beta) &= \frac{\|\mathbf{w}\|^2}{2} + C \sum_{i=1}^N \xi_i \\ &- \sum_{i=1}^N \alpha_i [d_i (\mathbf{w}^t \mathbf{x}_i + w_0) - b + \xi_i] \\ &- \sum_{i=1}^N \beta_i \xi_i, \end{aligned} \quad (2.4)$$

$$\text{with } \alpha_i \geq 0, \beta_i \geq 0, \xi_i \geq 0$$

$$\text{여기서, } g(x) = \mathbf{w}^t \mathbf{x} + w_0.$$

이때 최적화 문제의 해는 가중치, 바이어스, 오차 ( $\mathbf{w}, w_0, \xi_i$ )에 관하여 최소, 라그랑지안 승수 ( $\alpha, \beta$ )에 관하여 최대가 되어야 한다.

쌍대 공간(dual space)에서의 라그랑지안  $L(\alpha)$ 는 아래와 같다[3, 7].

$$\begin{aligned} L(\alpha) &= b \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j \mathbf{x}_i^t \mathbf{x}_j, \\ \text{subject to} & \\ \sum_{i=1}^N \alpha_i d_i &= 0, \\ 0 &\leq \alpha_i \leq C. \end{aligned} \quad (2.5)$$

바이어스  $w_0$ 는 가중치  $\mathbf{w}$ 와 독립적으로 아래와 같다[4].

$$w_0 = \frac{\sum_{i=1}^{N_s} (d_i - \mathbf{w}^t \mathbf{x}_i)}{N_s} \quad (2.6)$$

여기서  $N_s$ 는 SV의 개수이다. SVM의 학습 방법은 라그랑지안 계수  $\alpha$ 의 해를 QP로 구하게 된다. QP는 근본적으로 복잡한 연산과 많은 메모리량을 요구하므로 SVM은 그 자체가 가진 알고리즘이 우수함에도 불구하고 입력 패턴의 수가 커질 경우 그 패턴들의 계산 과정에서 엄청난 시간과 동시에 많은 메모리가 필요하게 되는 최대의 취약점을 갖게 되며  $L(\alpha)$ 에서 두 개의 제약 조건중  $\sum_{i=1}^N \alpha_i d_i = 0$ 으로 인하여 순차적 학습을 할 수 없게 되었다. 이에 가중치를 바이어스만큼 확장시킴으로 기존과 달리 가중치와 바이어스를 동시에 학습시킴으로 순차적 학습을 할 수 있다.

입력 공간에서의 가중치  $\mathbf{w}$ 와 바이어스 항  $w_0$ 으로 확장

한 확장 공간에서 가중치 벡터  $\mathbf{a} = [\mathbf{w}, w_0]^t$ 를 최소화하면  $\mathbf{w}$ 와  $w_0$ 는 동시에 최소화된다. 이때  $\mathbf{y} = [\mathbf{x}, 1]^t$ 로 확장된다[3]. 이와 같은 확장된 쌍대 공간에서  $L(\mathbf{a})$ 는 기존의 SVM의 등식 제한 조건이 제거되었다.

$$L(\mathbf{a}) = b \sum_{i=1}^N \alpha_i d_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j \mathbf{y}_i^t \mathbf{y}_j \quad (2.7)$$

subject to  $0 \leq \alpha_i, d_i \leq C$ .

이로써 커널 공간에서 가중치  $\mathbf{a}$ 를 학습할 수 있는 순차 학습 알고리즘이 가능하게 된다.

### 3. 커널완화법(Kernel Relaxation)

커널은 특징공간의 정확한 형태를 고려하지 않아도 최적의 초평면(optimal hyperplane)을 구성하기 위해 사용한다[7]. 내적 커널의 정의는 다음과 같다.

$$\mathbf{K}(\mathbf{x}, \mathbf{x}_i) = \boldsymbol{\varphi}^T(\mathbf{x}) \boldsymbol{\varphi}(\mathbf{x}_i), \quad (3.1)$$

여기서  $\boldsymbol{\varphi}(\mathbf{x})$ 는 크기가  $(m+1)$ -1로 확장된 특징 벡터이다. 실험에서는 머서의 정리 [7]를 만족하는 아래의 RBF 커널을 사용하였다.

$$\exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{x}_i\|^2\right). \quad (3.2)$$

이때  $\sigma^2$ 은 사용자에게 의해 정의된다. 커널 공간에서 커널 행렬을 패턴 행렬로 커널의 행과 열을 패턴 벡터로 정의하면 라그랑지안 승수 ( $\boldsymbol{\alpha}$ )는 패턴의 가중치가 되어 아래와 같이 커널 공간에서의 식  $\mathbf{g}_k(\mathbf{Y})$ 를 정의한다.

$$\mathbf{g}_k(\mathbf{Y}) \equiv \text{sign}(\mathbf{K}\boldsymbol{\alpha}) = \mathbf{d}. \quad (3.3)$$

여기서 커널 판별 함수는 다음과 같다[4].

$$\mathbf{g}_k(\mathbf{y}_j) \equiv \boldsymbol{\alpha}^t \mathbf{K}(:, j) = \mathbf{K}(j, :)\boldsymbol{\alpha}. \quad (3.4)$$

따라서 입력 공간은 방정식의 수가 미지수의 수 보다 많은 과잉결정 구조(overdetermined system)이고 커널공간은 방정식의 수와 미지수의 수가 같은 완전 결정 구조(exactly determined system)가 된다. 결국 커널 공간에서는 비특이 행렬  $\mathbf{K}$ 에 대하여 해가 언제나 존재하게 된다[4].

따라서 선형 판별 함수의 모든 학습 방법을 그대로 커널 공간에 적용할 수 있게 된다. 커널의 최대 강하/상승법은 아래와 같다[4].

$$\Delta \alpha_j = \eta (d_j - \boldsymbol{\alpha}^t \mathbf{K}(:, j)). \quad (3.5)$$

하지만 주어진 패턴  $\mathbf{y}_j$ 에 대해 벡터  $\boldsymbol{\alpha}$ 의 요소 중에서  $\alpha_j$ 만 학습하는 것으로 이 방법은 최대 강하 학습법이 아니라 좌표 강하법이다[4].

KR은 주어진 가중치  $\boldsymbol{\alpha}$ 를 학습하는 방법으로 유도 과정은 다음과 같다. 식 (3.3)에 의해 오류 비용 함수(error cost function)를 아래와 같이 정의한다[7].

$$E = \xi^2 = \frac{1}{2} \|\mathbf{d} - \mathbf{K}\boldsymbol{\alpha}\|^2. \quad (3.6)$$

$$\frac{\partial E}{\partial \boldsymbol{\alpha}} = (\mathbf{d} - \mathbf{K}\boldsymbol{\alpha})(-\mathbf{K}). \quad (3.7)$$

따라서 위 식은 주어진 가중치  $\boldsymbol{\alpha}$ 를 학습하는 아래의 식으로 변경되며, 신경망에서 델타 규칙(delta rule)으로 알려져 있다.

$$\Delta \alpha_j |_{\mathbf{y}_j} = \eta (d_j - \boldsymbol{\alpha}^t \mathbf{K}(:, j)) \mathbf{K}(:, j). \quad (3.8)$$

### 4. 동적 모멘트를 이용한 학습방법

KR은 KA보다 적은 수의 SVs만으로도 보다 효과적인 학습 성능을 얻을 수 있는 방법이지만 수렴속도가 느려 KA보다 많은 학습 시간을 요한다[3].

선형 판별함수의 학습법에서 완화는 다른 학습법에 비해 보다 유연한 초평면을 가지고 수렴하기때문에 많은 학습 시간이 소요된다[2]. KR은 완화를 커널 공간으로 확장한 방법으로 수렴점에 대해 매우 유연한 초평면으로 정의되며 결국 수렴점에 가까워질수록 발진현상이 일어난다. 이러한 발진현상을 개선하기 위한 방법으로 학습률의 조절과 모멘트 항의 도입이 소개되었으며 이러한 모멘트를 이용한 SVM의 학습 성능 향상은 KA를 이용한 방법이 있다[7].

기존의 모멘트를 이용한 학습 방법은 단순히 선형판별함수의 학습법에서 제안된 모멘트 항을 KA알고리즘에 적용한 것으로 각 알고리즘에 대한 최적 회로의 수렴과 성능 향상을 위한 모멘트의 선택에 많은 어려움을 가진다. 여기에 정적 모멘트를 매우 완만한 곡선 형태로 수렴에 근접해가는 최대 강하 학습법인 KR에 적용하였을 경우 아래와 같은 두 가지 문제점이 발생한다.

첫째, 수렴을 위해 필요한 충분히 작은 모멘트를 선언했을 경우로 학습 초기 모멘트의 영향을 거의 받지 못함으로 수렴 시간과 성능면에서 정적모멘트 사용이전에 비해 향상된 변화를 기대 할 수 없다는 것이다.

둘째, 초기 학습의 속도를 높이기 위해 수렴점에서의 모멘트보다 큰 모멘트가 주어졌을 경우이다. 이 경우 모멘트 사용 이전보다 빠른 수렴 속도를 얻을 가능성은 있지만 수렴점 부근에서는 과도한 모멘트가 부과되어 과잉학습을 하게 되는 오류를 범할 수 있다. 결국 정적모멘트는 초기 선언된

모멘트가 전체 학습에서 균등하게 적용됨으로써 KR의 특성인 수렴점에 대해 매우 유연한 곡선으로 근접하는 학습법에 대해서는 수렴속도와 성능 개선에 대한 효과적인 대안이 되지 못한다. 따라서 최적의 모멘트는 정적모멘트에서 나타난 단점들을 보완하고 장점을 살릴 수 있는 모멘트 스케줄링 방법이 된다.

본 논문에서는 수렴에 따른 가변적인 모멘트를 이용하여 SVM의 학습 성능을 개선하는 새로운 방법을 제안한다. 이러한 스케줄링 방법을 위해 본 논문에서는 학습 횟수와 성능을 한정적 비례관계로 가정한다. 한정적 비례관계란 전체 학습에 있어서의 학습 성과와 학습 횟수의 관계를 말하는 것이 아니라 초기 한정된 구간내의 학습에서 최초 학습 성능 향상을 기대할 수 있는 임의의 학습 횟수와 성과와의 관계를 말하며 이때 학습 성능은 학습 횟수와 비례관계를 가지게 된다. 즉, 한정적인 비례관계를 갖는 구간에서는 일정 범위 내에서 학습 성능은 학습 횟수에 따라 증가하게 된다. 따라서 순차 학습법의 발전은 초기 한정적 비례관계의 구간에서 또 다른 한정적 비례관계로 가는 중간 과정에서 나타나는 일련의 현상으로 볼 수 있다. 이러한 일련의 현상들은 다음 한계점까지의 도달 간격을 조정하여 간격을 좁힐수 있으며 이는 곧 학습에서 발전을 억제하면 성능 개선을 가져온다. 결국 동적모멘트를 이용한 학습 방법은 학습 초기 모멘트의 크기를 크게하여 수렴 속도를 높였으며 점차 학습 수행 횟수가 많아져 수렴에 가까워질수록 모멘트의 크기는 정밀하게 조정된다. 일반적으로 모멘트 향이 학습률을 증가할 경우 시스템이 국부적인 변화에 덜 민감해지는 것으로 알려져 있다[11].

하지만 학습 횟수에 따라 연속적으로 모멘트를 증가시킬 경우 모멘트는 단조 증가 함수가 되어 학습 횟수가 매우 많아지게 되며 과거의 학습이 현재의 학습을 능가하게 되어 학습 성과와 수렴 속도에 대해 개선효과를 가져오지 못하게 된다. 따라서 제안된 동적모멘트를 이용한 학습법은 조정된 모멘트의 크기가 초기 최대 모멘트 크기보다 작은 범위내에서 조정되게 된다. 결국 KR의 수렴특성에 따라 초기 모멘트 값은 학습 횟수에 따른 증가를 보이며 수렴에 가까워질수록 다시 모멘트향의 크기를 최대 모멘트 크기 이내에서 결정되게 함으로써 전체적으로 KR의 수렴속도를 향상시키고 성능을 개선시키는 방법이 된다. 본 논문에서 제안하는 모멘트 향은 **M**으로 아래와 같이 정의된다.

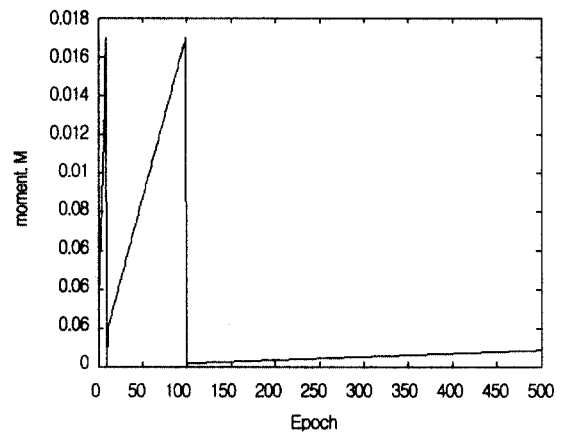
$$\begin{aligned}
 \mathbf{M} &= \frac{m(k+1)}{\tau} \\
 &\text{if } \mathbf{M} > m \\
 \mathbf{M} &= 0 ; \\
 \tau &= \tau^2 ;
 \end{aligned}
 \tag{4.1}$$

여기서  $\tau$ 는 초기 모멘트의 값의 크기를 최소화하기 위해 사용한 제어 변수로 예상된 최초 한계치에서의 학습 횟수가

되며, 본 논문에서는 성능 향상값인  $T = 10$ 의 값을 이용하였다. 보다 상세한 설정 방법은 이후 실험 부분에서 설명된다.  $m$ 은 기존의 상수 모멘트로 동적 모멘트를 이용한 학습 방법에서는 적용될 모멘트의 상한값이 된다. 결국 동적모멘트  $\mathbf{M}$ 의 값이 모멘트의 상한값인  $m$ 보다 클 경우 동적모멘트를 다시 0으로 하고  $\tau$ 를  $\tau^2$ 으로 다음번의 예상된 한계치를 최대로 돕으로써 예상된 한계치 내에서 다시 모멘트를 증가하면서 학습하게 한다.

즉,  $\mathbf{M}$ 이 단조 증가되는 것을 제한함으로써 기존의 상수 모멘트인  $m$ 을 초기 선언된 모멘트 내에서 가변적 변화가 가능하도록 하였다. 이때 처음 정의된 한계치에서의 모멘트의 증가는  $\frac{m(k+1)}{\Delta\tau}$ 의 증가치를 가지며 학습하게 된다.

이러한 학습 방법은 사용자 정의에 의해 정적 모멘트로 결정하기 어려운 KR의 유연한 초월면에 대한 모멘트의 값을 지정할 수 있다. 또한 기존의 모멘트 값의 변경에만 의존하던 모멘트 향의 정의에 예상된 한계치에 대한 학습 횟수를 추가함으로써 보다 적극적인 모멘트 스케줄링을 정의 할 수 있다. 다음은 학습에 참여하지 않은 단순한 모멘트변화를 나타낸 그래프이다.



(그림 1) 모멘트변화 그래프

(그림 1)은  $\tau = 10$ 일때 모멘트의 변화를 나타낸 것으로 가로축은 학습 횟수, 세로축은 모멘트의 변화를 나타낸다. 여기서  $M$ 가 상한값이 될때  $M$ 는 다시 초기값을 갖게되며, 이때  $\tau$ 는 각각 10과 100이 되며 다음번  $\tau$ 는 10000이되나 (그림 1)에서는 제외하였다. 즉,  $\tau$ 와 모멘트 향의 관계를 보여주는 그래프로 학습 횟수가 증가함에  $\tau$ 의 영향으로 100번째 학습이후에는 상당히 완만한 경사를 이루며 증가하는 것을 볼 수 있다. 결국 이러한 모멘트의 스케줄링 방법은 최대 강하 학습법인 KR의 수렴곡선이 가지는 미세한 발전현상을 줄여 수렴 속도의 개선을 가져온다. 여기에서 정의되는 모멘트의 값은 기존의 정적모멘트에서 사용하는 수동적인 방법으로는 예측 불가능하다.

### 5. 실험 및 결과

#### 5.1 실험 환경

본 실험은 Windows 2000 서버를 사용하였으며, 시스템의 구현을 위해 Matlab Ver. 6.1을 사용하였다. 본 실험에서는 사용한 데이터는 신경망 분류기 표준 평가 데이터인 SONAR 데이터[9]로 차원의 수는 60차원, 데이터의 총 개수는 208개로 처음 104개를 학습 데이터로 나머지 104개를 테스트 데이터로 사용하여 실험하였다. 첫 번째 실험은  $\tau$ 가 학습에 미치는 영향을 알아보는 실험으로 최적의  $\tau$ 를 선택할 경우 성능 향상에 기여하였다.

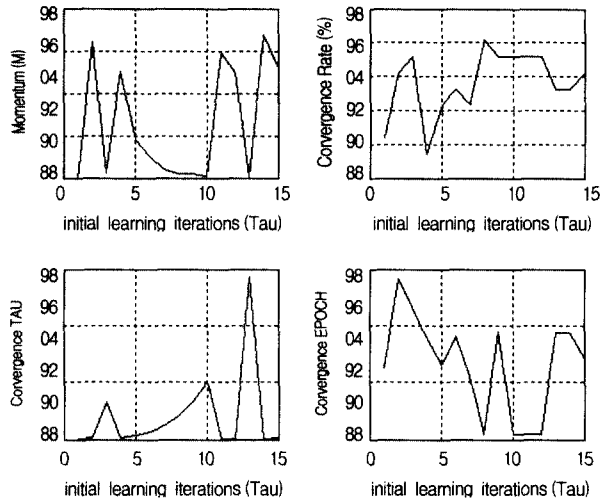
본 논문에서 정의한  $\tau$ 는 모멘트가 적극적으로 학습에 참여하게 되는 시점을 조절하는 변수로 최적의  $\tau$ 의 값은 기준치 이상의 학습 성능에 대해 상승 시점에 위치한다. 동시에 전체적인 학습 성능의 안정성을 찾아가는 초기시점이 된다. 이를 시점 대신 구간으로 볼때 한정적 비례관계가되며 이 구간에서의 학습 성능은 (그림 2)와 함께 <표 2>의 실험 결과에서와 같이  $\tau$ 가 8에서 12사이의 값을 가질때 유사한 학습 성능 향상이 있는 것을 알 수 있다. 하지만 KR은 전체 데이터를 한꺼번에 학습하여 가중치를 증가시키는 Batch Relaxation 대신 메모리의 한계를 극복하기 위해 Single-Sample Relaxation 방법을 적용한 학습법이다[2]. 따라서 학습에 참여하는 데이터 양과 학습 순서에 따라 수렴전의 각 학습 Epoch에 대해 다른 학습율을 나타낼 수 있다. 결국 최적의  $\tau$  값을 정확히 예측하는 것은 매우 어려운 문제가 된다. 하지만 본 논문에서는 이러한 타우 값의 예측범위를 좁혀 나가는 방법으로 다음과 같이 제안한다.

첫째, 초기의 현저히 불안정한 학습 상태에 있는 최소의 Epoch를 예측 범위의 하한선으로 정한다. 이는 모멘트의 가중 시점을 안정한 상태에 근접하도록하여 보다 신뢰성있는 학습 가중치를 이용하기 위함이다.

둘째, 최대 학습 Epoch의 10%이내에서  $\tau$ 를 결정한다. 식 (4.1)은 KR의 수렴조건에 적용시키기 위해  $\tau$  증가를  $\tau^2$ 으로 정의하였다. 그러나  $\tau$ 가 지나치게 큰 값을 갖게되면 초기 모멘트 항  $\mathbf{M}$ 이 너무 작은 값을 가지게되므로 수렴 속도 개선에 도움을 주지 못하게 된다. 또한 너무 작은  $\tau$ 의 값을 갖게 될경우 빈번한  $\tau$ 의 변경이 이루어지며 이때  $\tau$ 의 변화는 급격한  $\tau$ 값의 상승으로 이어져 수렴 속도 개선에 도움을 주지 못하게 된다.

따라서 전체 학습에 있어 2~3번 정도의 변화를 가져올 수 있도록 최장 학습 Epoch의 10%이내에서  $\tau$ 를 결정하는 것이 보다 효과적이다.

본 논문의 실험에 적용한  $\tau$  값은 이상의 두 가지 조건을 모두 만족시키며 동시에 모멘트 항  $\mathbf{M}$ 의 값을 쉽게 예상할 수 있는 값으로  $\tau = 10$ 으로 두었으며 실험 결과 기존의 정적모멘트를 이용한 결과 보다 우수한 성능을 얻을 수 있었다.



(그림 2)  $\tau$  변화에 따른 상태그래프

<표 2>  $\tau$ 의 변화에 따른 상태표

$\tau$	State	CT	CE	M	CTRD	CTSD
1	1	1	163	0	99.04	90.38
2	256	241	1.600391e-02		99.04	94.23
3	6561	216	5.596708e-04		100	95.19
4	256	190	1.261719e-002		99.04	89.42
5	625	166	4.515200e-003		100	92.31
6	1296	191	2.505401e-003		100	93.27
7	2401	155	1.097459e-003		100	92.31
8	4096	105	4.357910e-004		100	96.15
9	6561	195	5.052583e-004		100	95.19
10	10000	105	1.785000e-004		100	95.19
11	121	105	1.475207e-002		100	95.19
12	144	105	1.239583e-002		100	95.19
13	28561	194	1.154721e-004		100	93.27
14	196	194	1.682653e-002		100	93.27
15	225	171	1.292000e-002		100	94.23
$\tau$	Tau		CE	Convergence Epoch		
CT	Convergence Tau		M	Momentum		
CTRD	Learning Rate of Convergence Training Data					
CTSD	Learning Rate of Convergence Testing Data					

또한 거의 모든 경우 학습 횟수 5~15사이에서 최초 학습 성능이 향상 단계에 있는 것을 실험을 통하여 알 수 있었으며 대부분의 경우  $\tau=10$ 에서 성능향상을 보였다. 그러나 문제에 따라  $\tau$ 는 실험을 통하여 사용자에게 의해서 재정의 될 수 있으며 이때 적절한  $\tau$ 의 선택은 전체 학습 성능의 향상을 가져올 수 있다. (그림 2)와 <표 2>를 통하여 실제 학습에

있어  $\tau$ 가 10일 경우보다  $\tau$ 가 8일 경우 성능이 더 높음을 알 수 있다. (그림 2)는  $\tau = 1$ 에서  $\tau = 15$ 까지의 변화에 따른 실제 적용모멘트(Momentum : M), 수렴율, 수렴시  $\tau$ 의 값 그리고 수렴시 학습 횟수(Epoch)를 나타낸 그래프이다.

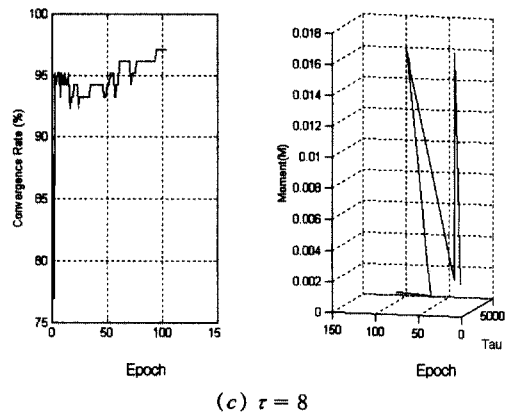
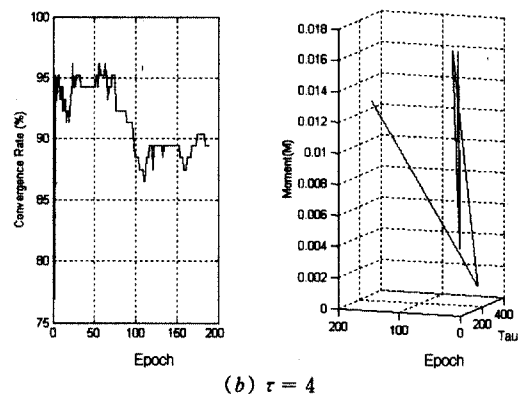
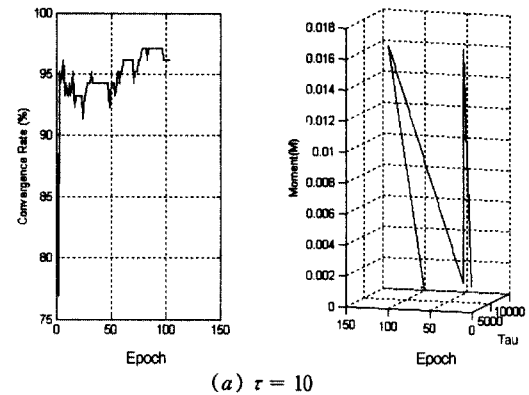
(그림 3)(a)는 Epoch에 따른  $\tau$ 의 값, 실제 적용 모멘트 M, 학습률을 나타낸 그래프로 본 논문의 실험에서 사용한  $\tau = 10$  일때의 상태 그래프이다. (그림 3)(a)를 통하여 알 수 있듯 실제 적용된 모멘트는  $1.785000e-004$ 으로 상당히 세밀하게 조정되는 것을 알 수 있으며 수렴시 적용모멘트가 (그림 1)에 나타난 것과 같이 Epoch 100에서 완만한 경사를 이루며 증가되는 것을 확인할 수 있다. 그림에서 좌측은 Epoch에 따른 학습률의 변화를, 그리고 우측은 Epoch에 따른  $\tau$ 의 변화와 실제 적용된 모멘트 M의 변화를 나타낸다. (그림 3)(b)와 (그림 3)(c)는 가장 낮은 학습 성능을 보인  $\tau = 4$ 에서의 상태 그래프와 최상의 학습 성능을 보인  $\tau = 8$ 에서의 각각의 그래프를 나타낸 것이다. (그림 3)(c)를 통하여 KR의 학습 성능이 모멘트에 따라 매우 민감하게 반응하는 것을 알 수 있다. (그림 3)(b)에서는 실제 적용된 모멘트가  $1.261719e-002$ 으로 (그림 3)(c)에 적용된 모멘트  $4.357910e-004$ 보다 상당히 큰 값을 가지는 것을 알 수 있다. 하지만 학습 성능은 단순히 모멘트의 크기에 따른 것이 아니라 한정적 비례관계에 따른  $\tau$ 의 결과와 최소한의 안정된 학습을 가져올 수 있는 학습 횟수에 있다. 이것은 <표 1>에 나타난 실험 결과에서 실제 적용된 모멘트 값 M의 크기가  $\tau = 12$ 에서  $\tau = 4$ 의 값과 유사한 값을 가지나 학습 성능에 있어서는 많은 차이를 보이는 것을 설명할 수 있다.

이러한 실험을 통하여 KR이 수렴점에 대해 매우 유연한 곡선의 형태를 취하기 때문에 수렴점 부근에서 큰 모멘트를 취할 경우 과잉 학습하게 된다는 것을 간접적으로 알 수 있으며  $\tau$ 값이 모멘트의 결정에 큰 영향을 주고 있으며, 따라서 학습 성능 향상에 많은 도움을 주고 있음을 알 수 있다.

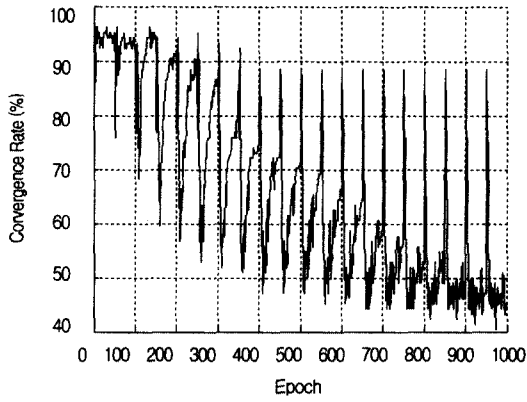
두 번째 실험은 제안된 동적모멘트의 학습 성능을 평가하기 위한것으로 모멘트 적용 이전의 KR과 정적모멘트, 동적모멘트를 이용한 KR의 각각에 대한 학습 성능을 비교하였다. 실험에 사용된 정적모멘트의 값은 0~2까지 0.1씩 증가시키며 각각의 모멘트 값에 대해 매회 50번씩 학습한 결과중 가장 우수한 학습 성능을 가진 모멘트를 선택하였다. 정적모멘트에 대한 첫 번째 실험으로 정적모멘트 0~2.0까지 Epoch 50당 정적 모멘트를 0.1씩 증가하였으며 약 Epoch 0~50에서 학습 성능이 가장 우수한 것으로 나타났다.

보다 정확한 모멘트 값을 구하기 위하여 Epoch 0~50에서의 정적모멘트 값인 0~0.1까지 0.01씩 증가시키고 이전 실험과 동일한 조건으로 실험한 결과 Epoch 250에서부터 진동이 심함을 볼 수 있었으며 성능이 가장 좋은 부분 역시 처음 실험과 동일하게 Epoch 0~50까지이며, 정적모멘트 값은 0~0.01이다. 실험에서 정적모멘트 값은 실험 결과 최적의 분

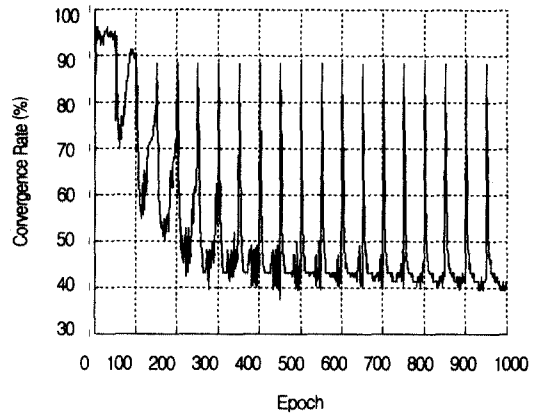
류성능을 보인  $4.0e-3$ 를 적용하였다. 동적모멘트의 최적 값을 산출하기 위한 실험 역시 정적모멘트와 동일하게 모멘트를 변화시켰다. 각각의 실험에서 최적의 분류 성능을 보인 부분을 재실험 하였으며 모멘트가 0에서 2사이에서는 Epoch 0~50에서 성능이 높음을 알 수 있었다. 정확한 동적모멘트 값을 알아보기 위해 모멘트를 세분하게 0.01씩 증가시켜 실험한 결과 동적 모멘트 0.1~0.2에서 최적의 성능을 확인할 수 있었다. 또한 동적모멘트를 변화하는 과정에서 학습 성능이 가장 높았던 Epoch 50~100에서 동적모멘트가  $1.7e-2$ 임을 살펴볼 수 있었다. 이하 실험에서 동적모멘트의 값은 실험 결과 최적의 분류 성능을 보인  $1.7e-2$ 를 적용하였다.



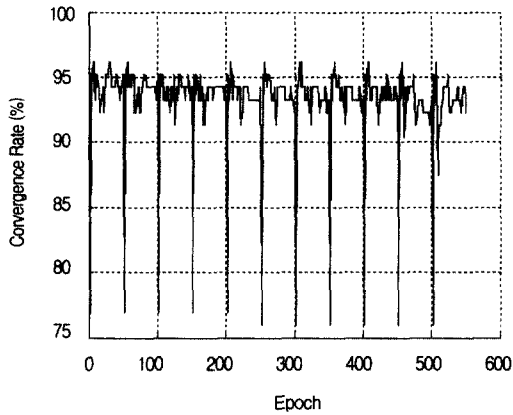
(그림 3) 타우( $\tau$ ) 값에 따른 상태 그래프



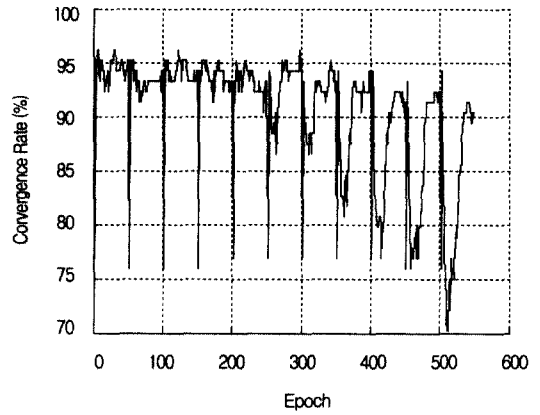
(a) 동적모멘트 : 0~2/0.1



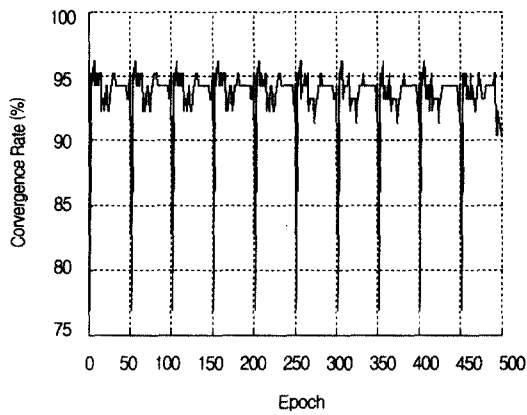
(b) 정적모멘트 : 0~2/0.1



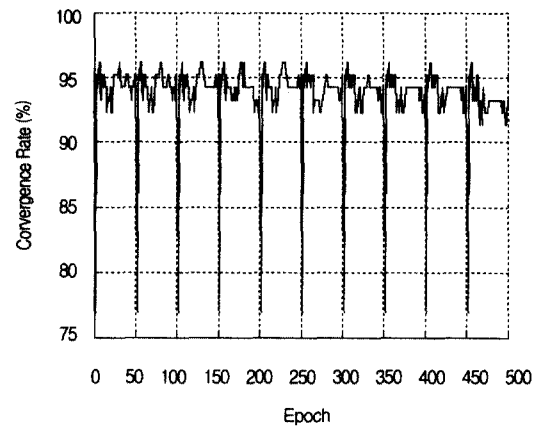
(c) 동적모멘트 : 0~0.1/0.01



(d) 정적모멘트 : 0~0.1/0.01



(e) 동적모멘트 : 0.01~0.02/0.001



(f) 정적모멘트 : 0~0.01/0.001

(그림 4) 동적모멘트와 정적모멘트 각각에 대한 비교그래프((a)~(f))

(그림 4)는 정적모멘트와 동적모멘트의 각각에 대한 그래프이다. (그림 4)(a)과 (그림 4)(b)는 동적모멘트와 정적모멘트의 0에서 2.0까지 0.1씩 증가시킨 결과를 비교한 그래프로 정적모멘트 그래프가 동적모멘트 그래프보다 Epoch 0~50 이상이 되었을 경우 전반적으로 학습 성능이 떨어짐을 확인할 수 있었다.

(그림 4)(c)과 (그림 4)(d)는 동적모멘트와 정적모멘트의 0~0.1까지 0.01씩 변화시켜 나타낸 결과 비교 그래프로 동적모멘트가 진동이 적은 반면 정적모멘트는 Epoch 250이상에서부터 진동 현상이 현저한 것을 볼 수 있다. 실험에서 알 수 있듯이 동적모멘트는 정적모멘트에 비해 학습 성능이 매우 안정적인 것을 알 수 있다. (그림 4)(e)와 (그림 4)(f)는

동적모멘트와 정적모멘트의 실험에서 가장 좋은 성능을 보인 부분을 비교한 그래프이다.

두 실험에서 최적의 성능을 보인 그래프를 비교한 결과 이미 동적모멘트는 0.01사이에서 안정성을 가진 반면 정적모멘트는 아주 세분화해서 그 값을 찾아내게 된다. 따라서 최적의 정적모멘트 값을 찾았을 경우 동적모멘트와 비교적 동등한 학습률을 가져올 수 있지만 정적모멘트의 경우 최적의 값을 구하기 위해서는 매우 복잡한 과정이 요구된다.

5.2 실험 결과

본 실험은 RBF함수에서 사용자 정의 인수  $\sigma$  값의 변화에 따른 학습 성능을 알아보기 위해  $\sigma$ 를 0.1에서 0.9까지 0.1씩 변화시켰으며 각 Epoch당 학습 성능을 알아보기 위해 Epoch를 50번에서 200번까지 변화시켜 실험하였다. 또한 동적모멘트 사용이전의 KR과 동적모멘트를 사용한 KRDM의 최적의 학습 성능을 알아보기 위해 이전의 실험에서 얻은 최적의 모멘트를 이용하여  $\sigma$ 각각에 대한 학습 성능을 비교하였다. 이하 모멘트 사용하지 않은 것을 NM(not momentum), 정적모멘트를 적용한 것을 SM(static momentum), 동적모멘트를 적용한 것을 DM(dynamic momentum)라 칭한다.

<표 2>  $\sigma$ 에 대한 학습 성능비교

(단위 : %)

$\sigma$		Epoch				수렴	%
		50	100	150	200		
0.3	NM (Training/Testing)	100 93.26	100 93.26	100 93.26	100 93.26	34	100 93.26
	DM (Training/Testing)	100 93.26	100 93.26	100 93.26	100 93.26	34	100 93.26
	SM (Training/Testing)	100 93.26	100 92.26	100 92.26	100 92.26	34	100 92.26
0.4	NM (Training/Testing)	100 95.19	98.07 90.38	99.03 88.46	99.03 90.38	163	99.03 90.38
	DM (Training/Testing)	100 95.19	100 96.15	100 95.19	100 95.19	105	100 95.19
	SM (Training/Testing)	100 95.19	99.03 92.30	100 94.23	100 94.23	160	100 94.23

<표 2>는 모멘트를 사용하지 않은 경우와 실험에서 얻은 최적의 모멘트를 사용한 기존의 정적모멘트 및 본 논문에서 제안한 동적모멘트 각각에 대한 학습 성능을 비교한 것이다.

전체적인 실험은  $\sigma$ 의 변화에 따른 학습 성능을 비교해 보는 것으로  $\sigma$ 의 변화를 0.1~0.9까지 변화시킨 결과 중 NM, SM, DM의 성능이 가장 좋았던  $\sigma$ 값이 0.3과 0.4일 경우만을 나타내었다.

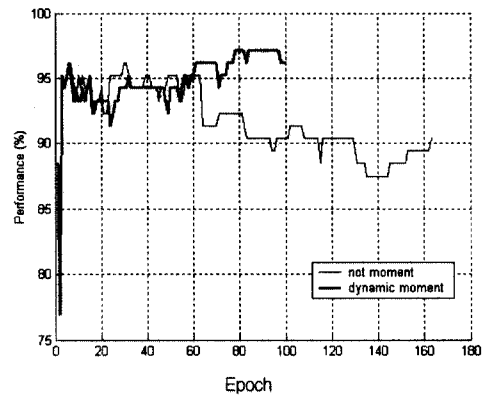
실험결과 훈련 데이터(training data)의 경우 수렴시 가장 높은 학습 성능은 100%로 나타났으며 실험 데이터(testing data)의 경우 모멘트를 사용하지 않은 경우 수렴시 가장 높은 학습 성능은  $\sigma=0.3$ 일때 93.26%, 정적모멘트를 사용했을 때  $\sigma=0.4$ 에서 최적 성능이 94.23%, 동적모멘트를 사용했을 때  $\sigma=0.4$ 에서 최적성능은 95.19%로 나타났다.

그 결과 학습률은 DM > SM > NM 순으로 나타났으며 전체적으로는 모멘트를 이용하여 학습하였을 경우 예상대로 보다 향상된 학습 성능을 보였으며 정적모멘트보다 본 논문에서 제안한 동적모멘트가 보다 우수한 것으로 나타났다.

실험 결과  $\sigma$ 값의 변화가 0.5이하에서 비교적 유사한 학습 성능을 보였으며 0.5이상에서는 학습 능력이 떨어지는 것을 알 수 있었다. 그러나 실험에서 모멘트를 이용하지 않은 KR의 학습 성능에서 알 수 있듯이  $\sigma$ 의 조정만으로는 성능 향상에 한계가 있음을 알 수 있다.

(그림 5)는  $\sigma=0.4$ 에서의 Epoch의 변화에 따른 학습 성능을 그래프로 나타낸 것이다.

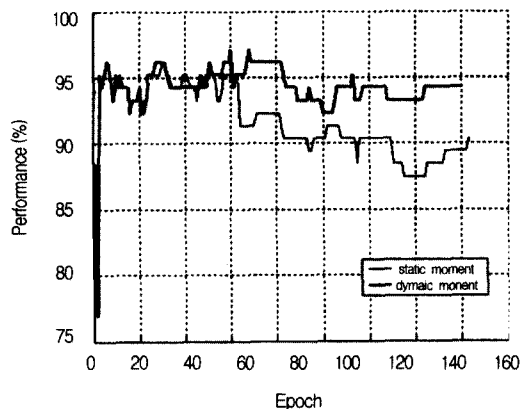
그림에서 굵은선이 동적모멘트를 사용한 KR의 학습 그래프이며, 가는선이 모멘트를 사용하지 않은 KR의 학습 그래프이다. 모멘트를 사용하지 않을때 Epoch 30정도에서 학습 성능이 높게 나왔으나 이것은 학습 중에 나타나는 발진에 불과하며 수렴에 근접해 갈수록 동적모멘트를 사용하였을 때의 학습 성능이 Epoch의 증가에 따라 향상되는 것을 볼 수 있었다.



(그림 5) NM과 DM의 비교

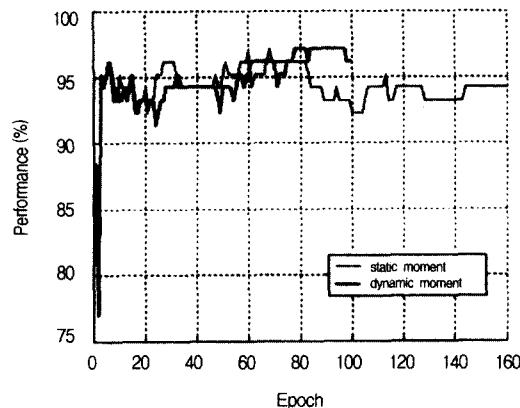
(그림 6)은 정적모멘트와 모멘트를 적용하지 않은 KR의 학습 성능을 Epoch단위로 비교해 본 결과이다. 이때 굵은선의 그래프는 정적 모멘트를 가는 선은 모멘트를 이용하지 않은 상태에서의 학습률을 나타낸다. (그림 6)에서와 같이 전체적으로 정적모멘트를 이용하여 학습 성능의 향상을 가져왔으나 수렴 속도는 약 160정도로 크게 변화가 없는 것을 알 수 있다.





(그림 6) NM과 SM의 비교

(그림 7)은 정적모멘트와 동적모멘트의 성능비교를 나타낸 그래프로 동적모멘트를 이용한 학습 방법이 정적모멘트를 이용한 학습 방법보다 우수한 학습 성능을 보였으며 수렴 속도가 두 배정도 빨라진 것을 알 수 있다.



(그림 7) SM과 DM의 비교

결과적으로 모멘트를 사용하지 않은 경우에 비해 모멘트를 사용한 경우 학습 성능의 향상을 가져옴을 알 수 있었으며 수렴점에 매우 유연한 초월면을 갖는 KR의 학습에 있어서 정적모멘트를 이용한 방법보다 동적모멘트를 이용한 학습 방법이 학습 성능과 수렴 속도 향상에 효과적인 방법임을 알 수 있다.

## 7. 결 론

본 논문에서는 SVM의 수렴률 개선을 위해 동적 모멘트를 이용한 학습 방법을 제안하였다. 제안된 학습 방법은 수렴에 따른 모멘트의 값을 동적으로 변경하는 방법이다.

제안된 동적모멘트를 이용한 학습법을 SVM의 순차 학습 방법인 KR에 적용하여 실험한 결과 제안된 동적 모멘트를 이용한 KR의 수렴률은 95.19%로 기존의 모멘트를 적용하지 않은 KR의 수렴률 93.26% 그리고 정적 모멘트를 적용한

수렴률 94.23%에 비해 수렴률이 개선되었으며 수렴 속도 또한 향상되었음을 확인하였다.

그러나 제안된 동적모멘트를 이용한 학습법에서 보다 향상된 학습 성능을 얻기 위해서는 최초의 모멘트 적용 시점인  $r$ 와 최대 모멘트인  $m$ 의 결정에 대한 보다 체계적인 연구가 추가적으로 필요하며, 제안된 동적모멘트를 이용한 학습법이 모든 순차학습에서 효과적인 학습법으로 적용 가능하게 하기 위해서는 보다 다양한 응용분야에서의 연구가 필요하다.

## 참 고 문 헌

- [1] B. Boser, I. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," Fifth Annual Workshop on Computational Learning Theory, San Mateo, CA : Morgan Kaufmann, pp.144-152, 1992.
- [2] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," Advances in Kernel Methods-Support Vector Learning, B. Scholkopf, C. Burges, A. Smola, editors, MIT-Press, pp.185-208, 1998.
- [3] R. O. Duda, P. E. Hart, D. G. Stork, *Pattern Classification*, Second Edition by John Wiley and Sons, Inc, 2001.
- [4] 류재홍, 정종철 "커널 이완절차에 의한 커널 공간의 저밀도 표현 학습", 한국퍼지 및 지능시스템학회, 2001년도 추계 학술대회 학술발표논문집, 제11권 제2호, pp.60-64, Dec., 2001.
- [5] T. T. Friess, N. Cristianini, C. Campbell, "The Kernel-Adatron Algorithm : a Fast and Simple Learning Procedure for Support Vector Machines," in Shavlik, J., ed., Machine Learning : Proceedings of the 15th Int. Conf., Morgan Kaufmann Publishers, San Francisco, CA 1998.
- [6] D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd Ed., Addison-Wesley Publishing Company Inc, 1984.
- [7] S. Haykin, *Neural Networks, A Comprehensive Foundation* Second Edition by Prentice-Hall, Inc, 1999.
- [8] 조용현, "모멘트를 이용한 Support Vector Machines의 학습 성능개선", 정보처리학회논문지 제7권 제5호, pp.1446-1455, May, 2000.
- [9] Y. Li, et al. , "The relaxed online maximum margin algorithm," In Advances in NIPS 13, 1999.
- [10] E. Osuna, R. Freund, and F. Girosi. "Training support vector machines: an application to face detection," CVPR'97, 1997.
- [11] 이성환 편저, *패턴인식의 원리 II*, 홍릉과학출판사, 1994.
- [12] O. L. Mangasarian and D. R. Musicant, "Active Set Support Vector Machine Classification," Neural Information Processing Systems 2000(NIPS 2000), T. K. Lee, T. G. Dietterich and V. Tresp, editors, MIT Press, pp.577-583, 2001.
- [13] 김상운 편저, *패턴인식의 입문*, 홍릉과학출판사, 1997.

### 김 은 미

e-mail : em1342@hotmail.com

2001년 여수대학교 컴퓨터공학과(공학사)

2001년~현재 전남대학교 대학원 컴퓨터  
공학과 재학

관심분야 : 신경망, 패턴인식, 영상처리,  
전문가시스템 등

### 이 배 호

e-mail : bhlee@chonnam.ac.kr

1978년 한양대학교 전자공학과(공학사)

1980년 한국과학기술원 전기 및 전자공학  
과(공학석사)

1993년 미국 University of Missouri,  
Columbia(공학박사)

1980년~1983년 국방과학연구소

1993년~현재 전남대학교 전자컴퓨터정보통신공학부, 정보통신  
연구소 부교수

관심분야 : 영상처리, 컴퓨터 비전, 인공지능, 패턴인식, 신경회로  
망 등