TÜBİTAK

Turkish Journal of Mathematics

http://journals.tubitak.gov.tr/math/

Research Article

Turk J Math
(2017) 41: 110 – 121

doi:10.3906/mat-1411-51

Near optimal step size and momentum in gradient descent for quadratic functions

Engin TAŞ[1,*], **Memmedağa MEMMEDLİ**[2]

[1]Department of Statistics, Faculty of Science and Literature, Afyon Kocatepe University, Afyonkarahisar, Turkey
[2]Department of Statistics, Faculty of Science, T.C. Anadolu University, Eskişehir, Turkey

Received: 22.11.2014 • **Accepted/Published Online:** 18.03.2016 • **Final Version:** 16.01.2017

Abstract: Many problems in statistical estimation, classification, and regression can be cast as optimization problems. Gradient descent, which is one of the simplest and easy to implement multivariate optimization techniques, lies at the heart of many powerful classes of optimization methods. However, its major disadvantage is the slower rate of convergence with respect to the other more sophisticated algorithms. In order to improve the convergence speed of gradient descent, we simultaneously determine near-optimal scalar step size and momentum factor for gradient descent in a deterministic quadratic bowl from the largest and smallest eigenvalues of the Hessian. The resulting algorithm is demonstrated on specific and randomly generated test problems and it converges faster than any previous batch gradient descent method.

Key words: Gradient descent, step size, momentum, convergence speed, stability

1. Introduction

In domains like statistics, finance, bioinformatics, information retrieval, collaborative filtering, and social network analysis, learning tasks such as regression, classification, and ranking start with a loss function that measures the error between the prediction of the model and the actual output value. An empirical risk function is then defined over a training data to estimate this loss accordingly. Consider, for example, least-squares regression; we seek the plane that minimizes the mean squared error between the predictions and the actual values of the response variables. In classification, we try to minimize the cost we pay for incorrectly assigning the observations to the wrong class. Ranking tasks are different than the regression and classification tasks where the empirical risk is defined as the normalized pairwise least-squares loss over the training data. However, all of these methods use numerical optimization algorithms, in one way or the other, to minimize the empirical risk.

Learning a regression, classification, or ranking function from data requires evaluation of the objective function. This involves basically the summation of squared errors over the training dataset used in building the model. Gradient-based methods must compute this sum for each evaluation of the empirical risk, respectively its gradient, whereas standard numerical optimization techniques such as variations of Newton's method and conjugate gradient algorithms also need second-order information [4]. As available data sets grow ever larger and/or when there are many parameters to be fit, such classical second-order methods are impractical in almost all useful cases. Gradient-based methods, by contrast, have a major advantage in large and redundant data sets with higher dimensionality. In fact, simple stochastic gradient descent outperforms sophisticated second-order batch methods in general, since the computational requirements of stochastic methods are extremely reduced by the fact that they only work with a single randomly picked example from the training data (e.g., [2, 9, 14]).

*Correspondence: engintas@aku.edu.tr

Normally, we expect that any step in the negative gradient direction will take us closer to the global minimum, but, for real problems, error surfaces are typically complex and may have numerous local minima. Therefore, the risk of being stuck in a local minima is much higher in real-life problems. The inclusion ofa momentum term can help us to escape from these local minima and probably it is the most popular extension of the gradient descent algorithm. This generally leads to a significant improvement in the performance of the gradient descent but introduces a second parameter whose value needs to be chosen, in addition to that of the step size parameter. There have been numerous studies on the stability and convergence speed of the gradient descent with momentum (GDM) algorithm and research continues. Since the squared error function is approximately quadratic around a local minimum, recent studies focused on the analysis of quadratic error functions [1, 10, 12].

This paper starts with clarifying some results in [10, 12], and then, for a given step size, the changing intervals of the momentum factor, which ensures stability, are determined using a different approach other than the previous studies. Based on the suggestions in [10, 12] on how the choice of the momentum factor affects the convergence speed of GDM, this study proposes a near optimal step size and a corresponding momentum factor that improves the convergence speed much more.

At first, by considering the physical interpretation of GDM in [10], the variation intervals of the momentum factor are analyzed by examining the stability problem for the parametric form of the algorithm used in [12]. The results of [10, 12] and this study are compared for stability and convergence speed, and a better way of parameter selection is proposed. Consequently, suitable formulas are derived for a near optimal step size and a corresponding momentum factor that can significantly increase the convergence speed of GDM.

2. Stability

GDM can be written as

$$x_{t+1} = [(1+\mu)I - (1-\mu)\eta H]x_t - \mu x_{t-1} + (1-\mu)\eta b, \tag{1}$$

for the minimization of the following deterministic error function

$$F(x) = \frac{1}{2}x^T Hx - b^T x + c, \tag{2}$$

where I is an $n \times n$ identity matrix, η is the step size, μ is the momentum factor, H is an $n \times n$ symmetric positive definite matrix, b is an n-dimensional vector, and c is a given constant. The gradient of the quadratic function F at point x is $\nabla F(x) = Hx - b$. Since H is symmetric and positive definite, it can be diagonalized as

$$H = QKQ^T, \quad QQ^T = I,$$

where Q is a matrix formed by the orthonormal eigenvectors of H, and K is a diagonal matrix formed by the eigenvalues $\kappa_i > 0, i = 1, 2, \ldots, n$ of H. Applying the transformation $x' = Q^T x$, (1) becomes

$$x'_{t+1} = [(1+\mu)I - (1-\mu)\eta K]x'_t - \mu x'_{t-1} + (1-\mu)\eta b', \tag{3}$$

where $b' = Q^T b$. (3) is written in coordinates as

$$x'_{i,t+1} = [1 + \mu - (1-\mu)\eta\kappa_i]x'_{i,t} - \mu x'_{i,t-1} + (1-\mu)\eta b'_i \quad i = 1, 2, \ldots, n; \tag{4}$$

then the coordinates of vector x are obtained by the linear combination of the coordinates of x'. Including the dummy equation $x'_{i,t} = x'_{i,t}$, we can write (4) in the form

$$\tilde{x}'_{i,t+1} = P_i \tilde{x}'_{i,t} + d_i, \quad \tilde{x}'_{i,t} = \begin{pmatrix} x'_{i,t-1} \\ x'_{i,t} \end{pmatrix} \quad i = 1, 2, ..., n, \tag{5}$$

where $P_i = \begin{pmatrix} 0 & 1 \\ -\mu & 1 + \mu - (1-\mu)\eta\kappa_i \end{pmatrix}$ is a 2×2 matrix, and $d_i = \begin{bmatrix} 0 \\ (1-\mu)\eta b'_i \end{bmatrix}$ is a two-dimensional vector $(i = 1, 2, ..., n)$. The linear dynamic system given by (5) is stable if the magnitudes of the eigenvalues of the P_i matrix are smaller than 1 [3]. Thus a relation is set up between the stability problem of the GDM algorithm (1) and the magnitudes of the eigenvalues of the P_i matrix.

We can write the corresponding characteristic equation for finding the eigenvalues of the P_i matrix and we have that the eigenvalues λ of the P_i matrix are the roots of the following quadratic equations [10, 12]:

$$\lambda^2 - [(1+\mu) - (1-\mu)\eta\kappa_i]\lambda + \mu = \mathbf{0}, \quad i = 1, 2, ..., n. \tag{6}$$

Two roots (real or complex) of (6) correspond to each κ_i $(i = 1, 2, ..., n)$. For the stability of the linear iterative process (5), the magnitude of each root of (6) must be smaller than 1. Therefore the stability problem of gradient descent with momentum algorithm (1) becomes the examination of (6). The roots of (6) corresponding to any κ eigenvalue of H matrix are calculated as

$$\lambda = \frac{[(1+\mu) - (1-\mu)\eta\kappa] \pm \sqrt{[(1+\mu) - (1-\mu)\eta\kappa]^2 - 4\mu}}{2}. \tag{7}$$

[12] examined the stability of the algorithm (1) by using (7), whereas we examined the stability of the GDM algorithm by considering the quadratic function on the left-hand side of (6) (with respect to λ):

$$\phi(\lambda) = \lambda^2 - [(1+\mu) - (1-\mu)\eta\kappa]\lambda + \mu. \tag{8}$$

This approach is apparent from a geometric perspective and it facilitates the determination of a near-optimal step size and a corresponding momentum factor, which we will study in the next section. The discriminant of the quadratic form (8) is $D = [(1+\mu) - (1-\mu)\eta\kappa]^2 - 4\mu$, and if we write according to the degrees of the momentum factor μ then

$$D(\mu) = (1+\eta\kappa)^2\mu^2 - 2(1+\eta^2\kappa^2)\mu + (1-\eta\kappa)^2. \tag{9}$$

In the case of $D < 0$, the roots of (6) are conjugate complex numbers and their magnitudes are constants equal to $|\lambda| = \sqrt{\mu}$. The quadratic form (9) has two distinct roots in the range $[0, 1]$: $\mu_1 = \dfrac{(1-\eta\kappa)^2}{(1+\eta\kappa)^2}$ and $\mu_2 = 1$. Therefore, the sign of the function $D(\mu)$ is determined as

$$D(\mu) = \begin{cases} < 0, & S(\eta\kappa) < \mu < 1 \\ = 0, & \mu = 1 \text{ or } \mu = S(\eta\kappa) \\ > 0, & \mu > 1 \text{ or } \mu < S(\eta\kappa) \end{cases}, \tag{10}$$

where $S(\eta\kappa) = \dfrac{(1-\eta\kappa)^2}{(1+\eta\kappa)^2}$, η is the step size, and κ is any eigenvalue of the matrix H. $S(\eta\kappa)$ has the following properties:

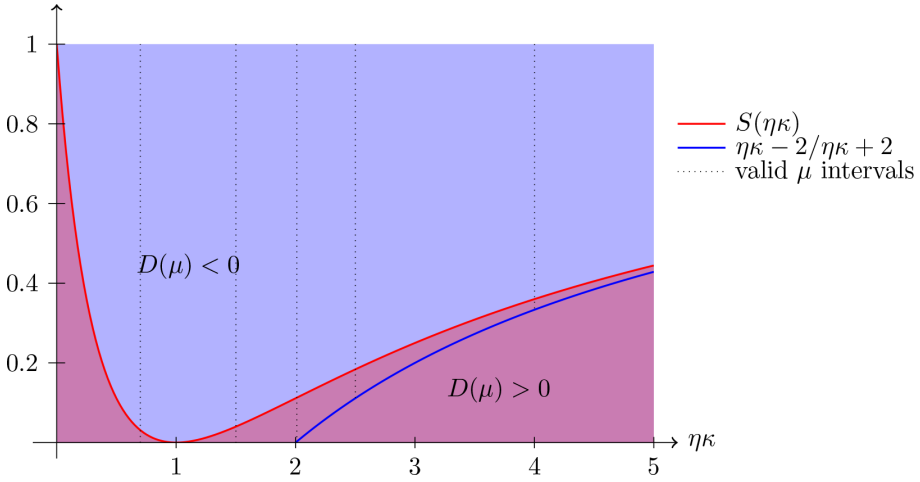

Figure 1. Plot of functions $S(\eta\kappa)$, $\dfrac{\eta\kappa - 2}{\eta\kappa + 2}$ and valid μ intervals.

$S(\eta\kappa)$ decreases from 1 to 0 in the segment $0 \leq \eta\kappa \leq 1$ and takes the minimum value 0 at $\eta\kappa = 1$, and increases when $\eta\kappa > 1$. $S(\eta\kappa)$ is convex in $0 \leq \eta\kappa \leq 2$, and concave in $(2, +\infty)$. $\eta\kappa = 2$ is the turning point (see Figure 1).

Following is a theorem that is similar to the stability results in [12], and the proof is given in a different way in this paper.

Theorem 1 (Stability) *Assume that η is the step size and κ_i, $i = 1, 2, ..., n$ are the eigenvalues of the symmetric positive definite matrix H. If $0 < \eta\kappa_i \leq 2$, $i = 1, 2, ..., n$ then the GDM algorithm (1) is stable for any momentum factor μ in the range $(0, 1)$; else if $\max_i \eta\kappa_i > 2$ then (1) is stable for any momentum factor μ in the range $\max_i \dfrac{\eta\kappa_i - 2}{\eta\kappa_i + 2} < \mu < 1$.*

Proof To prove the stability of the algorithm given by (1), it must be shown that the eigenvalues of matrix P are smaller than 1. The magnitude of any complex root of (6) is $|\lambda| = \sqrt{\mu}$, and it is smaller than 1, when condition $0 < \mu < 1$ is satisfied. Then it remains to show that the absolute value of any real root of the quadratic function (8) is smaller than 1. For the quadratic form $\phi(\lambda)$ defined by (8), $\phi(0) = \mu > 0$. The minimum of the function $\phi(\lambda)$,

$$\lambda_{min} = \frac{(1 + \mu) - (1 - \mu)\eta\kappa}{2}, \tag{11}$$

and the minimum value is

$$\phi(\lambda_{min}) = -\frac{D(\mu)}{4}. \tag{12}$$

Depending on the equality (12), in the case of $D(\mu) \geq 0$, we see that $\phi(\lambda_{min}) \leq 0$. By examining the eigenvalues λ according to $\eta\kappa$:

i. if $\eta\kappa = 1$ for $0 < \mu < 1$, then $D(\mu) < 0$, and λ eigenvalues are complex numbers, and in this case $|\lambda| = \sqrt{\mu} < 1$.

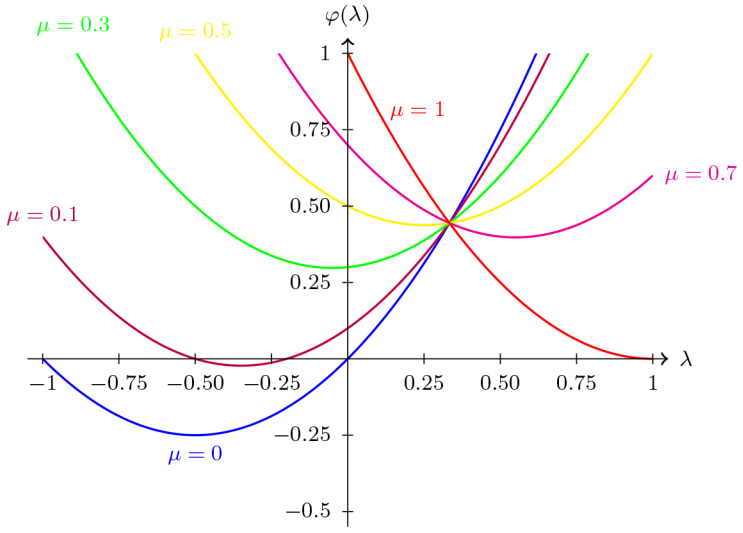

Figure 2. The plot of $\varphi(\lambda)$ for various values of μ between $[0,1]$.

ii. if $0 < \eta\kappa \leq 2$ and $\eta\kappa \neq 1$, then depending on the choice of $0 < \mu < 1$, $D(\mu)$ can be smaller than zero, larger than zero, or equal to zero. In this case, λ_{min} defined by (11) is evaluated as

$$\frac{3\mu - 1}{2} \leq \lambda_{min} = \frac{(1+\mu) - (1-\mu)\eta\kappa}{2} < \frac{1+\mu}{2}. \tag{13}$$

Since μ changes in range (0,1), we find the upper and lower bounds for λ_{min} from (13)

$$\frac{-1}{2} \leq \lambda_{min} < 1.$$

On the other hand,

$$\phi(1) = (1-\mu)\eta\kappa > 0, \quad (since\ \mu < 1),$$

and

$$\phi(-1) = 1 + (1+\mu) - (1-\mu)\eta\kappa + \mu = 2(1+\mu) - (1-\mu)\eta\kappa. \tag{14}$$

According to (14), $\phi(-1)$ is descending with respect to $(\eta\kappa)$ in $0 < \eta\kappa < 2$, and when $\eta\kappa = 2$,

$$\phi(-1) \geq 2(1+\mu) - 2(1-\mu) = 4\mu > 0. \tag{15}$$

Thus, when the condition $0 < \eta\kappa \leq 2$ is satisfied, $\phi(-1) > 0$ for any momentum factor in $0 < \mu < 1$. Now, in the case of $D(\mu) > 0$, we have $\phi(0), \phi(1), \phi(-1) > 0$ and $\frac{-1}{2} < \lambda_{min} < 1$. Therefore, the schematic plot of $\phi(\lambda)$ will be similar to one of the following plots in Figure 2, and in both cases we see that $\mid \lambda \mid < 1$.

iii. Now assume that the condition $\eta\kappa > 2$ is satisfied. Let us show that the absolute values of the real roots of the quadratic form (8) are smaller than 1 for momentum factors that satisfy $\frac{\eta\kappa - 2}{\eta\kappa + 2} < \mu < 1$. Then it is sufficient that the conditions $\lambda_{min} \in (-1, 0)$ and $\phi(-1) > 0$ are satisfied. According to (14), we have

$$\phi(-1) = (\eta\kappa + 2)\mu - (\eta\kappa - 2).$$

Thus, when $\eta\kappa > 2$, in order to $\phi(-1) > 0$, we see the necessity for the inequality

$$\mu > \frac{\eta\kappa - 2}{\eta\kappa + 2}, \tag{16}$$

to be satisfied. On the other hand, when $0 < \mu < 1$, the condition $D(\mu) > 0$ defined by formula (10) is satisfied when

$$\mu < \frac{(\eta\kappa - 1)^2}{(\eta\kappa + 1)^2}. \tag{17}$$

In the case of $\eta\kappa > 1$, we find

$$\lambda_{min} = \frac{1}{2}[(\eta\kappa + 1)\mu - (\eta\kappa - 1)] < \frac{1 - \eta\kappa}{1 + \eta\kappa} < 0,$$

from (11) and (17) and according to (13) and (16) we have

$$0 < (\eta\kappa + 2)\mu - (\eta\kappa - 2) = [(\eta\kappa + 1)\mu - (\eta\kappa - 1)] + \mu + 1 = 2\lambda_{min} + (\mu + 1).$$

Since μ takes values between 0 and 1, we find that $\lambda_{min} > -\frac{\mu+1}{2} > -1$. Therefore, the roots of (6) settle in $(-1, 0)$ while $\lambda_{min} \in (-1, 0)$ and the momentum factor μ changes in the range $\frac{\eta\kappa-2}{\eta\kappa+2} < \mu < 1$. Consequently, the iterative process (1) is stable while the condition $\max_i \frac{\eta\kappa_i - 2}{\eta\kappa_i + 2} < \mu < 1$ is satisfied.

□

Note 1. If we pay attention to the proof of theorem 1, we can see that the following statements are true:

- If $0 < \eta\kappa < 1$ and $D(\mu) > 0$, then the corresponding roots of (6) settle in $(0, 1)$.
- If $1 < \eta\kappa < 2$ and $D(\mu) > 0$, then the corresponding roots of (6) settle in $(-1, 0)$.

Note 2. Theorem 1 can be expressed in brief:
Assuming that η is the step size and $\kappa_i, i = 1, 2, ..., n$ are the eigenvalues of the symmetric positive definite matrix H, the GDM (1) is stable for the momentum factors in the range

$$\max\{0, \max_i \frac{\eta\kappa_i - 2}{\eta\kappa_i + 2}\} < \mu < 1.$$

Note 3. From the proof of theorem 1, it is clear that: When the momentum factor μ changes in $-1 < \mu < 1$, $\max\{0, \max_i \frac{\eta\kappa_i - 2}{\eta\kappa_i + 2}\} < \mu < 1$ is the necessary and sufficient condition for the stability of GDM (1).

In Figure 1, the variation interval of μ with respect to $\eta\kappa$ is demonstrated geometrically for the stability of (1).

3. Convergence speed

As explained in [12], the convergence speed of the algorithm (1) depends on the magnitudes of the λ eigenvalues, that is, the smaller the magnitude the faster the convergence. This implies that for a given step size the choice

$\mu = \max_i \frac{(1-\eta\kappa_i)^2}{(1+\eta\kappa_i)^2} = \max_i S(\eta\kappa_i)$ provides a better convergence speed in general. However, there is no examination of the suitable choice of step size η in [12]. In fact, a better choice of step size η should shrink the magnitudes of the λ eigenvalues much more. In this paper, we propose to determine $\eta = \eta^0$ from the following minimax problem:

$$\max_i \frac{(1-\eta^0\kappa_i)^2}{(1+\eta^0\kappa_i)^2} = \min_{\eta>0} \max_i \frac{(1-\eta\kappa_i)^2}{(1+\eta\kappa_i)^2}.$$

Thus, taking $\mu = \mu^0 = \max_i \frac{(1-\eta^0\kappa_i)^2}{(1+\eta^0\kappa_i)^2}$, a better convergence speed is ensured. Assume that the eigenvalues of the symmetric positive definite H matrix are ordered in this way: $0 < \kappa_n \leq \kappa_{n-1} \leq \ldots \kappa_2 \leq \kappa_1$, where κ_n is the smallest and κ_1 is the largest eigenvalue. In this case, the plot of functions $S_i(\eta) = S(\eta\kappa_i) = \frac{(1-\eta\kappa_i)^2}{(1+\eta\kappa_i)^2}$, $i = 1, 2, \ldots, n$ is illustrated in Figure 3. Let us mark the nonzero intersection point of functions $S_1(\eta)$ and $S_n(\eta)$ with $\eta_{1,n}$. From the equality $S_1(\eta) = S_n(\eta)$, i.e. $\frac{(1-\eta\kappa_1)^2}{(1+\eta\kappa_1)^2} = \frac{(1-\eta\kappa_n)^2}{(1+\eta\kappa_n)^2}$, we find that $\eta_{1,n} = \frac{1}{\sqrt{\kappa_1\kappa_n}}$.

Lemma *i. If* $< \eta \leq \frac{1}{\sqrt{\kappa_1\kappa_n}}$ *then* $S_n(\eta) \geq S_i(\eta)$, $i \in \{1, 2, \ldots, n-1\}$.

ii. If $\eta \geq \frac{1}{\sqrt{\kappa_1\kappa_n}}$ *then* $S_1(\eta) \geq S_i(\eta)$, $i \in \{2, \ldots, n\}$.

Proof

i. Consider the inequality $S_n(\eta) \geq S_i(\eta)$, $i \neq n$, i.e.

$$\frac{(1-\eta\kappa_n)^2}{(1+\eta\kappa_n)^2} \geq \frac{(1-\eta\kappa_i)^2}{(1+\eta\kappa_i)^2}, \quad \eta > 0. \tag{18}$$

(18) is satisfied for any $\eta > 0$ when $\kappa_n = \kappa_i$. Therefore, we consider the case $\kappa_n < \kappa_i$, and (18) becomes

$$2\eta(\kappa_i - \kappa_n)(2 - 2\eta^2\kappa_i\kappa_n) \geq 0.$$

Since $\eta(\kappa_i - \kappa_n) \geq 0$, we have $2 - 2\eta^2\kappa_i\kappa_n \geq 0$, and for $0 < \eta \leq \frac{1}{\sqrt{\kappa_i\kappa_n}}$ inequality (18) is satisfied. On the other hand, since $\frac{1}{\sqrt{\kappa_1\kappa_n}} \leq \frac{1}{\sqrt{\kappa_i\kappa_n}}$ $i \in \{2, \ldots, n-1\}$, the inequality $\eta \leq \frac{1}{\sqrt{\kappa_i\kappa_n}}$ $i \in \{2, \ldots, n-1\}$ is satisfied when $\eta \leq \frac{1}{\sqrt{\kappa_1\kappa_n}}$. Finally, for $0 < \eta \leq \frac{1}{\sqrt{\kappa_1\kappa_n}}$, (18) is also satisfied, that is, $S_n(\eta) \geq S_i(\eta)$ $i = 1, 2, \ldots, n$.

ii. Consider the inequality $S_1(\eta) \geq S_i(\eta)$, $i \neq 1$, i.e.

$$\frac{(1-\eta\kappa_1)^2}{(1+\eta\kappa_1)^2} \geq \frac{(1-\eta\kappa_i)^2}{(1+\eta\kappa_i)^2}, \quad \eta > 0.$$

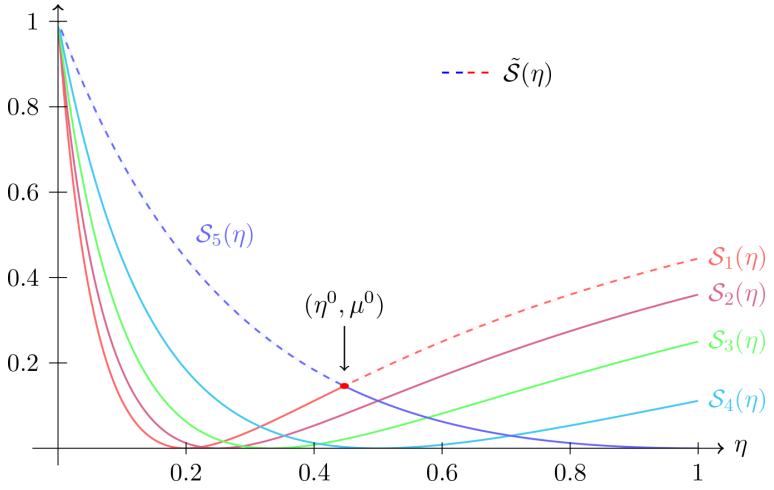

Figure 3. Plots of $S_i(\eta)$ and $\tilde{S}(\eta) = \max_i S_i(\eta)$ and near optimal step size η^0 and momentum μ^0.

In this case, we have

$$2\eta(\kappa_i - \kappa_1)(2 - 2\eta^2\kappa_i\kappa_1) \geq 0.$$

Since $\eta(\kappa_i - \kappa_1) \leq 0$, we have $2 - 2\eta^2\kappa_i\kappa_1 \leq 0$ and we conclude that the inequality $S_1(\eta) \geq S_i(\eta)$ is satisfied for $\eta \geq 1/\sqrt{\kappa_1\kappa_i}$. On the other hand, since $\frac{1}{\sqrt{\kappa_1\kappa_i}} \leq \frac{1}{\sqrt{\kappa_1\kappa_n}}$ $i \in \{2, \ldots, n-1\}$, the inequality $\eta \geq \frac{1}{\sqrt{\kappa_1\kappa_i}}$ $i \in \{2, \ldots, n-1\}$ is satisfied when $\eta \geq \frac{1}{\sqrt{\kappa_1\kappa_n}}$. Therefore, for $\eta \geq \frac{1}{\sqrt{\kappa_1\kappa_n}}$, $S_1(\eta) \geq S_i(\eta)$ $i = 2, 3 \ldots, n$.

□

Theorem 2 *$\eta^0 = 1/\sqrt{\kappa_1\kappa_n}$ is the unique solution of the minimax problem*

$$\min_{0<\eta} \max_i S_i(\eta) = \max_i S_i(\eta^0).$$

Proof According to the Lemma, it is clear that

$$\tilde{S}(\eta) = \max_i S_i(\eta) = \begin{cases} S_n(\eta), & 0 < \eta < \eta^0 \\ S_1(\eta), & \eta \geq \eta^0 \end{cases}.$$

$\tilde{S}(\eta)$ is shown with dashed curves in Figure 3. Now we can show that η^0 is the minimum of $\tilde{S}(\eta)$. Considering the properties of $S(\eta\kappa)$, $S_n(\eta) = S(\eta\kappa_n)$ is decreasing in $0 < \eta\kappa_n \leq 1$. Since $\eta^0\kappa_n = \frac{\kappa_n}{\sqrt{\kappa_1\kappa_n}} \leq 1$, $S_n(\eta)$ takes its minimum at $\eta = \eta^0 = \frac{1}{\sqrt{\kappa_1\kappa_n}}$ in $0 < \eta \leq \eta^0$. In the same way, it is clear that $\eta = \eta^0$ is also the minimum of $S_1(\eta)$ in $\eta^0 \leq \eta < +\infty$. Thus, $\eta = \eta^0$ is the minimum of $\tilde{S}(\eta)$ and the corresponding momentum factor is $\mu^0 = \tilde{S}(\eta^0) = \frac{(\sqrt{\kappa_1} - \sqrt{\kappa_n})^2}{(\sqrt{\kappa_1} + \sqrt{\kappa_n})^2}$.

□

Eventually, near-optimal step size and momentum factor are given in the following formulas:

$$\eta^0 = \frac{1}{\sqrt{\kappa_1 \kappa_n}}, \quad \mu^0 = \frac{(\sqrt{\frac{\kappa_1}{\kappa_n}} - 1)^2}{(\sqrt{\frac{\kappa_1}{\kappa_n}} + 1)^2}, \quad \kappa_n > 0. \tag{19}$$

This near-optimal learning parameter pair (η^0, μ^0) is calculated for the simple preliminary problem given in the first row of Table 4, and it is indicated with a red point in Figure 3. Furthermore, one can obtain the following relations using (19):

$$(1 - \mu^0)\eta^0 = \frac{4}{(\sqrt{\kappa_1} + \sqrt{\kappa_n})^2}, \tag{20}$$

$$\text{if} \quad \frac{\kappa_1}{\kappa_n} \to 1, \quad \text{then} \quad \mu^0 \to 0, \tag{21}$$

$$\text{if} \quad \frac{\kappa_1}{\kappa_n} \to \infty, \quad \text{then} \quad \mu^0 \to 1. \tag{22}$$

(20) says that only the largest and smallest eigenvalues of the Hessian matrix have an effect on the gradient term when $\eta = \eta^0$ and $\mu = \mu^0$ used. Step size, which depends on the gradient, shrinks when the relevant eigenvalues of the Hessian matrix get larger. (21) shows that the effect of the momentum factor decreases when the eigenvalues of the Hessian matrix are close to each other, whereas the effect of the momentum factor increases, according to (22), when the range of the eigenvalues of the Hessian matrix is large.

4. Numerical results

In the first phase, best step size and momentum factor were found by trying all possible combinations in a valid interval of parameter pairs. This was done by first setting the step size η sequentially to a fixed value in $[0.01, 1.00]$, which is a reasonable range to ensure convergence in the test problems, and for each step size the momentum factor was chosen from [0.01, 0.99] sequentially. Thus, we had tested all possible combinations of the step size and momentum pairs, and the pair that gave the best speed of convergence was obtained. Then near-optimal step size and a corresponding momentum factor were calculated from (19), and the resulting algorithm (1) was executed with these parameters.

The results obtained are summarized in Table 4. We must keep in mind that the parameters that were found by trials had taken serious time and reasonable parameter range can differ from one problem to another. The results indicate that the near-optimal step size and momentum factor are close to the best parameter pair and there are no significant differences between the performances of the best parameter pairs and the proposed near-optimal parameter pairs.

In the second phase of the experiments, we modified GDM as to work with near-optimal step size and momentum factor, and the modified algorithm is named eGDM. Performance of the algorithm eGDM is compared with that of a conventional GDM and gradient descent with adaptive learning rate and momentum (GDX) algorithm on randomly generated small and medium scale test problems for a quadratic function. The problems are generated to allow the user to control the dimensionality of the problem d (e.g., the dimension of the weight vector) and the condition number of the Hessian matrix H. For simplicity, we have assumed that the stationary point of the quadratic function was at the origin, and that it had a zero value there. The terms

Table 1. Near optimal parameters vs. best learning parameters on simple problems.

Eigenvalues	Best step size and momentum factor found by trial			Near-optimal step size and momentum factor calculated by (19)		
	η	μ	iteration	η^0	μ^0	iteration
1, 2, 3, 4, 5	0.45	0.15	18	0.4472	0.1459	21
75.83, 37.95, 40.49, 56.21, 55.31	0.02	0.05	15	0.0186	0.0294	14
2.29, 7.19, 17.67, 19.46, 18.68	0.15	0.25	28	0.1499	0.2396	30
1.60, 138.03, 99.63, 51.02, 62.76	0.06	0.68	98	0.0673	0.6489	104

Table 2. Comparison of convergence performances of the algorithms for the dimension of the problem $d = 10$.

Condition number	10		100		1000		10000	
	epochs	time	epochs	time	epochs	time	epochs	time
eGDM	**38**	**0.01**	**157**	**0.04**	**557**	**0.03**	**1967**	**0.06**
GDX	643	0.05	1139	0.09	6411	0.78	60919	7.46
GDM	6370	0.22	19854	1.11	59487	1.93	1.68E+05	5.51

b and c in (2) vanish under this assumption. If c is nonzero then the function is simply increased in magnitude by c at every point. The shape of the contours does not change. When b is nonzero and H is invertible, the shape of the contours is not changed, but the stationary point of the function moves to $x^* = -H^{-1}b$. Therefore, the objective function (2) is determined solely by the Hessian matrix H. We generate H as follows: $H = QKQ^T$, where Q is a randomly generated orthogonal matrix and K is a diagonal matrix. To generate the orthogonal matrix Q, we use the QR decomposition of a randomly generated square matrix, each of whose elements is chosen from the standard normal distribution. The condition number of H is determined by the diagonal elements of K, which are determined as follows:

$$\begin{aligned} K_{11} &= 1/t' \\ K_{ii} &= (t')^{u_i}, \quad i = 2, \ldots, n-1, \\ K_{nn} &= t' \end{aligned}$$

where t' is the square root of the desired condition number t and each u_i is a uniform variate on the interval $(-1, +1)$ (see e.g. [6]).

The results are summarized for different dimensions of the problem $d = 10, d = 100$, and $d = 1000$, respectively, in Tables 2, 3, and 4. Bold numbers indicate statistically significant performance differences. Comparisons of the algorithms for a quadratic performance function indicate that the gradient descent with momentum algorithm with near optimal learning parameters, eGDM, outperforms GDM and GDX. eGDM has the best performance in all problems compared both in epochs and time (seconds). In particular, the increase in performance is significant when the dimension of the problem d gets larger. Experiments to find out the effect of the eigenvalue distribution of the Hessian on convergence show that the convergence speed of all the algorithms was mainly affected by the condition number of the problem. The distribution of the remaining

Table 3. Comparison of convergence performances of the algorithms for the dimension of the problem $d = 100$.

Condition number	10		100		1000		10000	
	epochs	time	epochs	time	epochs	time	epochs	time
eGDM	**40**	**0.06**	**158**	**0.05**	**544**	**0.06**	**1963**	**0.14**
GDX	802	0.13	1564	0.26	6817	1.15	69838	12.15
GDM	6524	0.39	19395	1.15	58003	3.55	1.73E+05	10.67

Table 4. Comparison of convergence performances of the algorithms for the dimension of the problem $d = 1000$.

Condition number	10		100		1000		10000	
	epochs	time	epochs	time	epochs	time	epochs	time
eGDM	**42**	**9.47**	**164**	**7.03**	**567**	**12.66**	**1950**	**23.97**
GDX	881	16.21	1546	29.51	7631	144.22	74674	1384.30
GDM	6985	67.68	20912	204.15	63781	623.49	1.82E+05	1819.30

eigenvalues also has an effect on the convergence behavior of the algorithm; however, this effect is rather small compared with the effect of the largest eigenvalue κ_1 and the smallest eigenvalue κ_n of the Hessian matrix. The results obtained in this phase of the experiments also support the relations given in (20), (21), and (22).

5. Conclusion

Gradient descent with momentum is a competitive optimization method in regression and classification problems with large and redundant data sets. Step size and momentum factor should be carefully tuned in order to take advantage of the safe, global convergence properties of the gradient descent method. We propose to determine near-optimal step size and momentum factor (19) simultaneously for gradient descent in a stochastic quadratic bowl from the largest and smallest eigenvalue of the Hessian. Numerical results indicate that the gradient descent with near-optimal learning parameters (eGDM) outperforms the simple gradient descent in the case of a quadratic function.

An application of this approach to a popular back-propagation algorithm in neural networks can be found in [7]. In this paper, training time of a multilayer neural network had been reduced significantly in various types of benchmark problems. In general, near-optimal learning parameters (19) can be adapted to any field where one wishes to optimize a performance function by using local quadratic approximation. Now we are working to extend our approach to stochastic optimization problems where the local quadratic approximation of the performance function is performed at every step of the optimization process. In this way, we can use this approach in on-line learning of regression and classification functions.

There are two possible avenues for future research to develop the stochastic version of this approach (eGDM). The first one is to analyze whether the theoretical situation changes for a stochastic quadratic in the realizable vs. nonrealizable case. For instance, we could use (16) in [11] as our stochastic quadratic model and try to apply near-optimal parameters in that setting. A stochastic analysis will be difficult but highly valuable, since on-line gradient descent with momentum is a competitive optimization method in some situations, where the batch version never is.

Secondly, we can use the fact that the largest and smallest eigenvalues of the Hessian can be efficiently estimated empirically to derive a heuristic for step size and momentum factor that can be used when the Hessian is unknown. For this purpose, we can replace the finite difference calculations in [5] with exact Hessian-vector products that can be computed efficiently either analytically [8], by forward-mode automatic differentiation (http://www.autodiff.org/), or even by co-opting complex arithmetic (see Section 2.5 of [13]). Finally, we will evaluate these ideas on standard benchmark datasets, including nonquadratic, nonconvex, and stochastic (on-line) problems.

References

[1] Bhaya A, Kaszkurewicz E. Steepest descent with momentum for quadratic functions is a version of the conjugate gradient method. Neural Networks 2004; 17: 65-71.

[2] Bottou L. Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010, Springer. 2010; pp. 177-186.

[3] Brogan WL. Modern Control Theory (3rd Ed.). Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1991.

[4] Dennis JE Jr, Schnabel RB. Numerical Methods for Unconstrained Optimization and Nonlinear Equations (Classics in Applied Mathematics, 16). Soc for Industrial & Applied Math, 1996.

[5] LeCun Y, Simard PY, Pearlmutter B. Automatic learning rate maximization by on-line estimation of the hessian's eigenvectors. Advances in neural information processing systems 1993; 5: 156-163.

[6] Lenard ML, Minkoff M. Randomly generated test problems for positive definite quadratic programming. Acm T Math Software 1984; 10: 86-96.

[7] Mammadov M, Tas E, Omay RE. Accelerating backpropagation using effective parameters at each step and an experimental evaluation. J Stat Comput Sim 2008; 78: 1055-1064.

[8] Pearlmutter BA. Fast exact multiplication by the hessian. Neural Comput 1994; 6: 147-160.

[9] Polyak BT, Juditsky AB. Acceleration of stochastic approximation by averaging. Siam J Control Optim 1992; 30: 838-855.

[10] Qian N. On the momentum term in gradient descent learning algorithms. Neural networks 1999; 12: 145-151.

[11] Schraudolph NN, Yu J, Günter S. A stochastic quasi-newton method for online convex optimization. In: International Conference on Artificial Intelligence and Statistics. pp. 436-443.

[12] Torii M, Hagan M. Stability of steepest descent with momentum for quadratic functions. Ieee T Neural Networ 2002; 13: 752-756.

[13] Vishwanathan S, Schraudolph NN, Schmidt MW, Murphy KP. Accelerated training of conditional random fields with stochastic gradient methods. In: Proceedings of the 23rd international conference on Machine learning. ACM, pp. 969-976.

[14] Xu W. Towards optimal one pass large scale learning with averaged stochastic gradient descent. CoRR 2011; abs/1107.2490.