Engineering Applications of Artificial Intelligence 32 (2014) 54–62

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

A neural network model for solving convex quadratic programming problems with some applications

Alireza Nazemi*

Department of Mathematics, School of Mathematical Sciences, Shahrood University, P.O. Box 3619995161-316, Shahrood, Iran

ARTICLE INFO

Article history:
Received 29 August 2013
Received in revised form
10 January 2014
Accepted 25 February 2014
Available online 25 March 2014

Keywords:
Neural network
Quadratic programming
Optimality conditions
Dynamic model
Convergent
Stability

ABSTRACT

This paper presents a capable neural network for solving strictly convex quadratic programming (SCQP) problems with general linear constraints. The proposed neural network model is stable in the sense of Lyapunov and can converge to an exact optimal solution of the original problem. A block diagram of the proposed model is also given. Several applicable examples further show the correctness of the results and the good performance of the model.

1. Introduction

The quadratic program arises in a wide variety of scientific and engineering applications including regression analysis, function approximation, signal processing, image restoration, parameter estimation, filter design, and robot control (Agrawal and Fabien, 1999; Avriel, 1976; Bazaraa et al., 1993; Bertsekas, 1989; Boyd and Vandenberghe, 2004; Fletcher, 1981; More and Toroaldo, 1991). In many real-time applications these optimization problems have a time-varying nature, they have to be solved in real time. One promising approach for handling these optimization problems with high dimensions and dense structure is to employ an artificial neural network based on circuit implementation. The main idea of the neural network approach for optimization is to construct a nonnegative energy function and establish a dynamic system that represents an artificial neural network. The dynamic system is usually in the form of first order ordinary differential equations. Furthermore, it is expected that the dynamic system will approach its static state (or an equilibrium point), which corresponds to the solution for the underlying optimization problem, starting from an initial point. The main advantage of neural network approach to optimization is that the nature of the dynamic solution procedure is inherently parallel and distributed. Therefore, the neural network approach can solve optimization problems in running time at the orders of magnitude much faster than the most popular optimization algorithms executed on general purpose digital computers. In addition, neural networks for solving optimization problems are hardware-implementable; that is, the neural networks can be implemented by using integrated circuits.

The neural network for solving mathematical programming problems was first proposed by Tank and Hopfield (1986). Since then, neural networks for solving different kinds of quadratic programming problems have been rather extensively studied and some important results have also been obtained. For example one can see (Ai et al., 2006; Ding and Huang, 2008; Effati and Nazemi, 2006; Effati and Ranjbar, 2011; Gao and Liao, 2006, 2010; Hu, 2009; Hu and Wang, 2007; Huang, 2002; Jiang et al., 2009; Leung et al., 2001; Lv et al., 2010; Maa and Shanblatt, 1992; Nazemi, 2011; Tao et al., 2001; Xia, 1996; Xia and Wang, 1999, 2000, 2004a; Xia et al., 2004; Xia and Feng, 2003, 2005; Xue and Bian, 2007, 2009; Ding and Huang, 2008; Zhang and Constantinides, 1992; Zhang et al., 2011; Wu et al., 1996) where several neural network models for solving convex quadratic programming, degenerate quadratic programming, convex quadratic bilevel programming and convex quadratic minimax problems have been proposed. All these mentioned literatures can be classified into two categories. One is functional transformation, which involves a mapping of an inequality constraint to an equality, and a penalty is constructed to penalize the inequality constraint violation. The other is an approach that converts the inequality constraints into the equality constraints by adding slack or surplus variables. Both approaches, however, deal with inequality constraints indirectly. The neural

* Tel./fax: +98 273 3300235.
E-mail address: nazemi20042003@yahoo.com

http://dx.doi.org/10.1016/j.engappai.2014.02.014
0952-1976/

network models formulated on the basis of them are rather complicated and difficult to be realized in the form of hardware. Thus, on the basis of the above analysis, proposing an efficient neural network for solving the SCQP problems with a simple structure, good stability and convergence results is very necessary and meaningful.

By modifying the multipliers associated with inequality constraints, we can directly solve the convex quadratic programming problem without nonnegative constraints of the multipliers associated with inequality constraints. Hence it is no longer necessary to convert the inequality constraints into the equality constraints by using the slack variables, and consequently complexity and difficulty of computation can be reduced. Utilizing this technique, in this paper a neural network model for solving the SCQP problem with ease of computation and desirable stability results is exhibited. Based on the saddle point theorem, the equilibrium point of the proposed neural network is proved to be equivalent to the solution of the Karush Kuhn Tucker (KKT) conditions of the SCQP problem. The existence and uniqueness of an equilibrium point of the presented neural network model are also analyzed. By constructing a suitable Lyapunov function, a sufficient condition to ensure global stability in the sense of Lyapunov of the unique equilibrium point of the proposed model is precisely studied.

This paper is organized as follows. Section 2 describes the system model and gives some necessary preliminaries. Section 3 discusses the stability of the equilibrium point and the convergence of the optimal solution. Section 4 provides several numerical examples to demonstrate the validity of the obtained results. Some conclusions are drawn in Section 5.

2. A neural network model

We are concerned with a SCQP problem of the following form:

$$\min \quad \tfrac{1}{2}x^TQx+D^Tx \tag{1}$$

subject to

$$Ax-b\le 0, \tag{2}$$

$$Ex-f=0, \tag{3}$$

where $Q\in\mathbb{R}^{n\times n}$ is a symmetric positive definite matrix, $A\in\mathbb{R}^{m\times n}$, $b\in\mathbb{R}^m$, $E\in\mathbb{R}^{l\times n}$, $f\in\mathbb{R}^l$, $x\in\mathbb{R}^n$ and $\mathrm{rank}(A,E)=m+l$.

For the convenience of later discussions, it is necessary to introduce a few notations. Throughout this paper, $\mathbb{R}^n$ denotes the space of n-dimensional real column vectors and ^{T} denotes the transpose. In what follows, $\|\cdot\|$ denotes the l^2-norm of $\mathbb{R}^n$ and $x=(x_1,x_2,\ldots,x_n)^T$. For any differentiable function $f:\mathbb{R}^n\to\mathbb{R}$, $\nabla f(x)\in\mathbb{R}^n$ means the gradient of f at x and $\nabla^2 f(x)\in\mathbb{R}^{n\times n}$ means the Hessian matrix of $f(x)$. For any differentiable mapping $F=(F_1,\ldots,F_m)^T:\mathbb{R}^n\to\mathbb{R}^m$, $\nabla F=[\nabla F_1(x),\ldots,\nabla F_m(x)]\in\mathbb{R}^{n\times m}$ denotes the transposed Jacobian of F at x. If $A\in\mathbb{R}^{m\times n}$, then the ith row of A is denoted by $A_{i.}$ and the jth column of A is denoted by $A_{.j}$.

Consider the Lagrange function of (1)–(3) similar to Huang (2002) as

$$L(x,u,v)=\frac{1}{2}x^TQx+D^Tx+\frac{1}{2}\sum_{k=1}^{m}u_k^2(a_{k.}x-b_k)+\sum_{p=1}^{l}v_p(e_{p.}x-f_p). \tag{4}$$

According to the KKT conditions for problem (1)–(3) in Bazaraa et al. (1993), $x^*\in\mathbb{R}^n$ is an optimal solution of (1)–(3) if and only if there exist $u^*\in\mathbb{R}^m$ and $v^*\in\mathbb{R}^l$ such that $(x^{*T},u^{*T},v^{*T})^T$ satisfies

$$\begin{cases} u^*\ge 0,\ Ax^*-b\le 0,\ u^{*T}(Ax^*-b)=0,\\ Qx^*+D+A^Tu^*+E^Tv^*=0,\\ Ex^*-f=0. \end{cases} \tag{5}$$

x^* is called a KKT point of (1)–(3) and a pair $(u^{*T},v^{*T})^T$ is called the Lagrangian multiplier vector corresponding to x^*. From Bazaraa et al. (1993) we see that x^* is an optimal solution of (1)–(3), if and only if x^* is a KKT point of (1)–(3).

Now, let $x(\cdot)$, $u(\cdot)$ and $v(\cdot)$ be some time dependent variables. In order to use a neural network method to solve the quadratic optimization problem (1)–(3), a neural network system have to be constructed and make the steady points of the neural network system to satisfy the KKT conditions (5). Therefore, our aim now is to design a neural network that will settle down to the saddle point of $L(x,u,v)$. We may describe the neural network model corresponding to (1)–(3) and its dual by the following nonlinear dynamical system:

$$\frac{dx}{dt}=-\nabla_xL(x,u,v)=-\left(Qx+D+\frac{1}{2}\sum_{k=1}^{m}u_k^2a_{k.}^T+\sum_{p=1}^{l}v_pe_{p.}^T\right), \tag{6}$$

$$\frac{du}{dt}=\nabla_uL(x,u,v)=\mathrm{diag}(u_1,\ldots,u_m)(Ax-b), \tag{7}$$

$$\frac{dv}{dt}=\nabla_vL(x,u,v)=Ex-f, \tag{8}$$

with an initial point $(x(t_0)^T,u(t_0)^T,v(t_0)^T)^T$ and $u(t_0)>0$. Note that in the neural network (6)–(8) the multipliers corresponding to the inequality constraints are defined as u_k^2 $(k=1,\ldots,m)$. The advantage of modifying the multipliers associated with inequality constraints is that the nonnegative constraint can be eliminated, thus the relevant optimality conditions (5) in convex programming theory can be transplanted here in their original forms.

To simplify the discussion, we denote $y=(x^T,u^T,v^T)^T\in\mathbb{R}^{n+m+l}$, D^* as the optimal point set of (1)–(3) and its dual, and

$$F(y)=\begin{bmatrix} -(Qx+D+\frac{1}{2}A^Tu^2+E^Tv)\\ \mathrm{diag}(u_1,\ldots,u_m)(Ax-b)\\ Ex-f \end{bmatrix}.$$

Thus neural network (6)–(8) can be written as

$$\frac{dy}{dt}=\eta F(y), \tag{9}$$

$$y(t_0)=y_0,\quad u(t_0)>0, \tag{10}$$

where η is a scale parameter and indicates the convergence rate of the neural network (9) and (10). For simplicity of our analysis, we let $\eta=1$. An indication on how the neural network (9) and (10) can be implemented on hardware is provided in Fig. 1.

To see how well the present neural network model (9) and (10) is, we compare it with some existing models for solving (1)–(3). First, let us consider

$$\min \quad \tfrac{1}{2}x^TQx+D^Tx \tag{11}$$

subject to

$$Ax-b\le 0, \tag{12}$$

$$x\ge 0, \tag{13}$$

where $Q\in\mathbb{R}^{n\times n}$ is a symmetric positive definite matrix, $A\in\mathbb{R}^{m\times n}$, $b\in\mathbb{R}^m$. In Xia (1996) a primal-dual neural network model for (11)–(13) was developed as

$$\frac{d}{dt}\begin{bmatrix}x\\u\end{bmatrix}=\begin{bmatrix}Q+I_n & A^T\\ -A & I_m\end{bmatrix}\times\begin{bmatrix}-x+(x-(Qx+D)+A^Tu)^+\\ u-(u-Ax+b)^+\end{bmatrix}, \tag{14}$$

where I_n and I_m are $n\times n$ and $m\times m$ identity matrices, respectively. The proposed neural network model for solving (11)–(13) is then given by