Available online at www.sciencedirect.com
ScienceDirect
Mathematics and Computers in Simulation 101 (2014) 103–112
MATHEMATICS AND COMPUTERS IN SIMULATION
www.elsevier.com/locate/matcom

Original Article

A new neural network for solving quadratic programming problems with equality and inequality constraints[☆]

Yongqing Yang [a,*], Jinde Cao [b,c], Xianyun Xu [a], Manfeng Hu [a], Yun Gao [a]

[a] *School of Science, Jiangnan University, Wuxi 214122, PR China*
[b] *Department of Mathematics, Southeast University, Nanjing 210096, PR China*
[c] *Department of Mathematics, Faculty of Science, King Abdulaziz University, Jeddah 21589, Saudi Arabia*

Received 14 March 2011; received in revised form 10 April 2012; accepted 28 February 2014
Available online 25 March 2014

Abstract

A new neural network is proposed in this paper for solving quadratic programming problems with equality and inequality constraints. Comparing with the existing neural networks for solving such problems, the proposed neural network has fewer neurons and an one-layer architecture. The proposed neural network is proven to be global convergence. Furthermore, illustrative examples are given to show the effectiveness of the proposed neural network.
© 2014 IMACS. Published by Elsevier B.V. All rights reserved.

Keywords: Neural network; Convergence; Stability; Quadratic programming; Positive semidefinite

1. Introduction

In the past decades, neural networks for optimization have been widely investigated. Hopfield and Tank first proposed the neural network for linear programming problems [11,20]. Their work has inspired many researchers to investigate other neural network models for solving programming problems. Kennedy and Chua [14] presented a neural network for solving nonlinear programming problems. It is known that the neural network model contains finite penalty parameters and generates approximate solutions only. To avoid using penalty parameters, many other methods have been proposed in recent years, see [28,23,7,21,26,2].

Quadratic programming problems arises in a broad variety of scientific and engineering applications, including regression analysis, signal processing, image restoration, robot control, etc. Also, some nonlinear optimization problems

☆ This work was jointly supported by the National Natural Science Foundation of China under Grant 60875036, the Key Research Foundation of Science and Technology of the Ministry of Education of China under Grant 108067, and Program for Innovative Research Team of Jiangnan University.

* Corresponding author. Tel.: +86 051085910532.
E-mail addresses: yyq640613@gmail.com, yongqingyang@163.com (Y. Yang), jdcao@seu.edu.cn (J. Cao).

http://dx.doi.org/10.1016/j.matcom.2014.02.006
0378-4754/© 2014 IMACS. Published by Elsevier B.V. All rights reserved.

are usually approximated by a quadratic programming problem. In general, the quadratic programming problem can be written the following form

$$\begin{aligned} &\text{minimize} && \frac{1}{2}x^T Wx + c^T x \\ &\text{subject to} && Ax = b \\ &&& x \geq 0. \end{aligned}$$

Many neural networks for the above problem have been proposed. For example, some primal-dual neural networks were presented [24,22,3,4]. In order to simplify the architecture of the dual neural network, a simplified dual neural network was introduced [15,12]. Using the projection theorem, several projection neural networks were developed to solve quadratic programming problems [8,27,10,18,13,5,16,17,29,9], which were globally convergent to exact optimal solutions. In [16,17], Liu and Wang proposed some one-layer recurrent neural networks for solving quadratic programming problems. The one-layer recurrent neural networks have more simple architecture complexity than the other neural networks such as Lagrangian network and projection network.

Up to now, only a few neural networks have been developed to solve the following quadratic programming problems with equality and inequality constraints.

$$\begin{aligned} &\text{minimize} && \frac{1}{2}x^T Wx + c^T x \\ &\text{subject to} && Ax = b \\ &&& Bx \leq d \end{aligned} \tag{1}$$

where $x = (x_1, x_2, \ldots, x_n)^T \in R^n$, W is an $n \times n$ real symmetric positive semidefinite matrix, $c \in R^n$, $A \in R^{m\times n}$, $rank(A) = m\ (0 < m < n)$, $B \in R^{p\times n}$, $b \in R^m$, $d \in R^p$.

Zhang and Wang [30] and Xia et al. [25] treated equality and one-sided inequality constraints as special cases of two-sided bound constraints. It need sufficiently large numbers to represent down bound $-\infty$ or up bound $+\infty$, which is disadvantageous for the design of neural network. Effati and Nazemi [6] proposed a neural network for solving this problem from the Karash–Kuhn–Tucker (KKT) conditions. However, this neural network has more neurons and requires stronger convergence conditions.

Motivated by the above discussions, the aim of this paper is to develop a new neural network for solving (1). It has fewer state variables, the lower structure complexity and weaker convergence conditions.

The rest of this paper is organized as follows. In Section 2, a new neural network model is proposed. In Section 3, the global convergence of the proposed neural network is investigated. In Section 4, several illustrate examples are given to show the effectiveness of the proposed neural network. Conclusions are found in Section 5.

2. The neural network model

In this section, we will give the neural network model for solving problem (1).

Several neural network models based on the KKT conditions of the convex optimization problems have been proposed for solving quadratic programming problems with equality and inequality constraints [6,25,30]. These neural network models need m state variables to satisfy equality constraints (5) in the KKT conditions. To simply the network architecture, we give the following Lemma.

Lemma 1. *x^* is an optimal solution of (1) if and only if there exist $y^* \geq 0$ such that $(x^*, y^*)^T$ satisfies:*

$$(I - P)(Wx^* + c + B^T y^*) + Q(Ax^* - b) = 0 \tag{2}$$

$$(y^* + Bx^* - d)^+ - y^* = 0 \tag{3}$$

where $P = A^T(AA^T)^{-1}A$, $Q = A^T(AA^T)^{-1}$, and $(y)^+ = ([y_1]^+, \ldots, [y_n]^+)^T$, $[y_i]^+ = \max\{0, y_i\}$.

Download English Version:

https://daneshyari.com/en/article/7543403

Download Persian Version:

https://daneshyari.com/article/7543403

Daneshyari.com