Neurocomputing 144 (2014) 493–500

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Improvement of the convergence speed of a discrete-time recurrent neural network for quadratic optimization with general linear constraints

María José Pérez-Ilzarbe *

Departamento de Automática y Computación, Universidad Pública de Navarra, Campus de Arrosadía s/n, 31006 Pamplona, Spain

ARTICLE INFO

Article history:
Received 6 February 2014
Received in revised form
30 April 2014
Accepted 2 May 2014
Communicated by S. Hochreiter
Available online 22 May 2014

Keywords:
Discrete-time neural networks
Recurrent neural networks
Quadratic optimization
Constrained optimization
Conditioning

ABSTRACT

In this work a specific preconditioning technique is developed to improve the convergence speed of a discrete-time recurrent neural network for quadratic optimization with general linear constraints. The discrete-time network is a model recently published with the broadest range of applicability to various optimization problems and constraints. The proposed preconditioning technique is shown to improve the convergence speed of the model significantly, and thus contribute to enhance the application of the model in these problems. In addition to the theoretical analysis, extensive experimental results are presented to illustrate the technique developed, and to show the significant improvement attained.

1. Introduction

Constrained quadratic optimization has numerous scientific and engineering applications, e.g. systems identification, control systems, signal and image processing, and many others. The use of neural models is a powerful approach to solve optimization problems mainly because such models are simpler than classical optimization techniques and amenable for parallel realization. Many neural networks have been proposed for quadratic optimization [1–16]. Most of them are continuous-time dynamical systems [10–16]. The interest on developing discrete-time neural networks for optimization has been increasing in the last few years [1–8], due to the advantages they have over continuous-time models in digital implementations for real-time applications. In addition, they are specifically designed to be easily implemented in hardware. For the discrete networks, however, it is comparatively more difficult to secure stability and good convergence properties, therefore improvements in this sense are crucial.

The models published in [1,2,5] and [9] are discrete-time networks that are able to solve the quadratic optimization problem with linear constraints. This problem consists on minimizing a quadratic function in the space defined by a set of bound constraints, linear inequality constraints and linear equality constraints. The equality restrictions can be easily eliminated with a simple procedure proposed in [10]. Using it, the problem of quadratic optimization with general linear constraints can be formulated as

$$\text{minimize} \quad E=\frac{1}{2}x^T Mx+g^T x,$$

$$x\in\Omega, \Omega=\{x\in R^n | a_d\le Ax\le a_u, d\le x\le u\}, \tag{1}$$

where $x\in R^n$ is the vector of optimization variables, M is a symmetric nxn matrix, $g\in R^n$, A is a mxn matrix that defines the inequality constraints, $a_d, a_u\in R^m$ are vectors that contain the lower and upper limits of the inequality constraints, and $d, u\in R^n$ are vectors that contain the lower and upper bound constraints.

Convergence to the optimal solution of (1) is assured with the neural networks developed in [1,2,5] and [9], provided that certain conditions hold, thus these models have different capabilities. With respect to the conditions that the matrix M must fulfill, the systems published in [2] and [5] need a matrix M positive definite, whereas the ones in [1] and [9] can work with M positive semidefinite. With respect to the constraints, the models proposed in [5] and [9] can only work with a number of restrictions (bounds plus inequality) not higher than the number of variables. Only the models developed in [1] and [2] are able to work with any number of linear constraints of any type.

Thus, the neural network proposed in [1] is, up to our knowledge, the model with the broadest range of applicability since it can solve the convex quadratic problems with general linear constraints, whereas the one in [2] can only solve the strictly

* Tel.: +34 948169267.
E-mail address: mjperez@unavarra.es

http://dx.doi.org/10.1016/j.neucom.2014.05.015
0925-2312/

convex case. As was shown in [2], the handicap of the model developed in [1] is its extreme slowness in some optimization problems, and the goal of the present work is to improve the convergence speed of such model.

On the other hand, it is known that the performance of discrete-time algorithms is influenced by the numerical characteristics of the particular mathematical formulation of the problem. In this respect, the models proposed in [2] and [9] take advantage of a preconditioning procedure that reformulates the original optimization problem in the appropriate way to improve their convergence speed. Here, the numerical characteristics of the algorithm published in [1] are analyzed, and a preconditioning procedure is developed specifically to accelerate it.

The work starts with an analysis of the neural network for the strictly convex case with only bound constraints. Note that, if the number of restrictions is lower or equal than the number of variables, the total set of restrictions can be transformed onto a set of bound constraints. For this case, it has been developed a conditioning procedure which is optimal in the sense that it maximizes a lower bound of the convergence rate of the network. Based on this result, a conditioning procedure has been devised for the not strictly convex case with number of constraints greater than the number of optimization variables. The method has been tested with a number of examples having very different characteristics. The experimental results obtained constitute a strong evidence of the goodness of the proposed conditioning technique, since a significant improvement of network behavior has been found systematically.

2. Neural network model and conditioning procedure

In [1] a discrete-time neural network was proposed to solve the variational inequality problem consisting on finding a vector $x^* \in \Omega$ such that:

$$(Mx^*+g)^T(x-x^*) \geq 0, \quad \forall x \in \Omega \tag{2}$$

where Ω is a region defined by linear constraints as in (1). It is well known that, if M is symmetric, then problem (2) is equivalent to problem (1). The neural model proposed in [1] is

$$y(k+1) = y(k) + h(\hat{N}+\hat{M})^T f(\hat{N}y(k) - \hat{M}y(k) - \hat{g}) - \hat{N}y(k) \tag{3}$$

where $y^T=(x,\nu)^T$, ν are the m Lagrange multipliers associated with the inequality constraints in (1), f implements the bounds $[d,u]$x $[a_d,a_u]$, h is a positive constant, and:

$$\hat{M} = \begin{pmatrix} M & -A^T \\ O & I \end{pmatrix}; \quad \hat{N} = \begin{pmatrix} I & O \\ A & O \end{pmatrix}; \quad \hat{g} = \begin{pmatrix} g \\ 0 \end{pmatrix}$$

This neural network can solve problem (1), with M positive definite or semidefinite. The following results about the convergence of (3) to the solution of (1) can be found in [1]:

- If $\hat{N}^T\hat{M}$ is positive definite, then model (3) is exponentially convergent if $0 < h \leq \min(2/\eta_{\max}, 1/(2\pi_{\min}))$, where $\eta_{\max}$ is the maximum eigenvalue of $(\hat{N}+\hat{M})(\hat{N}+\hat{M})^T$ and $\pi_{\min}$ is the minimum eigenvalue of $\hat{N}^T\hat{M}$. The convergence rate has a lower bound $\sigma(h) = -\ln(r(h))$, where $r(h) = 1-2\,h\pi_{\min}$.
- If $\hat{N}^T\hat{M}$ is positive semidefinite, then model (3) is globally convergent if $0 < h \leq 1/\eta_{\max}$.

From here, in this work a theoretical analysis has been made for the case of $\hat{N}^T\hat{M}$ positive definite, with the aim of finding a way to maximize the lower bound of the convergence rate of the network. Based on the results obtained, a preconditioning technique for model (3) has been developed valid for both cases, $\hat{N}^T\hat{M}$ positive definite and semidefinite.

2.1. Case of $\hat{N}^T\hat{M}$ positive definite

2.1.1. New theoretical results

Result 1. The exponential convergence condition $0 < h \leq \min(2/\eta_{\max}, 1/(2\pi_{\min}))$ is equivalent to $0 < h \leq 2/(1+\lambda_{\max})^2$, where $\lambda_{\max}$ is the maximum eigenvalue of M.

Proof. First, it is easy to realize that $\hat{N}^T\hat{M}$ contains the matrix M in its nxn upper left corner, and zeros in the rest of its elements. So, the condition $\hat{N}^T\hat{M} > 0$ implies that there are no inequality constraints and M is positive definite: $\hat{N}^T = I$ and $\hat{N}^T\hat{M} = \hat{M} = M > 0$. Thus $(\hat{N}+\hat{M})(\hat{N}+\hat{M})^T = (I+M)^2$ and, consequently, $\eta_{max} = (1+\lambda_{\max})^2$ and $\pi_{min} = \lambda_{\min}$, where $\lambda_{\min}$ is the minimum eigenvalue of M. On the other hand, we have $(1+\lambda_{\max})^2 \geq (1+\lambda_{\min})^2 \geq 4\lambda_{\min} = 4\pi_{min}$, from which we obtain $2/\eta_{\max} = 2/(1+\lambda_{\max})^2 \leq 1/(2\pi_{\min})$ and, consequently, $\min(2/\eta_{\max}, 1/(2\pi_{\min})) = 2/(1+\lambda_{\max})^2$. □

Result 2. The lower bound of the convergence rate, $\sigma(h) = -\ln(r(h))$ where $r(h)) = 1-2\,h\pi_{\min}$, is maximum when $h = h_{opt} = 2/(1+\lambda_{\max})^2$, and is

$$\sigma_{opt} = -\ln(\text{ropt}) \text{ with } r_{opt} = 1 - \frac{4\lambda_{\min}}{(1+c_M\lambda_{\min})^2}, \tag{4}$$

where c_M is the condition number of M: $c_M = \lambda_{\max}/\lambda_{\min}$.

Proof. It is obvious that $\sigma = -\ln(r)$ is maximum when r is minimum, and $r = 1-2\,h\pi_{\min}$ is lower the higher is h, so the optimal value of this constant is the maximum one that assures convergence: $h_{opt} = 2/(1+\lambda_{\max})^2$. With it, taking into account that $\pi_{\min} = \lambda_{\min}$, we easily obtain (4). □

Result 3. The conditioning procedure that maximizes the lower bound of the convergence rate σ_{opt} is the one that makes c_M as low as possible and $\lambda_{\max} = 1$. The corresponding maximal value $\sigma_{opt}^{\max}$ is

$$\sigma_{opt}^{\max} = -\ln\left(r_{opt}^{\min}\right) \text{ with } r_{opt}^{\min} = 1 - 1/c_M \tag{5}$$

Proof. To maximize the lower bound of the convergence rate σ_{opt} we must minimize r_{opt}. Looking at (4) we can see that r_{opt} depends on c_M and $\lambda_{\min}$. In order to analyze the dependence of r_{opt} with $\lambda_{\min}$ the first and second derivatives are made:

$$\frac{dr_{opt}}{d\lambda_{\min}} = \frac{4c_M^2\lambda_{\min}^2 - 4}{(1+c_M\lambda_{\min})^2} = \frac{4(c_M\lambda_{\min}-1)}{(c_M\lambda_{\min}+1)} = 0 \Rightarrow \lambda_{\min} = 1/c_M$$

$$\Rightarrow \lambda_{\max} = 1; \quad \frac{d^2r_{opt}}{d\lambda_{\min}^2} = \frac{8c_M}{(c_M\lambda_{\min}+1)^2} > 0$$

where the fact that $\lambda_{\min} > 0$ and $c_M > 0$ has been used. So, for a given value c_M of the condition number, the corresponding r_{opt} takes its minimum value when $\lambda_{\min} = 1/c_M$, i.e., $\lambda_{\max} = 1$. Substituting it in (4) we obtain (5). On the other hand, it is obvious that for a fixed value of $\lambda_{\min}$ the lowest possible value of r_{opt} corresponds to the lowest possible value of c_M. □

2.1.2. Preconditioning procedure

From the analysis made in Section 2.1.1 we have concluded that, to maximize σ_{opt}, the condition number c_M should be made as low as possible. To decrease c_M the technique developed in [11] will be used. In [11] several matrix conditioners P were developed to obtain a matrix $H' = PHP$ better conditioned than H. Optimal diagonal conditioners were proven to be those with elements $p_{ii} = K/\sqrt{h_{ii}}$ with $K > 0$. This conditioning has been used successfully in non-linear neural systems such as [2,9,12] and [13]. At this point, it is interesting to remark that the values of the convergence rates obtained in [2,9,12] and [13] for the corresponding networks only depend on the condition number of the corresponding