ARTICLE IN PRESS

PERGAMON

Neural Networks xx (2003) xxx–xxx

Neural Networks

www.elsevier.com/locate/neunet

Steepest descent with momentum for quadratic functions is a version of the conjugate gradient method

Amit Bhaya*, Eugenius Kaszkurewicz

Department of Electrical Engineering, Federal University of Rio de Janeiro, PEE/COPPE/UFRJ, P.O. Box 68504, Rio de Janeiro, RJ 21945-970, Brazil

Received 20 August 2002; accepted 12 May 2003

Abstract

It is pointed out that the so called momentum method, much used in the neural network literature as an acceleration of the backpropagation method, is a stationary version of the conjugate gradient method. Connections with the continuous optimization method known as heavy ball with friction are also made. In both cases, adaptive (dynamic) choices of the so called learning rate and momentum parameters are obtained using a control Liapunov function analysis of the system.

Keywords: Backpropagation; Steepest descent; Momentum; Conjugate gradient algorithm; Convergence; Continuous optimization; Bilinear system; Control Liapunov function

1. Introduction

The backpropagation with momentum algorithm (BPM) has been much analyzed in the neural network literature and even compared with other methods, in particular, the conjugate gradient (CG) method (Kamarthi & Pittner, 1999; Yu & Chen, 1997). However, although it has been noticed in these two papers that BPM is actually a stationary (or time-invariant) version of the well known CG algorithm, this fact does not seem to have been enunciated explicitly enough in order to have permeated the neural network community, since papers continue to be written on the analysis of BPM without mentioning or exploiting this connection (Qian, 1999; Torii & Hagan, 2002). The purpose of this note is to point out this fact as well as generalize the results of both these papers to the time-varying or dynamic case. In particular, for a quadratic error function, the choice of learning and momentum parameters that has been referred to as 'optimally tuned' is shown to be exactly equivalent to using the CG algorithm.

For brevity, this note will focus on the contributions of Qian (1999) and Torii and Hagan (2002) which are recent and clearly written time-invariant analyses of the BPM method, which has been extensively analyzed, both theoretically and experimentally (see, for example, Hagiwara and Sato (1995), Kamarthi and Pittner (1999), Phansalkar and Sastry (1994), Yu and Chen (1997), and Yu, Chen, and Cheng (1995) and references therein).

In Torii and Hagan (2002), a detailed analysis of the convergence of the BPM method is carried out for the case where a quadratic error function is to be analyzed. In Qian (1999), the BPM algorithm is shown to result from the discretization of a continuous-time ordinary differential equation that models the movement of Newtonian particles through a viscous medium in a conservative force field. These authors do not make connections between their results and the CG algorithm or earlier results on continuous optimization.

This paper points out that the BPM method presented in Torii and Hagan (2002) for fixed momentum and steepest descent parameters (or gains) is actually a special case of the more general CG method, in which both these parameters are chosen dynamically in feedback form. It can be shown, in fact, that a control Liapunov function approach leads, in a straightforward manner, to the CG choice of these parameters (Bhaya & Kaszkurewicz, 2002). In respect of Qian's analogy between Newtonian particles and the BPM method, it is pointed out that there are results on so called continuous optimization, starting as far back as Polyak

[*] Corresponding author. Tel.: +55-21-2562-8080; fax: +55-21-2562-8081.

E-mail addresses: amit@nacad.ufrj.br (A. Bhaya), eugenius@coep.ufrj.br (E. Kaszkurewicz).

0893-6080/03/$ - see front matter © 2003 Published by Elsevier Ltd.
doi:10.1016/S0893-6080(03)00170-9

(1964), Tsypkin (1971), and, more recently, in Attouch, Goudou, and Redont (2000) and Polyak (1987), where this method is referred to as the 'heavy ball with friction' (HBF) method. It turns out that, in fact, both Qian's continuous version of BPM and a continuous version of the CG method can be regarded as the HBF method. However, we point out that it is more fruitful to introduce a continuous version of the CG method that is strictly analogous to the discrete CG algorithm and to regard the learning rate and momentum factor parameters as inputs to this system. When this is done, the resulting continuous-time system is a pair of coupled bilinear systems that permits a simple control Liapunov function analysis, furnishing state-dependent (i.e. feedback) choices of the learning rate and momentum factor that provide global asymptotic stability.

Finally, the control Liapunov function analysis technique used in this paper is very general and can be used to analyze other proposals for tuning parameters in BPM algorithms.

2. Steepest descent plus momentum equals frozen conjugate gradient

In order to fix notation, we describe the BPM problem. Torii and Hagan (2002) studied the problem of determining a set of network weights $\mathbf{x}$ that minimize a quadratic error function

$$f(\mathbf{x}) = \frac{1}{2}\langle \mathbf{x}, \mathbf{A}\mathbf{x}\rangle - \langle \mathbf{b}, \mathbf{x}\rangle + c \tag{1}$$

where $\mathbf{A}$ is a symmetric positive definite matrix. The gradient $\nabla f(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b} =: -\mathbf{r}$ is also called the residue (in the context of solution of the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$). In this notation, the BPM algorithm can be written as

$$\mathbf{x}_{k+1} - \mathbf{x}_k = \mu_k(\mathbf{x}_k - \mathbf{x}_{k-1}) + (1 - \mu_k)\lambda_k \mathbf{r}_k \tag{2}$$

Notice that this equation is the same as that studied by Torii and Hagan (2002), with one important difference: the *learning rate* λ and the *momentum factor* μ are both allowed to be time-varying. Much of the existing analysis in the neural network literature (Haykin, 1999; Phansalkar & Sastry, 1994; Qian, 1999; Torii & Hagan 2002) is restricted to the case of constant λ and μ. There is also a large literature on the time-varying case, generally referred to as dynamic or adaptive choice of the learning rate and momentum factors (see Kamarthi and Pittner (1999) and references therein), but, to the best of our knowledge, the observations made in this paper are new.

We now present the CG method from a control viewpoint, which is the inspiration for the results obtained here. The CG method is conveniently viewed as an acceleration of the steepest descent method, or, equivalently, as an example of the standard feedback control system with a proportional controller. The acceleration is achieved by using a discrete version of a classical control strategy for faster 'closed-loop' response (i.e. acceleration of convergence to the equilibrium): this strategy is known as derivative action in the control (Goodwin, Graebe, & Salgado, 2001). The development of this approach is as follows.

First consider the steepest descent method

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{r}_k \tag{3}$$

Now suppose that a new method is to be derived from the above method by adding a new term that is proportional to a discrete derivative of the state vector $\mathbf{x}_k$. In other words, the new increment $\Delta\mathbf{x}_k := \mathbf{x}_{k+1} - \mathbf{x}_k$ is a linear combination of the steepest descent direction $\mathbf{r}_k$ and the previous increment or discrete derivative of the state $\mathbf{x}_k - \mathbf{x}_{k-1}$. Putting in scalar gains α_k and γ_k, this can be expressed mathematically as follows

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k[\mathbf{r}_k + \gamma_k(\mathbf{x}_k - \mathbf{x}_{k-1})] \tag{4}$$

This can be rewritten as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k, \tag{5}$$

defining $\mathbf{p}_k$ as follows

$$\begin{aligned}\mathbf{p}_k &= \mathbf{r}_k + \gamma_k(\mathbf{x}_k - \mathbf{x}_{k-1}) = \mathbf{r}_k + \gamma_k\alpha_{k-1}\mathbf{p}_{k-1} \\ &= \mathbf{r}_k + \beta_{k-1}\mathbf{p}_{k-1}\end{aligned} \tag{6}$$

where

$$\beta_{k-1} := \gamma_k \alpha_{k-1} \tag{7}$$

Combining these formulas leads to

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{p}_k \\ \mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k \\ \mathbf{p}_{k+1} &= \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k\end{aligned} \tag{8}$$

which are the standard CG formulas (Greenbaum, 1997).

From the point of view of control theory, one approach to understand this algorithm is to think of the 'parameters' α_k and β_k as scalar control inputs. The motivation for doing this is the observation that the systems to be controlled then belong to the class of systems known as *bilinear* in control theory, since they are linear in the control input if the state is held fixed, as well as in the state input, if the control input is fixed. More precisely, taking $\mathbf{r}_k$ and $\mathbf{p}_k$ as the state variables, the heart of the CG algorithm above is the following pair of interconnected bilinear systems

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k \tag{9}$$

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k \tag{10}$$

The control objective is to choose the scalar controls α_k, β_k so as to drive the state vectors $\mathbf{r}_k$ and $\mathbf{p}_k$ to zero. A natural idea is to use feedback, i.e. to choose the controls α_k and β_k as functions of the state vectors. This has been done in Bhaya and Kaszkurewicz (2002), where it is shown that this approach leads to exactly the same algorithm as the conventional CG algorithm. Rather than repeat the analysis here, we will just point out the relation between the optimal choice of the 'controls' α_k, β_k and the learning rate and momentum factors. These choices are

$$\alpha_k = \frac{\langle \mathbf{r}_k, \mathbf{p}_k \rangle}{\langle \mathbf{A}\mathbf{p}_k, \mathbf{p}_k \rangle} \qquad \beta_k = -\frac{\langle \mathbf{p}_k, \mathbf{A}\mathbf{r}_{k+1} \rangle}{\langle \mathbf{p}_k, \mathbf{A}\mathbf{p}_k \rangle} \tag{11}$$

Note that these equations are equivalent to the more commonly used forms (Greenbaum, 1997):

$$\alpha_k = \frac{\langle \mathbf{r}_k, \mathbf{r}_k \rangle}{\langle \mathbf{p}_k, \mathbf{A}\mathbf{p}_k \rangle} \qquad \beta_k = \frac{\langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} \rangle}{\langle \mathbf{r}_k, \mathbf{r}_k \rangle} \tag{12}$$

Comparing Eq. (2) with Eq. (4), the following equivalences between the parameters α_k, β_k of the CG method and λ_k and μ_k of the BPM method are clear

$$\alpha_k = (1-\mu_k)\lambda_k \qquad \alpha_k\gamma_k = \mu_k \tag{13}$$

Using Eq. (7), one can solve for the learning rate and momentum factor in terms of the CG parameters α_k and β_k, for which the optimal choices are well known (Eq. (12)) as follows.

$$\mu_k = \frac{\alpha_k}{\alpha_{k-1}}\beta_{k-1} \qquad \lambda_k = \frac{\alpha_k\alpha_{k-1}}{\alpha_{k-1} - \alpha_k\beta_{k-1}} \tag{14}$$

The discussion above can be summarized in the following theorem.

Theorem 2.1. *Consider the backpropagation method with dynamic learning rate λ_k and dynamic momentum factor μ_k given by*

$$\mathbf{x}_{k+1} - \mathbf{x}_k = \mu_k(\mathbf{x}_k - \mathbf{x}_{k-1}) + (1-\mu_k)\lambda_k\mathbf{r}_k$$

which seeks to minimize a quadratic error function

$$f(\mathbf{x}) = \frac{1}{2}\langle \mathbf{x}, \mathbf{A}\mathbf{x} \rangle - \langle \mathbf{b}, \mathbf{x} \rangle + c$$

where $\mathbf{A}$ is a symmetric positive definite matrix and the gradient $\nabla f(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b} =: -\mathbf{r}$ is also called the residue (in the context of solution of the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$). Consider also the CG method, parameterized by α_k and β_k, and given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{p}_k$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k\mathbf{A}\mathbf{p}_k$$

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k\mathbf{p}_k$$

where

$$\alpha_k = \frac{\langle \mathbf{r}_k, \mathbf{r}_k \rangle}{\langle \mathbf{p}_k, \mathbf{A}\mathbf{p}_k \rangle}$$

$$\beta_k = \frac{\langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} \rangle}{\langle \mathbf{r}_k, \mathbf{r}_k \rangle}$$

Then, with the learning rate and momentum factor chosen dynamically as follows

$$\mu_k = \frac{\alpha_k}{\alpha_{k-1}}\beta_{k-1}$$

$$\lambda_k = \frac{\alpha_k\alpha_{k-1}}{\alpha_{k-1} - \alpha_k\beta_{k-1}}$$

and an appropriate choice of initial conditions, the two methods produce the same set of vectors $\mathbf{x}_k$, $\mathbf{r}_k$. With this choice of parameters λ_k and μ_k, the BPM method is said to be optimally tuned, where optimality refers to the maximum reduction in 2-norm of the vectors $\mathbf{r}_k$ and $\mathbf{p}_k$ *at each iteration.*

Eqs. (12) and (14) show that the optimal learning rate and momentum factor can be calculated in terms of the state variables $(\mathbf{r}, \mathbf{p})$, although this involves calculation of more inner products than the CG method. It may be possible to derive simpler formulas for μ_k and λ_k, but we suggest that it is easier and safer just to use the standard CG method, rather than the equivalent 'optimally tuned' BPM algorithm. In addition, the CG method also has numerous tried and tested variants, both linear and nonlinear (Nocedal & Wright, 1999).

Finally, note that the formula for μ_k was obtained in Yu and Chen (1997) where it was referred to as the optimal momentum factor, however, the corresponding formula for λ_k was not derived.

2.1. Practical implications for tuning of parameters

Although the focus of this paper is on theory and, specifically, the connections between the BPM and CG algorithms, a few words on practical implementations are in order.

There are literally hundreds of proposals, some totally heuristic and some based on convergence analysis, for acceleration of the backpropagation algorithm using learning rate and momentum factors defined in different ways in each proposal. One of the points we emphasize is that the control Liapunov function approach that we use in this paper is very general and can therefore also be used to analyze other proposals in a systematic manner. To be more specific, the parameters that are 'tuned' (such as learning rate and momentum factor, or some combination of these) in any specific proposal should be regarded as 'control inputs' which are to be chosen so as to make some appropriately chosen Liapunov function negative definite (i.e. make it into a control Liapunov function). The difficulty lies in the choice of the Liapunov function, as is always the case. However, as shown above, for the ‘vanilla’ BPM algorithm, the straightforward choice of 2-norms and weighted 2-norms work as control Liapunov functions and lead to the optimal choices of parameters, in the sense of maximal reduction of the residual norm in each iteration.

Of course, there are many practical issues involved in determining ultimate performance and the complexity and cost of each iteration of an optimal algorithm may offset its faster rate of convergence. For a discussion of some of these practical computational issues, we refer the reader to Kamarthi and Pittner (1999) and Yu and Chen (1997) (and references therein).

3. Continuous optimization, BPM and the conjugate gradient algorithm

Early papers on ‘continuous iterative methods’ and ‘analog computing’ (see, e.g. Polyak (1964), Rybashov (1969) and Tsypkin (1971)) proposed the use of dynamical systems to compute the solution of various optimization problems. Specifically, Polyak (1964) investigates the idea of using a dynamical system that represents a HBF, moving under Newtonian dynamics in a conservative force field. Later, a more detailed analysis of this system was carried out by Attouch et al. (2000). In order to avoid the repetitive use of the phrase ‘continuous version of X’ or ‘X dynamical system’, the abbreviation ‘X ODE’ will be used below (where X can be BPM, HBF, steepest descent, or CG).

Specifically, the HBF ODE is

$$\ddot{\mathbf{x}}(t) + \theta\dot{\mathbf{x}}(t) + \nabla\Phi(\mathbf{x}(t)) = \mathbf{0} \tag{15}$$

This is to be compared with the steepest descent ODE

$$\dot{\mathbf{x}}(t) = -\epsilon\nabla\Phi(\mathbf{x}(t))$$

As pointed out by Attouch et al. (2000), the damping term $\theta\dot{\mathbf{x}}(t)$ confers optimizing properties in Eq. (15), but it is isotropic and ignores the geometry of Φ. The second derivative term $\ddot{\mathbf{x}}(t)$, which induces inertial effects, is a singular perturbation or regularization of the classical continuous Newton ODE, which may be written as follows

$$\nabla^2\Phi(\mathbf{x}(t))\dot{\mathbf{x}}(t) + \nabla\Phi(\mathbf{x}(t)) = \mathbf{0}$$

In the neural network context, $\mathbf{x}$ is the weight vector, usually denoted $\mathbf{w}$, and the potential energy function Φ is the error function usually denoted $E(\mathbf{w})$ (as in Qian (1999)). In fact, with these changes of notation, it is clear that the HBF Eq. (15) is exactly the equation proposed by Qian (1999) as the continuous analog of BPM. This is no surprise, since the physical model underlying Qian’s model (point mass moving in a viscous medium with friction under the influence of a conservative force field and with Newtonian dynamics) is the same as HBF.

Thus, the continuous version of BPM is the HBF ODE and may be regarded either as a regularization of the steepest descent ODE or the classical Newton ODE.

Qian (1999) proposes the Liapunov function

$$E_T = \frac{1}{2}\langle\dot{\mathbf{x}}, \dot{\mathbf{x}}\rangle + \Phi(\mathbf{x}) \tag{16}$$

which has time derivative along the trajectories of Eq. (15) given by

$$\frac{\mathrm{d}E_T}{\mathrm{d}t} = \langle\ddot{\mathbf{x}}, \dot{\mathbf{x}}\rangle + \langle\nabla\Phi, \dot{\mathbf{x}}\rangle = -\langle\dot{\mathbf{x}}, \dot{\mathbf{x}}\rangle \le 0. \tag{17}$$

In view of the fact that the standard state space representation of the second order (vector) ODE (Eq. (15)) would involve both the position ($\mathbf{x}$) and the velocity ($\dot{\mathbf{x}}$), it is clear that the Liapunov function (16) has a time derivative along the trajectories of Eq. (15) that is only negative semidefinite. Thus, a conclusion of local asymptotic stability would require the use of LaSalle’s principle or equivalent.

The other observation worth making is that the CG algorithm allows for α_k and β_k (adaptive or dynamic choice of parameters), whereas most existing analyses of BPM assume that the learning and momentum parameters are constant. In the context of the HBF ODE, allowing time-varying parameters amounts to allowing the parameter θ in Eq. (15) vary with time. More generally, introducing two time-varying coefficients into Eq. (15) gives

$$\eta(t)\ddot{\mathbf{x}}(t) + \theta(t)\dot{\mathbf{x}}(t) + \nabla\Phi(\mathbf{x}(t)) = \mathbf{0}. \tag{18}$$

where $\eta(t)$ and $\theta(t)$ are nonnegative time-varying parameters to be chosen adequately in order that the trajectories of Eq. (18) converge to an equilibrium (that is a minimum of the energy function). Choosing the obvious modification of Eq. (16)

$$E_T = \frac{1}{2}\eta(t)\langle\dot{\mathbf{x}}, \dot{\mathbf{x}}\rangle + \Phi(\mathbf{x}) \tag{19}$$

the time derivative along the trajectories of Eq. (15) given by

$$\begin{aligned}\frac{\mathrm{d}E_T}{\mathrm{d}t} &= \frac{1}{2}\dot{\eta}(t)\langle\dot{\mathbf{x}}, \dot{\mathbf{x}}\rangle + \eta(t)\langle\ddot{\mathbf{x}}, \dot{\mathbf{x}}\rangle + \langle\nabla\Phi, \dot{\mathbf{x}}\rangle \\ &= \left(-\theta(t) + \frac{1}{2}\dot{\eta}(t)\right)\langle\dot{\mathbf{x}}, \dot{\mathbf{x}}\rangle \le 0 \end{aligned} \tag{20}$$

Once again, this is a negative semidefinite time derivative, provided that $\theta(t) > \frac{1}{2}\dot{\eta}(t)$, $\forall t$; however, any conclusions about asymptotic stability now require more work and additional assumptions, since LaSalle’s theorem is not applicable in the time-varying case.

Section 3.1 explores this connection in more detail and then proposes what we regard as a more fruitful continuous-time version of the discrete CG iteration.

3.1. Analysis of conjugate gradient flow

In this subsection, the HBF ODE is specialized to the case of a quadratic potential function and the resulting ODE is referred to as the CG ODE. In fact, there are many different ways to write down a continuous version of the discrete CG iteration. One natural approach is to write down continuous versions of Eqs. (9) and (10) as follows

$$\dot{\mathbf{r}} = -\alpha\mathbf{A}\mathbf{p} \tag{21}$$

$$\dot{\mathbf{p}} = \mathbf{r} - \beta\mathbf{p} \tag{22}$$

Elimination of the vector $\mathbf{p}$ yields the vector second order CG ODE:

$$\ddot{\mathbf{r}} + \beta\dot{\mathbf{r}} + \alpha\mathbf{A}\mathbf{r} = \mathbf{0} \tag{23}$$

Observe that Eq. (23) is a version of Eq. (15), since for the CG algorithm the ‘potential function’ being minimized is $\Phi = \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} - \mathbf{b}^{\mathrm{T}}\mathbf{x}$, for which $\nabla\Phi(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b} =: -\mathbf{r}$, so that the HBF ODE becomes (in $\mathbf{x}$-coordinates) $\ddot{\mathbf{x}} + \theta + \boldsymbol{\theta}\mathbf{x}\cdot\dot{\mathbf{x}} - \mathbf{r} = \mathbf{0}$. Multiplying through by $\mathbf{A}$ and observing that $\ddot{\mathbf{r}} = -\mathbf{A}\ddot{\mathbf{x}}$, $\dot{\mathbf{r}} = -\mathbf{A}\dot{\mathbf{x}}$, the latter becomes $\ddot{\mathbf{r}} + \theta\dot{\mathbf{r}} + \mathbf{A}\mathbf{r} = \mathbf{0}$, which is Eq. (23) with $\theta = \beta$, $\alpha = 1$. Thus, one can think of Eq. (23) as an equation embodying the HBF method, where the parameters β (friction coefficient) and α (related to the spring constant) need to be chosen in order to make the trajectories of Eq. (23) tend to zero asymptotically.

3.1.1. Analysis of constant α and β

The constant parameter case is easily dealt with using classical results of Rayleigh: a recent approach can be found in Datta and Rincon (1993, Lemma 2.1, Theorem 3.1), restated here for easy reference.

Lemma 3.1 (Datta and Rincon, 1993). *Let λ, $\mathbf{x}$ be an eigenvalue, eigenvector pair of the quadratic eigenvalue problem*:

$$(\lambda^2\mathbf{M} + \lambda\mathbf{D} + \mathbf{K})\mathbf{x} = \mathbf{0} \tag{24}$$

Suppose that $\mathbf{M} = \mathbf{M}^{\mathrm{T}} > \mathbf{0}$, $\mathbf{K} = \mathbf{K}^{\mathrm{T}} \geq \mathbf{0}$ and $\mathbf{D} = \mathbf{D}^{\mathrm{T}}$. Then

$$\mathrm{Re}(\lambda) = -\frac{|\lambda|^2\mathbf{D}_{\mathbf{x}}}{|\lambda|^2\mathbf{M}_{\mathbf{x}} + \mathbf{K}_{\mathbf{x}}} \tag{25}$$

where $\mathbf{S}_{\mathbf{x}}$ denotes $\mathbf{x}^H\mathbf{S}\mathbf{x}$.

Theorem 3.2 (Datta and Rincon, 1993). *Each eigenvalue λ of* Eq. (24) *satisfies the inequality*

$$|\lambda| \leq \frac{\rho(\mathbf{D})}{2\lambda_{\min}(\mathbf{M})} + \sqrt{\left(\frac{\rho(\mathbf{D})}{2\lambda_{\min}(\mathbf{M})}\right)^2 + \frac{\lambda_{\max}(\mathbf{K})}{\lambda_{\min}(\mathbf{M})}} \tag{26}$$

For Eq. (23), the matrices are $\mathbf{M} = \mathbf{I}$, $\mathbf{D} = \beta\mathbf{I}$ and $\mathbf{K} = \alpha\mathbf{A}$, leading to the results

$$\mathrm{Re}(\lambda) = -\frac{|\lambda|^2\beta}{|\lambda|^2 + \alpha(\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}/\mathbf{x}^{\mathrm{T}}\mathbf{x})} \leq -\frac{|\lambda|^2\beta}{|\lambda|^2 + \alpha\lambda_{\max}(\mathbf{A})} < 0 \tag{27}$$

proving that the zero solution of Eq. (23) is asymptotically stable. Given that Eq. (23) represents a linear system, it can be concluded that the stability is actually exponential. Note also that Theorem 3.2 provides the following estimate for the moduli of the eigenvalues of the second order pencil

$$|\lambda| \leq \frac{\beta}{2} + \sqrt{\frac{\beta^2}{4} + \alpha\lambda_{\max}(\mathbf{A})} \tag{28}$$

Eqs. (27) and (28) show how the choice of the parameters α and β affect the dynamics of convergence, without the need for modal decompositions used by Qian (1999).

3.1.2. Analysis of α and β chosen dynamically

As pointed out in Bhaya and Kaszkurewicz (2002), it is natural to consider the discrete CG iteration (Eqs. (9) and (10)) as a pair of coupled bilinear systems as the starting point in the derivation of the parameters α and β, regarded as control inputs. This approach will be repeated in the analysis of Eqs. (21) and (22), rather than simply analyzing stability properties of the second order vector ODE (Eq. (23)) with variable parameters α and β. A control Liapunov argument similar to that in Bhaya and Kaszkurewicz (2002) is used (also see Quinn (1980) and Ryan and Buckingham (1983)). Consider the Liapunov function candidate

$$V(\mathbf{r}, \mathbf{p}) = \frac{1}{2}\langle\mathbf{r}, \mathbf{A}^{-1}\mathbf{r}\rangle + \frac{1}{2}\langle\mathbf{p}, \mathbf{A}\mathbf{p}\rangle \tag{29}$$

Then

$$\dot{V} = \langle\dot{\mathbf{r}}, \mathbf{A}^{-1}\mathbf{r}\rangle + \langle\dot{\mathbf{p}}, \mathbf{A}\mathbf{p}\rangle = \langle -\alpha\mathbf{A}\mathbf{p}, \mathbf{A}^{-1}\mathbf{r}\rangle + \langle\mathbf{r} - \beta\mathbf{p}, \mathbf{A}\mathbf{p}\rangle$$
$$= -\alpha\langle\mathbf{p}, \mathbf{r}\rangle + \langle\mathbf{r}, \mathbf{A}\mathbf{p}\rangle - \beta\langle\mathbf{p}, \mathbf{A}\mathbf{p}\rangle$$

whence it follows that appropriate choices of α and β that make $\dot{V}$ negative semidefinite are as follows.

$$\text{if } \langle\mathbf{r}, \mathbf{p}\rangle \neq 0, \quad \alpha = \frac{\langle\mathbf{r}, \mathbf{A}\mathbf{p}\rangle}{\langle\mathbf{r}, \mathbf{p}\rangle}, \quad \beta > 0 \tag{30}$$

$$\text{if } \langle\mathbf{r}, \mathbf{p}\rangle = 0; \quad \beta \text{ such that } \langle\mathbf{r}, \mathbf{A}\mathbf{p}\rangle - \beta\langle\mathbf{p}, \mathbf{A}\mathbf{p}\rangle < 0 \tag{31}$$

Since β is positive and $\langle\mathbf{p}, \mathbf{A}\mathbf{p}\rangle$ is positive, it follows that the choice of β in Eq. (31) depends on the sign of $\langle\mathbf{r}, \mathbf{A}\mathbf{p}\rangle$: if this inner product is positive, then $\beta > \langle\mathbf{r}, \mathbf{A}\mathbf{p}\rangle/\langle\mathbf{p}, \mathbf{A}\mathbf{p}\rangle$; if it is negative or zero, any positive choice of β will do. Since $\dot{V}$ is only semidefinite, and α and β are functions of the state variables $\mathbf{r}$, $\mathbf{p}$, LaSalle’s theorem can be applied. It states that the trajectories of the CG flow (Eqs. (21) and (22)) will approach the maximal invariant set $\mathcal{M}$ in the set

$$\mathcal{G} := \{(\mathbf{r},\mathbf{p}) : \dot{V} = 0\} \tag{32}$$

Invariance of $\mathcal{M} \subset \mathcal{G}$ means that any trajectory of the controlled system starting in $\mathcal{M}$ remains in $\mathcal{M}$ for all t.

Given the choices of α and β in Eqs. (30) and (31), observe that $\dot{V} = 0$ can only occur if Eq. (30) occurs, implying that $\dot{V} = -\beta\langle \mathbf{p}, \mathbf{Ap}\rangle$, which, in turn, is zero if and only if $\mathbf{p} = \mathbf{0}$, so that $\mathcal{G}$ can be alternatively characterized as $\{\mathbf{p} = \mathbf{0}\}$. From Eq. (21), $\mathbf{p} = \mathbf{0} \Rightarrow \dot{\mathbf{r}} = \mathbf{0}$, which, in turn, implies that $\mathbf{r}$ is constant. From Eq. (22), $\mathbf{p} = \mathbf{0}$ implies that $\dot{\mathbf{p}} = \mathbf{r}$. Since $\mathbf{r} = c$ (constant), this means that c must be zero (otherwise $\mathbf{p} = \mathbf{0}$ could not occur). Global asymptotic stability of the origin now follows from LaSalle's theorem.

The paragraphs above have proved the following theorem.

Theorem 3.3. *Given the symmetric, positive definite matrix* $\mathbf{A}$, *and a quadratic function* $\Phi = \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{Ax} - \mathbf{b}^{\mathrm{T}}\mathbf{x}$, *the trajectories of the CG dynamical system, dependent on the positive parameters* α and β, *defined as*

$$\dot{\mathbf{r}} = \alpha\mathbf{Ap},$$

$$\dot{\mathbf{p}} = \mathbf{r} - \beta\mathbf{p}$$

converge globally to the minimum of the quadratic function Φ (*i.e. to the solution of the linear system* $\mathbf{Ax} = \mathbf{b}$) *if the parameters* α *and* β *are chosen as follows*

$$\text{if } \langle \mathbf{r},\mathbf{p}\rangle \neq 0, \qquad \alpha = \frac{\langle \mathbf{r},\mathbf{Ap}\rangle}{\langle \mathbf{r},\mathbf{p}\rangle}, \quad \beta > 0$$

$$\text{if } \langle \mathbf{r},\mathbf{p}\rangle = 0; \quad \beta \text{ such that } \langle \mathbf{r},\mathbf{Ap}\rangle - \beta\langle \mathbf{p},\mathbf{Ap}\rangle < 0$$

where $\mathbf{r} := \mathbf{b} - \mathbf{Ax}$. *The parameter* β *is chosen as follows: if the inner product* $\langle \mathbf{r},\mathbf{Ap}\rangle$ *is positive, then* $\beta > \langle \mathbf{r},\mathbf{Ap}\rangle / \langle \mathbf{p},\mathbf{Ap}\rangle$; *if it is negative or zero, any positive choice of* β *will do.*

Remarks. Note that the Liapunov function could be chosen as

$$V(\mathbf{r},\mathbf{p}) = \frac{1}{2}\langle \mathbf{r},\mathbf{A}^{-1}\mathbf{r}\rangle + \frac{1}{2}\langle \mathbf{p},\mathbf{Qp}\rangle,$$

where $\mathbf{Q}$ is any positive definite matrix. In particular, the choice $\mathbf{Q} = \mathbf{I}$ results in the simple choice of parameters

$$\alpha = 1, \qquad \beta > 0$$

demonstrating that the continuous version of the CG method can even utilize constant parameters, as opposed to the discrete CG method, where the 'parameters' α and β must be chosen as functions of the state vectors $\mathbf{r}$ and $\mathbf{p}$. Notice, however, that they are chosen either as constants or in state feedback form, rather than as some arbitrary functions of time that must be chosen so as to stabilize Eqs. (21) and (22). This makes it possible to use LaSalle's theorem to obtain a global asymptotic stability result.

It should also be noticed that the choices of α and β given in Theorem 3.3 correspond to the choices made in the discrete CG iteration.

A choice of initial conditions consistent with the discrete CG iteration is: $\mathbf{r}_0 = \mathbf{b} - \mathbf{Ax}_0$ and $\mathbf{p}_0 = \mathbf{r}_0$.

4. Conclusions

This paper establishes various connections between the CG algorithm and the backpropagation algorithm with momentum acceleration. In particular, a general control Liapunov function approach to the analysis as well as design of BPM type algorithms is proposed. A continuous version of the CG algorithm is put forward and it is shown how to choose the parameters (that correspond to the learning rate and momentum factor) in state feedback form in order to guarantee global asymptotic stability of this system, implying convergence of the error to zero. This result is new and should prove to be of value in understanding the theoretical properties of backpropagation type algorithms; specifically in understanding the role of the learning rate and momentum factor in modifying convergence properties. In conclusion, we quote a paragraph from Alber (1971) that is as relevant today (with some minor changes in the buzzwords) as when it was written 30 years ago: "The increasing interest in continuous-descent methods is due firstly to the fact that tools for the numerical solution of systems of ordinary differential equations are now well developed and can thus be used in conjunction with computers; secondly, continuous methods can be used on analog computers (= neural networks); thirdly, theorems concerning the convergence of these methods and theorems concerning the existence of solutions of equations and of minimum points of functionals are formulated under weaker assumptions than is the case for the analogous discrete processes" (Parentheses ours). Similar justification for the consideration of continuous versions of well known discrete-time algorithms can be found in Chu (1988) and Chu (1992).

Acknowledgements

This research was partially supported by the Brazilian Government agencies CAPES, CNPq, FINEP and FAPERJ. Amit Bhaya would like to thank Prof. H. Attouch for an interesting discussion on his paper Attouch et al. (2000) that led to the development of Section 3 of this paper.

References

Alber, Y. I. (1971). Continuous processes of the Newton type. *Differential Equations*, *7*(11), 1461–1471.

Attouch, H., Goudou, X., & Redont, P. (2000). The heavy ball with friction method, I. The continuous dynamical system: global exploration of the global minima of a real-valued function by asymptotic analysis of a dissipative dynamical system. *Communications in Contemporary Mathematics*, *2*(1), 1–34.

Bhaya, A., & Kaszkurewicz, E (2002). *Iterative methods as dynamical systems with feedback control*. Technical report NACAD/BK/01-02, Department of Electrical Engineering, Federal University of Rio de Janeiro, COPPE/UFRJ.

Chu, M. T. (1988). On the continuous realization of iterative processes. *SIAM Review*, *30*(3), 375–387.

Chu, M. T. (1992). Matrix differential equations: a continuous realization process for linear algebra problems. *Nonlinear Analysis Theory, Methods and Applications*, *18*(12), 1125–1146.

Datta, B. N., & Rincon, F. (1993). Feedback stabilization of a second-order system: a nonmodal approach. *Linear Algebra and its Applications*, *188*, 135–161.

Goodwin, G. C., Graebe, S. F., & Salgado, M. E. (2001). *Control system design*. Upper Saddle River, NJ: Prentice-Hall.

Greenbaum, A. (1997). *Iterative methods for solving linear systems*. Philadelphia: SIAM.

Hagiwara, M., & Sato, A. (1995). Analysis of momentum term in back-propagation. *IEICE Transactions on Information Systems*, *E78-D*(8), 1080–1086.

Haykin, S. (1999). *Neural networks* (2nd ed.). Upper Saddle River, NJ: Prentice-Hall.

Kamarthi, S. V., & Pittner, S. (1999). Accelerating neural network training using weight extrapolations. *Neural Networks*, *12*(9), 1285–1299.

Nocedal, J., & Wright, S. J. (1999). *Numerical optimization*. *Springer series in operations research*, New York: Springer.

Phansalkar, V. V., & Sastry, P. S. (1994). Analysis of the back-propagation algorithm with momentum. *IEEE Transactions on Neural Networks*, *5*(3), 505–506.

Polyak, B. T. (1964). Some methods of speeding up the convergence of iterative methods. *USSR Computational Mathematics and Mathematical Physics*, *4*(5), 1–17.

Polyak, B. T. (1987). *Introduction to optimization*. New York: Optimization Software.

Qian, N. (1999). On the momentum term in gradient descent learning algorithms. *Neural Networks*, *12*(1), 145–151.

Quinn, J. P. (1980). Stabilization of bilinear systems by quadratic feedback controls. *Journal of Mathematical Analysis and its Applications*, *75*(1), 66–80.

Ryan, E. P., & Buckingham, N. J. (1983). On asymptotically stabilizing feedback control of bilinear systems. *IEEE Transactions on Automatic Control*, *AC-28*(8), 863–864.

Rybashov, M. V. (1969). Method of differential equations in the problem of finding the extremum of a function using analog computers. *Automation and Remote Control*, *30*(5), 181–194.

Torii, M., & Hagan, M. T. (2002). Stability of steepest descent with momentum for quadratic functions. *IEEE Transactions on Neural Networks*, *13*(3), 752–756.

Tsypkin, Y. Z. (1971). *Adaptation and learning in automatic systems* (*Vol. 73*). *Mathematics in science and engineering*, New York: Academic Press, first published in Russian under the title *Adaptatsia i obuchenie v avtomaticheskikh sistemakh*, 1968, Moscow: Nauka.

Yu, X.-H., & Chen, G.-A. (1997). Efficient backpropagation learning using optimal learning rate and momentum. *Neural Networks*, *10*(3), 517–527.

Yu, X.-H., Chen, G.-A., & Cheng, S.-X. (1995). Dynamic learning rate optimization of the backpropagation algorithm. *IEEE Transactions on Neural Networks*, *6*(3), 669–677.