A New Nonlinear Neural Network for Solving QP Problems

Yinhui Yan(✉)

Shenzhen Airlines Co., Ltd., Shenzhen, China
A15601@shenzhenair.com

Abstract. In this paper, a new nonlinear neural network is proposed to solving quadratic programming problems subject to linear equality and inequality constraints without any parameter tuning. This nonlinear neural network is proved to be stable in the sense of Lyapunov under certain conditions. Simulation results are further presented to show the effectiveness and performance of this neural network.

Keywords: Nonlinear neural network · Lyapunov stability · Quadratic programming

1 Introduction

Quadratic programming (QP) studies problem of optimizing (minimizing or maximizing) a quadratic function of several variables subject to linear equality or linear inequality constraints on these variables. It has been successfully applied to various fields [1] such as transportation, energy, telecommunications, and manufacturing. Traditional approaches to solve QP problems [2–8] include interior point method, active set method, augmented Lagrangian method, conjugate gradient method and gradient projection method etc. However traditional methods usually require much computational time and can not meet real-time requirements in practical applications.

In 1986, based on a gradient method, Hopfield and Tank [9] in their paper proposed a new approach to solve LP problem by using recurrent neural network. The main advantage of this method is that it can be implemented by using analog electronic circuits, possibly on a VLSI (very large-scale integration) circuit, which can operate in parallel. In contrast with traditional approaches which may involve an iterative process and require long computational time, this model can potentially provide an optimal solution in real time. After their pioneer work [9,10], numerous neural network models have been developed to solve optimization problems, such as the Lagrangian neural network [11], the deterministic annealing neural network [12], the projection neural network [13], the delayed projection neural network [14], the dual neural network [15,16] and the primal-dual neural network [17]. In 1988, Kennedy and Kan [18] developed a neural network for solving nonlinear programming problems based on Karush-Kuhn-Tucker (KKT) optimal conditions. By using a penalty parameter its solution

Z. Zeng et al. (Eds.): ISNN 2014, LNCS 8866, pp. 347–357, 2014.
DOI: 10.1007/978-3-319-12436-0_39

usually approximates the optimal solution. Only when the penalty parameter is very large, it is same as the exact solution. Later Maa and Shanblatt [19] extended this penalty based method by using two-phase model and ensured that the model converges to the optimal solution. However, their model is more complex and still requires careful parameter selection. To overcome these drawbacks, Xia [20] proposed a primal and dual model to solving this problem. Zhang and Constantinides [11] invented a lagrangian neural network based on the idea of lagrangian multiplier. In this model slack variables are introduced as new variables to deal with inequality constraints, this may lead to high dimension thus require more computation. Unlike previous approaches using a fixed parameter, Wang etc. [12] used a time-variant temperature to design a deterministic annealing neural network to resolve the linear programs. In International Symposium on Mathematical Programming 2000, Nguyen [21] presented a novel recurrent neural network model to solve linear optimization problem. Compared with Xia's model, Nguyen's model not only retains the advantages of Xia's model but also have a more intuitive economic interpretation and much faster convergence. The most interested thing for this model is its nonlinear dynamic structure and high convergence speed. This paper will extend the Nguyen's neural network model to solving quadratic programming problems. For the background and details of neural networks, we refer to [22–31].

The rest of this paper is organized as follows: Section 2 presents a nonlinear neural network to solving quadratic problem and the convergence property of this neural network. Section 3 studies the stability of the proposed dynamical neural network and proves that this neural network is stable in the sense of Lyapunov under certain conditions. Section 4 demonstrates the power and effectiveness of the proposed neural network. In the end, Section 5 gives a summary of this paper and points out some future research directions.

2 Model Description

Consider the QP Problem

$$\begin{array}{ll} \text{Find } \mathbf{x} \text{ which minimizes}: & \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{e}^T\mathbf{x}, \\ \text{subject to} & \mathbf{D}\mathbf{x} = \mathbf{b}, \\ & \mathbf{A}\mathbf{x} \geq \mathbf{c}, \\ & \mathbf{x} \geq 0, \end{array} \tag{1}$$

where $\mathbf{x}$ and $\mathbf{e}$ are n-dimensional vectors, $\mathbf{Q}$ is an $n \times n$ symmetric positive definite matrix, $\mathbf{D} \in \mathbb{R}^{p\times n}$, $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^{p\times 1}$, $\mathbf{c} \in \mathbb{R}^{m\times 1}$. We call this problem as the primal QP problem.

The lagrangian function of this minimization problem can be written as

$$\mathcal{L}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{e}^T\mathbf{x} - \mathbf{y}^T(\mathbf{D}\mathbf{x} - \mathbf{b}) - \mathbf{z}^T(\mathbf{A}\mathbf{x} - \mathbf{c}), \tag{2}$$

where $\mathbf{z} \in \mathbb{R}^p_+ = \{\mathbf{z} \in \mathbb{R}^p | \mathbf{z} \geq 0\}$, $\mathbf{y} \in \mathbb{R}^m$ are Lagrangian multipliers. According to the Karush-Kuhn-Tucker (KKT) conditions [32,33], $\mathbf{x}^\star$ is a solution of (1) if and only if there exist $\mathbf{y}^\star \in \mathbb{R}^m$, $\mathbf{z}^\star \in \mathbb{R}_+^p$ so that $(\mathbf{x}^\star, \mathbf{y}^\star, \mathbf{z}^\star)$ satisfies the following conditions:

$$\begin{aligned} \mathbf{Q}\mathbf{x}^\star + \mathbf{e} - \mathbf{D}^T\mathbf{y}^\star - \mathbf{A}^T\mathbf{z}^* &\geq 0, \\ \mathbf{x}^{\star T}\left(\mathbf{Q}\mathbf{x}^\star + \mathbf{e} - \mathbf{D}^T\mathbf{y}^\star - \mathbf{A}^T\mathbf{z}^*\right) &= 0, \\ \mathbf{b} - \mathbf{D}\mathbf{x}^\star &= 0, \\ \mathbf{c} - \mathbf{A}\mathbf{x}^\star &\leq 0, \\ \mathbf{z}^{\star T}\left(\mathbf{c} - \mathbf{A}\mathbf{x}^\star\right) &= 0. \end{aligned} \tag{3}$$

We propose a recurrent neural network for solving the primal and dual problem as follows:

$$\dot{\mathbf{x}} = -\mathbf{Q}(\mathbf{x} + k\dot{\mathbf{x}}) - \mathbf{e} + \mathbf{D}^T(\mathbf{y} + k\dot{\mathbf{y}}) + \mathbf{A}^T(\mathbf{z} + k\dot{\mathbf{z}}), \mathbf{x} \geq 0, \tag{4a}$$

$$\dot{\mathbf{y}} = \mathbf{b} - \mathbf{D}(\mathbf{x} + k\dot{\mathbf{x}}), \tag{4b}$$

$$\dot{\mathbf{z}} = -\mathbf{A}(\mathbf{x} + k\dot{\mathbf{x}}) + \mathbf{c}, \mathbf{z} \geq 0, \tag{4c}$$

where k is a positive constant. The architecture of the proposed neural network model is shown in Fig. 1. The proposed neural network consists of two layers of

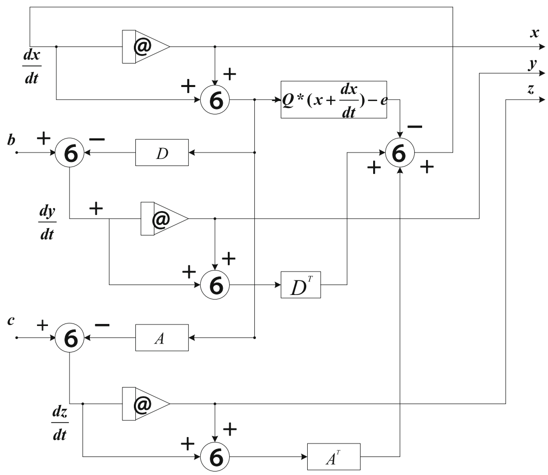

Fig. 1. Block diagram of the neural network (4a, 4b, and 4c)

neurons, i.e., primal neurons and dual neurons. The outputs from one layer are the inputs to the other layer. The inputs of the primal neurons are composed of the dual neuron's outputs and their derivatives, while the inputs of the dual neurons are composed of the primal neuron's outputs and their derivatives. Due to the involvement of these derivatives, this neural network model is a nonlinear dynamic system. The convergence property of the system is stated by the following theorem.

Theorem 1: If the neural network whose dynamics guided by the differential equations (4a, 4b, and 4c) converges to a steady state $(\mathbf{x}^\star, \mathbf{y}^\star, \mathbf{z}^\star)$, then $\mathbf{x}^\star$ will be the optimal solution of the primal QP problem and the Lagrangian multipliers $\mathbf{y}^\star$ and $\mathbf{z}^\star$ the optimal solution of the dual of the QP problem.

Proof. Let $\mathbf{x}_i$ be the ith component of $\mathbf{x}$, then the equation (4a) can be written as

$$\frac{d\mathbf{x}_i}{dt} = \{(-\mathbf{Q}(\mathbf{x} + k\frac{d\mathbf{x}}{dt}) - \mathbf{e}) + \mathbf{D}^T(\mathbf{y} + \frac{d\mathbf{y}}{dt}) + \mathbf{A}^T(\mathbf{z} + k\frac{d\mathbf{z}}{dt})\}_i \text{ if } \mathbf{x}_i > 0, \quad (5)$$

$$\frac{d\mathbf{x}_i}{dt} = \max\{\{(-\mathbf{Q}(\mathbf{x}+k\frac{d\mathbf{x}}{dt})-\mathbf{e})+\mathbf{D}^T(\mathbf{y}+\frac{d\mathbf{y}}{dt})+\mathbf{A}^T(\mathbf{z}+k\frac{d\mathbf{z}}{dt})\}_i, 0\} \text{ if } \mathbf{x}_i = 0. \quad (6)$$

Note that (6) is to ensure that $\mathbf{x}$ will bounded from below by 0. Let $\mathbf{x}^\star$, $\mathbf{y}^\star$ and $\mathbf{z}^\star$ be the limit of $\mathbf{x}(t)$, $\mathbf{y}(t)$ and $\mathbf{z}(t)$ respectively. In other words

$$\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}^\star \quad (7)$$

$$\lim_{t\to\infty} \mathbf{y}(t) = \mathbf{y}^\star \quad (8)$$

$$\lim_{t\to\infty} \mathbf{z}(t) = \mathbf{z}^\star \quad (9)$$

By the definition of convergence, we have $\frac{d\mathbf{x}^\star}{dt} = 0$, $\frac{d\mathbf{y}^\star}{dt} = 0$ and $\frac{d\mathbf{z}^\star}{dt} = 0$. From Eqns. (5) and (6) we conclude that

$$0 = \{(-\mathbf{Q}(\mathbf{x}^\star) - \mathbf{e} + \mathbf{D}^T\mathbf{y}^\star + \mathbf{A}^T\mathbf{z}^\star)\}_i \quad \text{if} \quad \mathbf{x}_i^\star > 0 \quad (10)$$

$$0 = \max\{\{(-\mathbf{Q}(\mathbf{x}^\star) - \mathbf{e} + \mathbf{D}^T\mathbf{y}^\star + \mathbf{A}^T\mathbf{z}^\star)\}_i, 0\} \quad \text{if} \quad \mathbf{x}_i^\star = 0 \quad (11)$$

In other words:

$$\left(-\mathbf{Q}(\mathbf{x}^\star) - \mathbf{e} + \mathbf{D}^T\mathbf{y}^\star + \mathbf{A}^T\mathbf{z}^\star\right)_i \leq 0 \quad (12)$$

$$\mathbf{x}_i^\star\left(-\mathbf{Q}(\mathbf{x}^\star) - \mathbf{e} + \mathbf{D}^T\mathrm{y}^\star + \mathbf{A}^T\mathbf{z}^\star\right)_i = 0 \quad (13)$$

or

$$\left(\mathbf{Q}(\mathbf{x}^\star) + \mathbf{e} - \mathbf{D}^T\mathbf{y}^\star - \mathbf{A}^T\mathbf{z}^\star\right) \geq 0 \quad (14)$$

$$\mathbf{x}^{\star T}\left(-\mathbf{Q}(\mathbf{x}^\star) - \mathbf{e} + \mathbf{D}^T\mathbf{y}^\star + \mathbf{A}^T\mathbf{z}^\star\right) = 0 \quad (15)$$

Similarly, from Eqns. (4b) and (4c), we have:

$$\mathbf{D}\mathbf{x}^\star - \mathbf{b} = 0 \quad (16)$$

$$\mathbf{A}\mathbf{x}^\star - \mathbf{c} \geq 0 \quad (17)$$

$$\mathbf{z}^{\star T}(\mathbf{A}\mathbf{x}^\star - \mathbf{c}) = 0 \quad (18)$$

By KKT conditions in (3) and conditions provided in (15-18) we have shown that $x^\star$ and $(y^\star, z^\star)$ are the optimal solutions for the problem (1) and its dual problem respectively. This concludes the proof.

3 Stability Analysis

It's easy to prove that the differential equations (4a), (4b) and (4c) are equivalent to the following second order differential equations:

$$\begin{aligned}(\mathbf{I} + k\mathbf{Q} + k^2\mathbf{D}^T\mathbf{D} + k^2\mathbf{A}^T\mathbf{A})\ddot{\mathbf{x}} &+ \\ (\mathbf{Q} + 2k\mathbf{D}^T\mathbf{D} + 2k\mathbf{A}^T\mathbf{A})\dot{\mathbf{x}} &+ \\ (\mathbf{D}^T\mathbf{D} + \mathbf{A}^T\mathbf{A})\mathbf{x} - (\mathbf{D}^T\mathbf{b} + \mathbf{A}^T\mathbf{c}) &= 0. \end{aligned} \tag{19}$$

Suppose the entity $\mathbf{D}^T\mathbf{D} + \mathbf{A}^T\mathbf{A}$ is non-singular, we introduce a transformation $\mathbf{x} = \mathbf{u} + (\mathbf{D}^T\mathbf{D} + \mathbf{A}^T\mathbf{A})^{-1}(\mathbf{D}^T\mathbf{b} + \mathbf{A}^T\mathbf{c})$, then we have $\dot{\mathbf{x}} = \dot{\mathbf{u}}$ and $\ddot{\mathbf{x}} = \ddot{\mathbf{u}}$. By this transformation, the ordinary differential equation (19) can be written as

$$\begin{aligned}(\mathbf{I} + k\mathbf{Q} + k^2\mathbf{D}^T\mathbf{D} + k^2\mathbf{A}^T\mathbf{A})\ddot{\mathbf{u}} + (\mathbf{Q} + 2k\mathbf{D}^T\mathbf{D} + 2k\mathbf{A}^T\mathbf{A})\dot{\mathbf{u}} \\ +(\mathbf{D}^T\mathbf{D} + \mathbf{A}^T\mathbf{A})\mathbf{u} = 0\end{aligned} \tag{20}$$

Now we would like to study the stability of the equation (20).

Generally we study the stability of the following second order ordinary differential equation

$$\mathbf{L}\ddot{\mathbf{u}} + \mathbf{M}\dot{\mathbf{u}} + \mathbf{N}\mathbf{u} = 0. \tag{21}$$

where $\mathbf{L}$, $\mathbf{M}$ and $\mathbf{N}$ are all positive definite.

First we consider the simplified second order ordinary differential equation

$$\ddot{\mathbf{u}} + \mathbf{M}\dot{\mathbf{u}} + \mathbf{N}\mathbf{u} = 0. \tag{22}$$

where $\mathbf{M}$ and $\mathbf{N}$ are both positive definite.

Theorem 2: If the coefficient matrices $\mathbf{M}$ and $\mathbf{N}$ of the system (22) are both positive definite, then this dynamic system is global asymptotic stable.

Proof. If we set $\mathbf{u}_1 = \mathbf{u}$, $\mathbf{u}_2 = \dot{\mathbf{u}}$, we have the system

$$\begin{cases} \dot{\mathbf{u}}_1 = \mathbf{u}_2, \\ \dot{\mathbf{u}}_2 = -\mathbf{M}\mathbf{u}_2 - \mathbf{N}\mathbf{u}_1. \end{cases}$$

In order to show the global asymptotic the stability of (22), we only need to show the real parts of the eigenvalues of $\mathbf{P}$ are negative, where $\mathbf{P} = \begin{pmatrix} 0 & \mathbf{I} \\ -\mathbf{N} & -\mathbf{M} \end{pmatrix}$. Suppose $\lambda \in \mathbb{C}^n$ be an eigenvalue of $\mathbf{P}$ with the corresponding non-zero eigenvector $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2)$, by the definition of eigenvector, we have

$$\begin{pmatrix} 0 & \mathbf{I} \\ -\mathbf{N} & -\mathbf{M} \end{pmatrix} \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{v}_2 \\ -\mathbf{N}\mathbf{v}_1 - \mathbf{M}\mathbf{v}_2 \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix}.$$

Since $\mathbf{N}$ is positive definite, $\mathbf{P}$ is non-singular. This concludes that λ can not be an eigenvalue of $\mathbf{P}$. Since $\lambda \neq 0$ and $\mathbf{v}_2 = \lambda\mathbf{v}_1$, we claim that $\mathbf{v}_1 \neq 0$ and

$\mathbf{v}_2 \neq 0$. Without loss of generality we may assume that $\mathbf{v}_1^* \cdot \mathbf{v}_1 = 1$, where $*$ denotes complex conjugate transpose. Using this assumption, we can write $\lambda^2 = \mathbf{v}_1^* \lambda^2 \mathbf{v}_1 = \mathbf{v}_1^* \lambda \mathbf{v}_2 = \mathbf{v}_1^*(-\mathbf{N}\mathbf{v}_1 - \mathbf{M}\mathbf{v}_2) = -\mathbf{v}_1^*\mathbf{N}\mathbf{v}_1 - \lambda \mathbf{v}_1^* \mathbf{M}\mathbf{v}_1$, where we have used the identity $\lambda \mathbf{v}_1 = \mathbf{v}_2$ and $\lambda \mathbf{v}_2 = -\mathbf{N}\mathbf{v}_1 - \mathbf{M}\mathbf{v}_2$. Since $\mathbf{N}$ is positive definite, the entity $\beta = \mathbf{v}_1^*\mathbf{N}\mathbf{v}_1$ is positive real. Similarly, the entity $\alpha = \mathbf{v}_1^*\mathbf{M}\mathbf{v}_1$ is positive because of the positive definiteness of $\mathbf{M}$. Substitute these scalars into the equation $\lambda^2 = -\mathbf{v}_1^*\mathbf{N}\mathbf{v}_1 - \lambda\mathbf{v}_1^*\mathbf{M}\mathbf{v}_1$, we have a quadratic equation of λ, i.e.,

$$\lambda^2 + \alpha\lambda + \beta = 0.$$

Note that every eigenvalue of $\mathbf{P}$ satisfies the above equation. The solution of the above equation is

$$\lambda_{1,2} = \frac{1}{2}(-\alpha \pm \sqrt{\alpha^2 - 4\beta}).$$

If $\alpha^2 - 4\beta \geq 0$, the real parts of $\lambda_{1,2}$ are:

$$Re\{\lambda_{1,2}\} = \frac{1}{2}(-\alpha \pm \sqrt{\alpha^2 - 4\beta}).$$

Recall that α and β are positive, we claim that $Re\{\lambda_{1,2}\}$ are negative.

If $\alpha^2 - 4\beta < 0$, the real parts of $\lambda_{1,2}$ are:

$$Re\{\lambda_{1,2}\} = -\frac{1}{2}\alpha.$$

Since α is positive, it's obviously that the real parts of $\lambda_{1,2}$ are negative.

In all cases, we conclude that the real parts of $\lambda_{1,2}$ are always negative.

Theorem 3: For the second order ordinary differential equation

$$\mathbf{L}\ddot{\mathbf{x}} + \mathbf{M}\dot{\mathbf{x}} + \mathbf{N}\mathbf{x} = 0,$$

if its coefficient matrices $\mathbf{L}$, $\mathbf{M}$ and $\mathbf{N}$ are all positive definite, then it is asymptotic stable.

Proof. Note that for the positive definite matrix $\mathbf{L}$, we have a decomposition such that $\mathbf{L} = \mathbf{L}^{\frac{1}{2}}\mathbf{L}^{\frac{1}{2}}$ and $\mathbf{L}^{-1} = \mathbf{L}^{-\frac{1}{2}}\mathbf{L}^{-\frac{1}{2}}$, where $\mathbf{L}^{\frac{1}{2}}$ and $\mathbf{L}^{-\frac{1}{2}}$ are positive definite. Now we define the transformation $\tilde{\mathbf{x}} = \mathbf{L}^{\frac{1}{2}}\mathbf{x}$ or $\mathbf{x} = \mathbf{L}^{-\frac{1}{2}}\tilde{\mathbf{x}}$, using this transformation we have

$$\mathbf{L}\mathbf{L}^{-\frac{1}{2}}\ddot{\tilde{\mathbf{x}}} + \mathbf{M}\mathbf{L}^{-\frac{1}{2}}\dot{\tilde{\mathbf{x}}} + \mathbf{N}\mathbf{L}^{-\frac{1}{2}}\tilde{\mathbf{x}} = 0 \tag{23}$$

Premultiplying $\mathbf{L}^{-\frac{1}{2}}$ to both hands of equation, we get

$$\ddot{\tilde{\mathbf{x}}} + \mathbf{L}^{-\frac{1}{2}}\mathbf{M}\mathbf{L}^{-\frac{1}{2}}\dot{\tilde{\mathbf{x}}} + \mathbf{L}^{-\frac{1}{2}}\mathbf{N}\mathbf{L}^{-\frac{1}{2}}\tilde{\mathbf{x}} = 0 \tag{24}$$

This system is exactly of the form used in *Theorem 2*, but instead of $\mathbf{M}$ and $\mathbf{N}$ we now have $\mathbf{L}^{-\frac{1}{2}}\mathbf{M}\mathbf{L}^{-\frac{1}{2}}$ and $\mathbf{L}^{-\frac{1}{2}}\mathbf{N}\mathbf{L}^{-\frac{1}{2}}$. If the later system is asymptotic stable, it implies that (1) is asymptotic stable, since the two systems differ only by a non-singular transformation. Therefore the global asymptotic stability of (6) follows from *Theorem 2*.

Theorem 4: If the matrix $\mathbf{A}^T\mathbf{A}$ or $\mathbf{D}^T\mathbf{D}$ is non-singular and $k > 0$, then the neural network described by differential Eqns. (4a, 4b and 4c) is asymptotic stable.

Proof. Since $\mathbf{A}^T\mathbf{A}$ or $\mathbf{D}^T\mathbf{D}$ is symmetric and non-singular, the matrices $\mathbf{I} + k\mathbf{Q} + k^2\mathbf{D}^T\mathbf{D} + k^2\mathbf{A}^T\mathbf{A}$, $\mathbf{Q} + 2k\mathbf{D}^T\mathbf{D} + 2k\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}^T\mathbf{A} + \mathbf{D}^T\mathbf{D}$ are positive definite. By *Theorem 3*, we conclude that the dynamical system of (4a, 4b and 4c) is asymptotic stable in the sense of Lyapunov.

4 Simulation Results

To demonstrate the behavior and properties of the proposed nonlinear neural network model, one example with four different initial vectors is simulated. The simulation is conducted with MATLAB. We use the Euler method to solve the neural system of ordinary differential equations (4a, 4b and 4c).
Consider the following quadratic programming problem:

$$\begin{array}{ll} \text{Minimize} & 0.4x_1 + 1.25x_1^2 + x_2^2 - x_1x_2 + 0.5x_3^2 + 0.5x_4^2, \\ \text{subject to} & -0.5x_1 - x_2 + x_4 \geq -0.5, \\ & x_1 + 0.5x_2 - x_3 = 0.4, \\ & \mathbf{x} \geq 0. \end{array} \tag{25}$$

We tested the proposed neural network guided by (4a, 4b and 4c) with four different initial vectors(four combination for feasible and infeasible vectors) for the primal and dual problems:
case 1: $\mathbf{x}_0 = (1, 1, 1.1, 2)^T$(feasible) and $(\mathbf{y}_0, \mathbf{z}_0) = (-1, 1)$(feasible),
case 2: $\mathbf{x}_0 = (1, 1, 1.1, 2)^T$(feasible) and $(\mathbf{y}_0, \mathbf{z}_0) = (1, -1)$(infeasible),
case 3: $\mathbf{x}_0 = (1, 2, -1, -2)^T$(infeasible) and $(\mathbf{y}_0, \mathbf{z}_0) = (-3, 1)$(feasible),
case 4: $\mathbf{x}_0 = (-1, 2, 4, 3)^T$(infeasible) and $(\mathbf{y}_0, \mathbf{z}_0) = (1, -1)$(infeasible),
and the transient behaviors of $\mathbf{x}(t)$ are depicted in Fig. 2, Fig. 3, Fig. 4, Fig. 5 respectively.

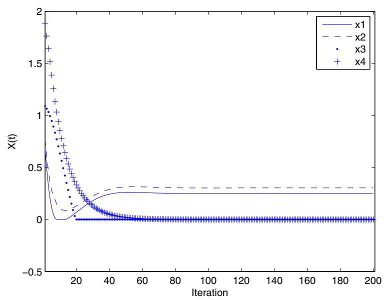

Fig. 2. Transient behavior of $\mathbf{x}(t)$ for case 1

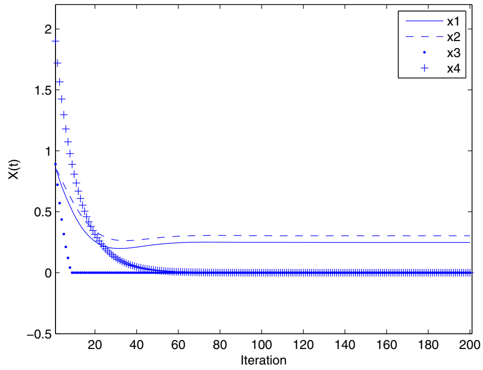

Fig. 3. Transient behavior of $\mathbf{x}(t)$ for case 2

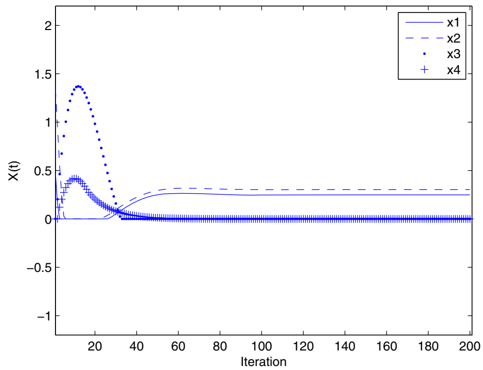

Fig. 4. Transient behavior of $\mathbf{x}(t)$ for case 3

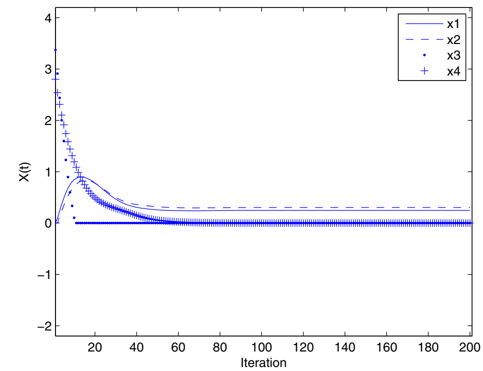

Fig. 5. Transient behavior of $\mathbf{x}(t)$ for case 4

It can be seen that after about 80 iterations the vector $\mathbf{x}$ will converge to the optimal solution $\mathbf{x}^\star = (0.2483, 0.3034, 0, 0)^T$ for all cases.

5 Conclusions

This paper presents a new nonlinear neural network to solving quadratic programming problems. It's proved that this novel neural network is stable in the sense of Lyapunov under certain conditions. Numerical simulation results show the effectiveness and efficiency this neural network. Future research direction include application the proposed neural network to solving the K-Winners-Take-All (KWTA) problem [34–36] based on linear programming or quadratic programming formulations, assignment problem [37,38] and maximum flow problem [39,40], extension the nonlinear model to convex programming and more general optimization problems.

References

1. Franklin, N.J.: Methods of Mathematical Economics: Linear and Nonlinear Programming. Society for Industrial and Applied Mathematics, Fixed-Point Theorems (2002)
2. Luenberger, D.G., Ye, Y.: Linear and Nonlinear Programming. Springer (2008)
3. Bazaraa, M.S., Sherali, D.H., Shetty C.M.: Nonlinear Programming: Theory and Algorithms. Wiley-Interscience (2006)
4. Surhone, L.M., Tennoe, M.F., Henssonow, S.F.: Quadratic Programming. Betascript Publishing (2010)
5. Gärtner, B., Matousek, J.: Approximation Algorithms and Semidefinite Programming. Springer (2012)
6. Griva, I., Nash, S.G., Sofer, A.: Linear and Nonlinear Optimization, 2nd edn. Society for Industrial Mathematics (2008)
7. Hertog, D.: Interior Point Approach to Linear, Quadratic and Convex Programming: Algorithms and Complexity. Springer (1994)
8. Lee, G.N.: Quadratic Programming and Affine Variational Inequalities: A Qualitative Study. Springer (2010)
9. Tank, D.W., Hopfield, J.J.: Simple Neural Optimization Networks: An A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit. IEEE Transactions on Circuits and Systems **33**(5), 533–541 (1986)
10. Hopfield, J.J., Tank, D.W.: Computing with Neural Circuits: A Model. Science **233**, 625–633 (1986)
11. Zhang, S., Constantinides, A.G.: Lagrange Programming Neural Networks. IEEE Transactions on Circuits and Systems II **39**(7), 441–452 (1992)
12. Wang, J.: A Deterministic Annealing Neural Network for Convex Programming. Neural Networks **5**(4), 962–971 (1994)
13. Ding, K., Huang, N.J.: A New Class of Interval Projection Neural Networks for Solving Interval Quadratic Program. Chaos, Solitons and Fractals **3**, 149–164 (2008)
14. Yang, Y., Cao, J.: Solving Quadratic Programming Problems by Delayed Projection Neural Network. IEEE Transactions on Neural Networks **17**(6), 1630–1634 (2006)

15. Zhang, Y., Wang, J.: A Dual Neural Network for Convex Quadratic Programming Subject to Linear Equality and Inequality Constraints. Physics Letters A **298**, 271–278 (2002)
16. Hu, X., Wang, J.: An Improved Dual Neural Network for Solving a Class of Quadratic Programming Problems and Its K-Winners-Take-All Application. IEEE Transactions on Neural Networks **19**(2), 2022–2031 (2009)
17. Xia, Y., Feng, G., Wang, J.: Primal-Dual Neural Network for Online Resolving Constrained Kinematic Redundancy in Robot Motion Control. IEEE Transactions on Systems, Man and Cybernetics **35**(1), 54–64 (2005)
18. Kennedy, M.P., Chua, L.O.: Neural Networks for Nonlinear Programming. IEEE Transactions on Circuits and Systems **35**(5), 554–562 (1988)
19. Maa, C.Y., Schanblatt, M.A.: A Two-Phase Optimization Neural Network. IEEE Transactions on Neural Network **3**(6), 1003–1009 (1992)
20. Xia, Y.: A New Neural Network for Solving Linear Programming Problems and Its Application. IEEE Transactions on Neural Networks **7**(2), 525–529 (1996)
21. Nguyen, K.V.: A Nonlinear Neural Network for Solving Linear Programming Problems. In: International Symposium on Mathematical Programming, ISMP 2000, Atlanta, GA, USA (2000)
22. Taylor, J.G.: Mathematical Approaches to Neural Networks. North-Holland (1993)
23. Harvey, R.L.: Neural Network Principles. Prentice Hall (1994)
24. Veelenturf, L.: Analysis and Applications of Artifical Neural Networks. Prentice Hall (1995)
25. Rojas, R., Feldman, J.: Neural Networks A Systematic Introduction. Springer (1996)
26. Mehrotra, K., Mohan, C.K., Ranka, S.: Elements of Artificial Neural Networks. MIT Press (1997)
27. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice Hall (1998)
28. Michel, A., Liu, D.: Qualitative Analysis and Synthesis of Recurrent Neural Networks. CRC Press (2001)
29. Hagan, M.T., Demuth, H.B., Beale, M.H.: Neural Network Design. Martin Hagan (2002)
30. Gurney, K.: An Introduction to Neural Networks. CRC Press (2003)
31. Graupe, D.: Principles of Artificial Neural Networks, 2nd edn. World Scientific Pub Co., Inc. (2007)
32. Boyd, S., Vandenbeghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
33. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and Distributed Computation: Numerical Methods. Prentice-Hall, Englewood Cliffs (1989)
34. Maass, W.: Neural Computation with Winner-Take-All as the Only Nonlinear Operation. Advances in Neural Information Processing Systems **12**, 293–299 (2000)
35. Marinov, C.A., Calvert, B.D.: Performance Analysis for a K-Winners-Take-All Analog Neural Network: Basic Theory. IEEE Transactions on Neural Networks **14**(4), 766–780 (2003)
36. Liu, S.B., Wang, J.: A Simplified Dual Neural Network for Quadratic Programming with Its KWTA Application. IEEE Transactions on Neural Networks **17**(6), 1500–1510 (2006)
37. Wang, J.: Analogue Neural Network for Solving the Assignment Problem. Electronics Letters **28**(11), 1047–1050 (1992)

38. Hu, X., Wang, J.: Solving the Assignment Problem with the Improved Dual Neural Network. In: Liu, D., Zhang, H., Polycarpou, M., Alippi, C., He, H. (eds.) ISNN 2011, Part I. LNCS, vol. 6675, pp. 547–556. Springer, Heidelberg (2011)
39. Effati, S., Ranjbar, M.: Neural Network Models for Solving the Maximum Flow Problem. Applications and Applied Mathematics **3**(3), 149–162 (2008)
40. Nazemi, A., Omidi, F.: A Capable Neural Network Model for Solving the Maximum Flow Problem. Journal of Computational and Applied Mathematics **236**(14), 3498–3513 (2012)