A New Recurrent Neural Network with Fewer Neurons for Quadratic Programming Problems

Sanfeng Chen[1], Xin Han[1,2], Fei Tang[1], and Guangming Lin[1](✉)

[1] Shenzhen Institute of Information Technology,
Shenzhen 518029, Guangdong, China
Linggm@sziit.edu.cn
[2] School of Mechanical and Electrical Engineering,
Jiangxi University of Science and Technology, Ganzhou 341000, Jiangxi, China

Abstract. A new recurrent neural network is presented to solve a general quadratic programming problem in real time. In contrast with the available neural networks, the new neural network is with fewer neurons for solving quadratic programming problems. The global convergence of the model is proven with contraction analysis. The discrete time model and an alternative model for solving the problem under irredundant equality constraints are also studied. Simulation results demonstrate that the proposed recurrent neural networks are effective.

Keywords: Recurrent neural network · Quadratic programming
Contraction analysis · Global convergence

1 Introduction

There are a lot of studies of recurrent neural networks focusing on the filed of signal processing [1, 2], pattern classification [3, 4], robotics [5, 6], optimization [7], and so on. Especially, with the invention of the Hopfield [8], it was specially invented for solving online optimization. Recurrent neural networks are becoming a popular research branch in the field of online optimization. They are with powerful parallelism and online solving capability. Recurrent neural networks have made huge advances for online optimization in both theory and application. A recurrent neural network [9] is developed for nonlinear programming problems, where a penalty term is introduced as equality and inequality constraints, and it converges to an approximate optimal solution. A switched-capacitor neural network is proposed [10] for solving nonlinear convex programming problems. However, the model will be unstable in the case that the optimal solution is outside the feasible region. A neural network is proposed for solving linear quadratic programming problems [11]. The optimal solution is proven globally converged. Some slack variables is introduced to the problem, which leads the dimension of model is too large. A dual neural network is proposed for reducing the dimension. It is composed of a single layer of neurons,and the dimension of the dual network is equal to its neurons. The model and its modifications [13, 14] are introduced to kinematic control of

K. Li et al. (Eds.): ISICA 2017, CCIS 873, pp. 3–16, 2018.
https://doi.org/10.1007/978-981-13-1648-7_1

robot [12, 15]. A simplified dual neural network is proposed [16]. It much reduces complexity while the convergence property is sound. The model is applied to the KWTA problem in real time [17], which is just a single neuron. However, it just deals with quadratic programming problem with a square quadratic term in box constraints and cost function. A recurrent neural network for solving general quadratic programming problems is proposed. It is with fewer neurons, and the dimension of the model is greatly reduced while keeping sound accuracy and efficiency.

The remainder of this paper is organized as follows. In Sect. 2, A neural network model is presented for solving quadratic programming problems. In Sect. 3, the convergence of the neural network is analyzed and it is proven to be globally convergent to the optimal solution of the quadratic programming problems. A discrete-time model in Sect. 4 for solving the same problem and an alternative neural network model for solving the quadratic programming problem under irredundant equality constraints are studied. In Sect. 5, numerical examples are given to demonstrate the effectiveness of our method. Section 6 is the conclusion.

In this paper, $\mathbb{R}$ denotes the real number field, A^T represents the transport matrix of A, I denotes a unitary matrix.

2 Mathematical Model

The general quadratic programming problem as following is studied:

$$\begin{aligned} &\min(\frac{1}{2}x^T Wx + c^T x) \\ &\text{s.t.} \quad \begin{aligned} Ax &= b \\ Ex &\le e \end{aligned} \end{aligned} \tag{1}$$

Where $W \in \mathbb{R}^{n \times n}$, $x \in \mathbb{R}^n$ denotes a positive definite matrix, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $e \in \mathbb{R}^q$ and $E \in \mathbb{R}^{q \times n}$. The equality constraint $Ax = b$ is transformed to two inequalities equivalently: $-Ax \le -b$ and $Ax \le b$. The Eq. (1) is transferred as a quadratic programming problem subject to inequality constraints. It is as following:

$$\begin{aligned} &\min(\frac{1}{2}x^T Wx + c^T x) \\ &\text{s.t.} \quad \frac{1}{2}Bx \le d \end{aligned} \tag{2}$$

$$B = \begin{bmatrix} A \\ -A \\ E \end{bmatrix}, \ d = \begin{bmatrix} b \\ -b \\ e \end{bmatrix} \tag{3}$$

Where $B \in \mathbb{R}^{p \times n}$, $d \in \mathbb{R}^{p}$ and $p = 2 \times m + q$. The inequality constraint of (2) is transformed as $\max(Bx - d) \leq 0$. The $\max(x)$ denotes the largest element of vector x. The problem (1) is equivalent to the Eq. (4):

$$\begin{aligned} &\min(\frac{1}{2}x^T Wx + c^T x) \\ &\text{s.t. } \; B_\sigma^T x - d_\sigma \leq 0 \end{aligned} \tag{4}$$

In the Eq. (4), σ denotes the row No. of the biggest element of $Bx - d$, and B_σ^T represents the σ^{th} row of B, d_σ denotes the σ^{th} element of d. According to the KKT terms, the solution of problem (4) meets the requirements:

$$\begin{gathered} Wx + c - \mu B_\sigma = 0 \\ \begin{cases} B_\sigma^T x - d_\sigma = 0 \; if \;\; \mu \leq 0 \\ B_\sigma^T x - d_\sigma \leq 0 \; if \;\; \mu = 0 \end{cases} \end{gathered} \tag{5}$$

The dual variable of inequality constraint in the Eq. (4) is represented with $\mu \in \mathbb{R}$. The Eq. (5) is simplified with an upper saturation function as following:

$$\begin{aligned} &Wx + c - \mu B_\sigma = 0 \\ &B_\sigma^T x - d_\sigma = g(B_\sigma^T x - d_\sigma - \mu) \end{aligned} \tag{6}$$

Where the upper saturation function $g(.)$ is as following:

$$g(.) = \begin{cases} 0 & x > 0 \\ x & x \leq 0 \end{cases} \tag{7}$$

The W is positive definite, x could be explicitly solved with μ and the first equality in Eq. (6) as following:

$$x = \mu W^{-1} B_\sigma - W^{-1} c \tag{8}$$

A dynamic neuron is used to solve μ in Eq. (6) as following:

$$\in \dot{\mu} = g(B_\sigma^T x - d_\sigma - \mu) - B_\sigma^T x + d_\sigma \tag{9}$$

Where $\in \; > 0$ is a scaling parameter. Substituting (8) into (9) generates the neural network dynamics with the following state equation and output equation,

State equation:

$$\in \dot{\mu} = g(-\mu + B_\sigma^T W^{-1} B_\sigma \mu - B_\sigma^T W^{-1} c - d_\sigma) - B_\sigma^T W^{-1} B_\sigma \mu + B_\sigma^T W^{-1} c + d_\sigma \tag{10}$$

Output equation:

$$x = \mu W^{-1} B_\sigma - W^{-1} c \tag{11}$$

Where σ is the row No. of $\max(Bx - d)$, d and B and are as Eq. (3).

Remark 2.1: Only one dynamic neuron is required in the neural network (10), which is nothing to do with the conditions of Eq. (1). There are at least q dynamic neurons in recurrent neural networks for solving a general quadratic programming problem in [13–16, 18]. It is the No. of inequalities of problem (1). However, the proposed model is just a single dynamic neuron, which greatly reduces the number of neurons and computational complexity.

Remark 2.2: The neural network dynamic modeled by (10) is a switched dynamic system. It switches in a family of dynamic systems under the endogenous switching signal σ (signal flow in the neural network is plotted in Fig. 1)

$$\begin{array}{l} \in \dot{\mu} = f_\sigma(\mu), \sigma \in S = \{1, 2.\ldots, p\} \\ \sigma : \mu \to S \end{array} \tag{12}$$

Where $f_\sigma(\mu) = g(B_\sigma^T x - d_\sigma - \mu) - B_\sigma^T x + d_\sigma$

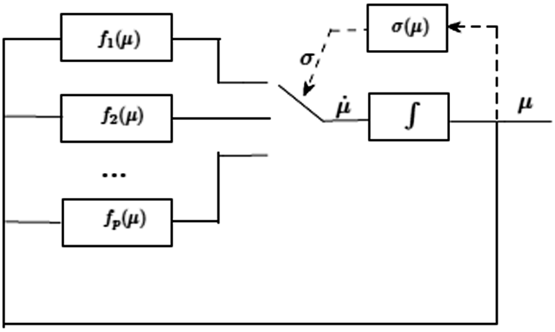

Fig. 1. Signal flow of the simplified neural network.

Remark 2.3: The proposed model is liable to be implemented with hardware. In comparison with the research [16], the new part with hardware implementation is a switching of the proposed network (10). The σ is a switching signal, which is equal to the row No. of $\max(Bx - d)$. It is the winner of $Bx - d$. σ is gotten with a WTA circuit [19]. It is worked as the switching signal for B_σ and d_σ in (10). There are a lot of studies on WTA circuits, and it is easily implemented with CMOS [20]. It is also easily implemented by a recurrent neural network just a single dynamic neuron [19]. There is just two dynamic neurons in the neural network (10), which is much less than the solutions of [13–18].

3 Convergence Analysis

It is a feasible way to prove the global convergence of the proposed system by constructing a common negative definite Lyapunov function. However, choosing such a common Lyapunov function is a difficult problem. Some researches on the contraction theory [21, 22] greatly simplify the proof process with virtual dynamics of the system. In this paper, the contraction analysis is made to prove the proposed model (10) convergent. The proof process is based on one definition and two lemmas:

Definition 3.1 ([22]): Given the system equations $\dot{x} = f(x, t)$, a region of the state space is called a contraction region with respect to a uniformly positive definite metric $M(x, t) = \Theta^T \Theta$ if $\frac{\partial f^T}{\partial x} M + M \frac{\partial f}{\partial x} + \dot{M} \leq -\beta M$ (with $\beta > 0$) in that region.

Lemma 3.1 [22]: Given the system equations
$\dot{x} = f(x, t)$, any trajectory, which starts in a ball of constant radius with respect to the metric $M(x, t)$, center at a given trajectory and contained at all times in a contraction region with respect o $M(x, t)$, remains in that ball and converges exponentially to this trajectory. furthermore, global exponential convergence to the given trajectory is guaranteed if the whole state space is a contraction region with respect to the metric $M(x, t)$.

Remark 3.1: As pointed out in [21], Lemma 3.1 holds for all switched systems in case that all subsystems in the switched family are within a same contraction region $M(x, t)$.

Lemma 3.2: The solution to problem (6) (represented by μ^*, x^*) is the optimal solution to problem (1). At the same time, μ^* is an equilibrium point of the state equation of the proposed model, which is as following:

$$g(-\mu + B_\sigma^T W^{-1} B_\sigma \mu - B_\sigma^T W^{-1} c - d_\sigma) - B_\sigma^T W^{-1} B_\sigma \mu + B_\sigma^T W^{-1} c + d_\sigma = 0$$

The optimal solution also satisfies (11) as following: $x^* = \mu W^{-1} B_\sigma - W^{-1} c$.

Proof: According to the KKT condition, equation sets (5) are the optimal solution to problem (4). Equation sets (6) are equivalent to equation sets (5), the solution to equation set (6) is equivalent to the optimal solution to problem (1). They are as following:

$$g(-\mu^* + B_\sigma^T W^{-1} B_\sigma \mu^* - B_\sigma^T W^{-1} c - d_\sigma) - B_\sigma^T W^{-1} B_\sigma \mu^* + B_\sigma^T W^{-1} c + d_\sigma = 0$$

Where $x^* = \mu W^{-1} B_\sigma - W^{-1} c$.

Now, we are on the stage to state the convergence result of the proposed model.

Theorem 3.1: The proposed (10) converges to the equilibrium point from any start point $\mu \in \mathbb{R}$, and the Eq. (11) is the optimal solution to Eqs. (1).

Proof: Based on the contraction theory for analyzing the convergence of (10):

$$\dot{\mu} = f(\mu, t) = \frac{1}{\varepsilon}(g(-\mu + B_\sigma^T W^{-1} B_\sigma \mu - B_\sigma^T W^{-1} c - d_\sigma) - B_\sigma^T W^{-1} B_\sigma \mu + B_\sigma^T W^{-1} c + d_\sigma) \quad (13)$$

The partial derivative $\frac{\partial f}{\partial \mu}$ is as following:

$$\frac{\partial f}{\partial \mu} = \begin{cases} -\frac{B_\sigma^T W^{-1} B_\sigma}{\varepsilon} \ \mathit{if}\ \upsilon \geq \ 0 \\ -\frac{1}{\varepsilon} \ \upsilon < 0 \end{cases} \quad (14)$$

Where $\upsilon = -\mu + B_\sigma^T W^{-1} B_\sigma \mu - B_\sigma^T W^{-1} c - d_\sigma$. Note that the derivate $g'(x) = 0$ if $x \geq \ 0$ and $g'(x) = 1$ if $x < 0$ (more exactly, it should be the upper right-hand derivative since the function $g(x)$ is non-smooth at $x = 0$.) is used in the derivation of (14). By choose $M = 1$, we have

$$\frac{\partial f^T}{\partial \mu} M + M \frac{\partial f}{\partial \mu} + \dot{M} = 2\frac{\partial f}{\partial \mu} \leq -\frac{2\min(1, B_\sigma^T W^{-1} B_\sigma)}{\varepsilon} \quad (15)$$

Since W^{-1} is positive definite and B_σ is not equal to 0 (otherwise, the corresponding inequality in (2) does not include x), we conclude that $B_\sigma^T W^{-1} B_\sigma > 0$ and $\frac{\min(1, B_\sigma^T W^{-1} B_\sigma)}{\varepsilon} > 0$.

By defining $\beta = \frac{\min\{1, \min_i(1, B_i^T W^{-1} B_i)}{\varepsilon}$, with B_i^T denoting the ith row of B and the set $\mathbf{S} = \{1, 2, \ldots, p\}$, we have the following inequality for all $\mu \in \mathbb{R}$ and all $\sigma \in \mathbf{S}$ under the same metric M,

$$\frac{\partial f^T}{\partial x} M + M \frac{\partial f}{\partial x} + \dot{M} \leq -\beta M \quad (16)$$

The whole state space of μ is a contraction region.

Based on Lemma 3.2, μ^* and x^*, is the optimal solution to problem (1), and it is also an equilibrium point to Eq. (10).

4 Extensions

4.1 Discrete-Time Model

In this part, we propose the discrete-time model to solve the problem (1) and give conditions for global convergence

Replacing μ, $\dot{\mu}$ with μ_n, $\frac{\mu_{n+1} - \mu_n}{\Delta t}$ in (10), respectively, and denoting $\Upsilon = \frac{\Delta t}{\varepsilon}$, we get the following state equation and output equation of the discrete model as following with some trivial manipulations. state equation:

$$\begin{aligned} \mu_{n+1} = {} & \Upsilon g(-\mu_n + B_\sigma^T W^{-1} B_\sigma \mu_n - B_\sigma^T W^{-1} c - d_\sigma) + (1 \\ & - \Upsilon B_\sigma^T W^{-1} B_\sigma)\mu_n + \Upsilon B_\sigma^T W^{-1} c + \Upsilon d_\sigma \end{aligned} \quad (17)$$

The output equation:

$$x_n = \mu_n W^{-1} B_\sigma - W^{-1} c \tag{18}$$

Where σ is the row No. of the largest element of $Bx_n - d$, B and d are as equation sets (3).

In the discrete-time case, the contraction theory is an extension of the well-known contraction mapping theorem. We still use the contraction theory to analyze the convergence. For discrete-time systems, the definition of a contraction region and the condition for contraction are stated as below.

Definition 4.1 ([22]): The discrete-time model is as following:

Equation $x_{n+1} = f_n(x_n, n)$, is a contraction region. Positive definite metric $M_n(x_n, n) = \Theta_n^T \Theta_n$, given in the region $\exists \beta > 0$, $F_n^T F_n - I \le -\beta I < 0$, where $F_n = \Theta_{n+1} \frac{\partial f_n}{\partial x_n} \Theta_n^{-1}$.

Lemma 4.1 ([22]): If the whole state space is a contraction region, The global convergence to the given trajectory is made sure.

Theorem 4.1: The discrete model (17) will converge to the equilibrium point after initialization $\mu \in \mathbb{R}$ and the output of equilibrium point, as shown in (18), is also the optimal solution to problem (1) under:

$$0 < \gamma < 2, \; \gamma < \frac{2}{B_i^T W^{-1} B_i} \text{ for all } i \in \mathbf{S} \tag{19}$$

where B_i^T denotes the ith row of B and $\mathbf{S} = \{1, 2, 3, \ldots, p\}$

Proof: Firstly, solve the equilibrium point for the discrete model (17), and get the output of the Eq. (18). Secondly, prove the global contraction to the equilibrium point.

Step 1: Comparing the discrete-time model (17) and its output (18) with the continuous-time model (10) and its output (11), we can find that they have the same equilibrium and the same output at this equilibrium. According to the Lemma 3.2, the solution to equation set (6) is an equilibrium to the discrete model (17) too, and the output at the point is the optimal solution to the problem (1).

Step 2: To the discrete model (17),

$$\begin{aligned} \mu_{n+1} &= f_n(x_n, n) \\ &= \Upsilon g(-\mu_n + B_\sigma^T W^{-1} B_\sigma \mu_n - B_\sigma^T W^{-1} c - d_\sigma) + (1 - \Upsilon B_\sigma^T W^{-1} B_\sigma)\mu_n + \Upsilon B_\sigma^T W^{-1} c + \Upsilon d_\sigma \end{aligned} \tag{20}$$

Calculate $\frac{\partial f_n}{\partial \mu_n}$ as follows:

$$\frac{\partial f_n}{\partial \mu_n} = \begin{cases} 1 - \Upsilon B_\sigma^T W^{-1} B_\sigma \; if \; v_n \ge 0 \\ 1 - \Upsilon if \; v_n < 0 \end{cases} \tag{21}$$

Where $v_n = -\mu_n + B_\sigma^T W^{-1} B_\sigma \mu_n - B_\sigma^T W^{-1} c - d_\sigma$. Choosing $\Theta_n = 1$ for $n = 0, 1, 2, \ldots, n$ then

$$F_n = \Theta_{n+1} \frac{\partial f_n}{\partial \mu_n} \Theta_n^{-1} \tag{22}$$

Based on (21) and (22),

$$\begin{aligned} F_n^T F_n &\leq (max\{|1-\gamma|, |1-\gamma B_\sigma^T W^{-1} B_\sigma|\})^2 \\ &\leq (max\{|1-\gamma|, max_{i\in \mathbf{S}}\{|1-\gamma B_\sigma^T W^{-1} B_i|\}\})^2 \end{aligned} \tag{23}$$

Set $\beta = 1 - (max\{|1-\gamma|, max_{i\in \mathbf{S}}\{|1-\gamma B_\sigma^T W^{-1} B_i|\}\})^2$. To all $\mu_n \in \mathbb{R}$, $\sigma \in \mathbf{S}$, the inequality as following (24) is valid.

$$F_n^T F_n - I \leq \beta I \tag{24}$$

Since $0 < \gamma < 2$, $0 < \gamma B_i^T W^{-1} B_i < 2$ for $i \in \mathbf{S}$, we get $|1-\gamma| < 1, |1 - \gamma B_i^T W^{-1} B_i| < 1$ for $i \in \mathbf{S}$. Therefore, $(max\{|1-\gamma|, max_{i\in \mathbf{S}}\{|1-\gamma B_\sigma^T W^{-1} B_i|\}\})^2 \leq 1$ and $\beta > 0$.

Then the whole state space is within contraction range. Let the equilibrium point of (17) to be the given trajectory, according to Lemma 4.1, the discrete model (17) will convergence to the equilibrium point.

4.2 Irredundant Equality Constraint

Based on that the equality constraint exists without redundancy, there is another way to tackle the problem (1). This is to say, the matrix $A \in \mathbb{R}^{m\times n}$ in (1) problem meets the requirement of rank $(A) = m$ and $m < n$.

$Ex \leq e$ in problem (1) is replaced by $max(Ex - e) \leq 0$, then

$$\begin{aligned} &\min\left(\frac{1}{2} x^T W x + C^T x\right) \\ &\text{s.t. } Ax = b \\ &E_\sigma^T x - e_\sigma \leq 0 \end{aligned} \tag{25}$$

Based on KKT conditions and the upper saturation function $g(\cdot)$, the solution to problem (25) meets,

$$\begin{gathered} Wx + c - A^T y - \mu B_\sigma^T = 0 \\ Ax = b \\ E_\sigma^T x - e_\sigma = g(E_\sigma^T x - e_\sigma - \mu) \end{gathered} \tag{26}$$

Where $y \in \mathbb{R}^m$, $\mu \in \mathbb{R}$. x and y can be explicitly solved in terms of μ under the above three equations:

$$\begin{aligned} &x = \mu P E_\sigma + s \\ &y = (AW^{-1}A^T)^{-1}(-AW^{-1}B_\sigma \mu + AW^{-1}c + b) \end{aligned} \tag{27}$$

Where $P = -W^{-1}A^T(AW^{-1}A^T)^{-1}AW^{-1} + W^{-1}$, S $= W^{-1}(A^T(AW^{-1}A^T)^{-1}(AW^{-1}c + b) - c)$ $P \in \mathbb{R}^{n\times n}$. The third equation in (26) is solved as following:

$$\varepsilon\dot{\mu} = g(E_\sigma^T x - e_\sigma - \mu) - E_\sigma^T x + e_\sigma \tag{28}$$

Where $\varepsilon > 0$ is a scaling parameter. The state equation of the neural network is as following:

$$\varepsilon\dot{\mu} = g(E_\sigma^T PE_\sigma\mu - \mu + E_\sigma^T s - e_\sigma) - E_\sigma^T PE_\sigma\mu - E_\sigma^T s + e_\sigma \tag{29}$$

Before stating the convergence result about the neural network (29), the following lemma is presented, which is used in the convergence proof.

Lemma 4.2: The symmetric matrix $P \in \mathbb{R}^{n\times n}, P = -W^{-1}A^T(AW^{-1}A^T)^{-1}AW^{-1} + W^{-1}$ is semi-positive definite, i.e., $z^T Pz \geq 0$ for all $z \in \mathbb{R}^n$ and $z^T(W^{-1}A^T(AW^{-1}A^T)^{-1}A - I) \neq 0$.

Proof: Since $W^{-1} \in \mathbb{R}^{n\times n}$ is positive definite, it can be factorized into $W^{-1} = QQ^T$ with Q positive definite. Defining $G \in \mathbb{R}^{m\times n}$, $G = AQ$, then G is also row full rank (since A is row full rank and Q is positive definite) and therefore G can be decomposed into $G = U[\wedge\ 0]V$ via singular value decomposition, where $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$ are both unitary matrices, $[\wedge\ 0] \in \mathbb{R}^{m\times n}$, $\wedge \in \mathbb{R}^{m\times m}$ is a diagonal matrix with all positive elements on the diagonal. Bringing

$$\begin{aligned} W^{-1} &= QQ^T,\ G = AQ,\ \text{and}\ G = U[\wedge\ 0]V \text{ into } P \\ &= -W^{-1}A^T(AW^{-1}A^T)^{-1}AW^{-1} + W^{-1} \end{aligned}$$

We get the equations as (30) as below. The stability of the neural network (29) and its global convergence to the optimal solution of (1) is guaranteed by the following theorem.

Theorem 4.2: The neural network (29) exponentially converges to its equilibrium from any initial point $\mu \in \mathbb{R}$, and its output at this equilibrium by following the output Eq. (27) is the optimal solution to problem (1), if $rank(A) = m$ in (1) and $(E_i^T(W^{-1}A^T(AW^{-1}A^T)^{-1}A - I) \neq 0$

For $i = 1, 2, \ldots, q$, with E_i^T denoting the ith row of the matrix E.

Proof: This theorem can be proven in a two-step procedure similar to the proof of Theorem 4.1. The first step is to solve the equilibrium and show that the output at this equilibrium is the optimal solution to the problem (1). The second step is to prove the global contraction to the equilibrium. Note that the condition $(E_i^T(W^{-1}A^T(AW^{-1}A^T)^{-1}A - I) \neq 0$ For $i = 1, 2, \ldots, q$, according to Lemma 4.2, guarantees $E_\sigma^T PE_\sigma > 0$ for all possible switching signal σ. Based on this result, global contraction of the neural network (29) can be proven. Detailed proof process for the two steps is omitted.

$$
\begin{aligned}
P &= -QQ^TA^T(AQQ^TA^T)^{-1}AQQ^T + QQ^T \\
&= -QG^T(GG^T)^{-1}GQ^T + QQ^T \\
&= -QV^T\begin{bmatrix} \Lambda \\ 0 \end{bmatrix}U^T(U\Lambda^{-2}U^T)U[\Lambda \quad 0]VQ^T + QQ^T \\
&= -QV^T\begin{bmatrix} I_{m\times m} & 0 \\ 0 & 0 \end{bmatrix}VQ + QV^TVQ^T \\
&= (VQ^T)^T\begin{bmatrix} 0 & 0 \\ 0 & I_{(n-m)\times(n-m)} \end{bmatrix}VQ^T \geq 0
\end{aligned}
\tag{30}
$$

5 Simulations

5.1 Continuous-Time Neural Network Model

The following problem will be solved for showing the convergence of the continuous-time neural network modelled with (10) and (11).

$$
\begin{aligned}
&\min(3x_1^2 + 3x_2^2 + 4x_3^2 + 5x_4^2 + 3x_1x_2 + 5x_1x_3 + x_2x_4 - 11x_1 - 5x_4) \\
&\text{s.t.} \quad
\begin{aligned}
&-3x_1 + 3x_2 + 2x_3 - x_4 \leq 0 \\
&-4x_1 - x_2 + x_3 + 2x_4 \leq 0 \\
&-x_1 + x_2 \leq -1 \\
&-2 \leq 3x_1 + x_3 \leq 4
\end{aligned}
\end{aligned}
\tag{31}
$$

To the model, the parameters are as following:

$$
\begin{aligned}
W &= \begin{bmatrix} 6 & 3 & 5 & 0 \\ 3 & 6 & 0 & 1 \\ 5 & 0 & 8 & 0 \\ 0 & 1 & 0 & 10 \end{bmatrix}, \; c = \begin{bmatrix} -11 \\ 0 \\ 0 \\ -5 \end{bmatrix} \\
B &= \begin{bmatrix} -1 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 \\ -3 & 0 & -1 & 0 \\ -3 & 3 & 2 & -4 \\ -4 & -1 & 1 & 2 \end{bmatrix}, \; D = \begin{bmatrix} -1 \\ 4 \\ 2 \\ 0 \\ 0 \end{bmatrix}
\end{aligned}
\tag{32}
$$

We run the neural network model in simulation with a random initialization of the state variable μ and we choose $\in = 10^{-8}$. Simulations is run for 5×10^{-8} s and the simulation result of μ and x are as shown in the Figs. 2 and 3. The optimal solution is about -19.1794310722, the proposed output $x = [1.8774617068, -1.0393873085, -1.6323851203, 0.6039387308]$, the optimum point is -19.179. The differences are both less than 1.0×10^{-10} for x and the optimum point.

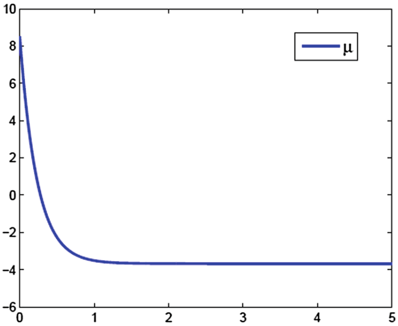

Fig. 2. Transient behavior of μ in Example V-A.

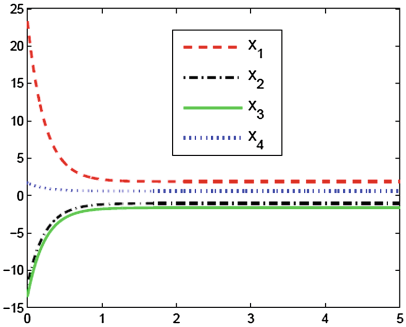

Fig. 3. Transient behavior of x in Example V-A.

5.2 Discrete-Time Model

We also consider the problem studied in Example V-A and solve it with the discrete model based on (17) and (18). In the simulation, $\gamma = 0.05$, randomly initialization μ. The evolution of μ and x are as shown in Figs. 4 and 5. The proposed network output is $x = [1.8774617068, -1.0393873085, -1.6323851203, 0.6039387308]$. The optimal value is -19.179. The difference is less than 5.85×10^{-6} for x and 4.2×10^{-5} for the optimal value.

5.3 Problem with Irredundant Equality Constraint

To show the convergence of the neural network, we solve a quadratic programming problem with irredundant equality constraints as following

$$\min(3x_1^2 + 3x_2^2 + 4x_3^2 + 5x_4^2 + 3x_1x_2 + 5x_1x_3 + x_2x_4 - 11x_1 - 5x_4)$$
$$\text{s.t.} \quad \begin{aligned} -3x_1 + 3x_2 + 2x_3 - x_4 &= 0 \\ -4x_1 - x_2 + x_3 + 2x_4 &= 0 \\ -x_1 + x_2 &\leq -1 \\ -2 \leq 3x_1 + x_3 &\leq 4 \end{aligned} \tag{33}$$

As the problem (34), A simulation is performed with a random initialization of the state variable μ and the scaling parameter $\in = 10^{-8}$. Simulation result is as shown in Figs. 6, 7 and 8, from which we can observe that the neural network converges very fast after a short transient period shorter after than 10^{-6} s. The error of solution at the final time of the simulation is less than 0.9×10^{-10} in l_2 norm sense, and the optimal solution is $x = [0.5, -0.5, -1.5, 0]$.

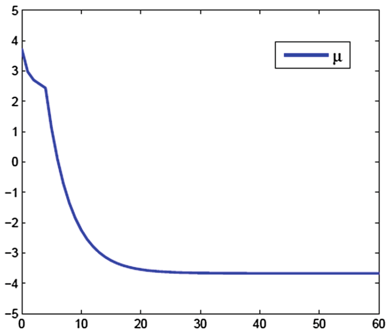

Fig. 4. Transient behavior of μ in Example V-B.

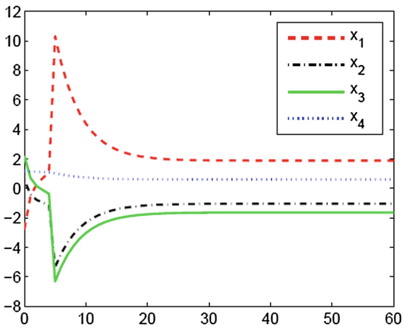

Fig. 5. Transient behavior of x in Example V-B.

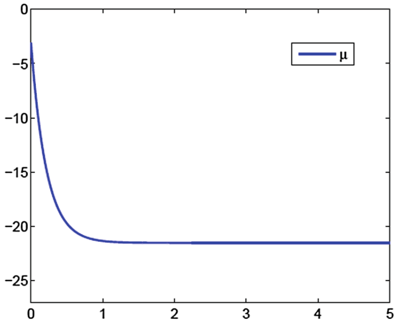

Fig. 6. Transient behavior of μ in Example V-C.

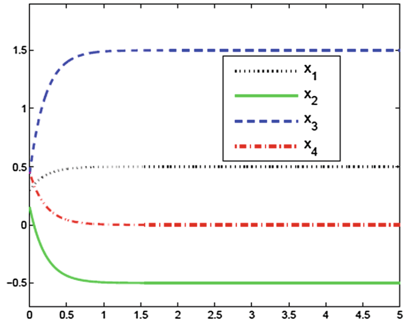

Fig. 7. Transient behavior of x in Example V-C.

$$
\begin{aligned}
W &= \begin{bmatrix} 6 & 3 & 5 & 0 \\ 3 & 6 & 0 & 1 \\ 5 & 0 & 8 & 0 \\ 0 & 1 & 0 & 10 \end{bmatrix}, \; c = \begin{bmatrix} -11 \\ 0 \\ 0 \\ -5 \end{bmatrix} \\
A &= \begin{bmatrix} 3 & -3 & -2 & 1 \\ 4 & 1 & -1 & -2 \end{bmatrix}, \; b = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
E &= \begin{bmatrix} -1 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 \\ -3 & 0 & -1 & 0 \end{bmatrix}, \; e = \begin{bmatrix} -1 \\ 4 \\ 2 \end{bmatrix}
\end{aligned}
\tag{34}
$$

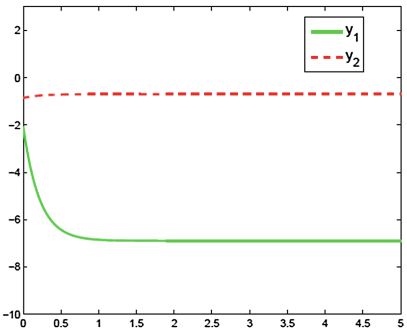

Fig. 8. Transient behavior of y in Example V-C.

6 Conclusions

For solving the general quadratic programming problems, a new recurrent neural network is proposed. The proposed model is with fewer neurons and simple structure. The neural network are shown global convergence based on contraction analysis. The discrete time counterpart to the proposed neural network is studied and an alternative recurrent neural network to solve the problem under irredundant equality constraint is also studied. Finally, simulation results demonstrate the models are efficient and accurate.

Acknowledgements. The authors would like to acknowledge the support of Guangdong Science Foundation of China under Grant No. S2011010006116 and No. 2015A030313587, Shenzhen Science Technology Project No. JCYJ20150417094158025, No. JCY20160307100530069 and GRCK20170424095924228,Shenzhen Institute of Information Technology Scientific Research Platform Cultivation Project (PT201704).

References

1. Hughes, T., Mierle, K.: Recurrent neural networks for voice activity detection. In: IEEE International Conference on Acoustics, pp. 7378–7382 (2013)
2. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673–2681 (2002)
3. Scardino, L., Niess, R.: Performance of recurrent neural networks applied to a simplified pattern recognition problem. J. Comput. Sci. Colleges **15**(3), 251–258 (2000)
4. Nan, B., Fukuda, O.: EMG-based motion discrimination using a novel recurrent neural network. J. Intell. Inf. Syst. **21**(2), 113–126 (2003)
5. Zhang, Y.N., Tan, Z.G.: Repetitive motion of redundant robots planned by three kinds of recurrent neural networks and illustrated with a four-link planar manipulator's straight-line example. Robot. Auton. Syst. **57**(6–7), 645–651 (2009)
6. Vázquez, L.A., Jurado, F.: Decentralized identification and control in real-time of a robot manipulator via recurrent wavelet first-order neural network. In: Mathematical models in Engineering, pp. 1–12 (2015)

7. Xu, R., Wunsch, D.: Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization. IEEE Trans. Comput. Biol. **4**(4), 681–692 (2007)
8. Hopfield, J.J.: Neurons with graded response have collective computational properties like those of two-state neurons. Proc. Nat. Acad. Sci. **81**(10), 3088–3092 (1984)
9. Liu, Q., Wang, J.: A one-layer recurrent neural network with a discontinuous hard-limiting activation function for quadratic programming. IEEE Trans. Neural Netw. **19**(4), 558–570 (2008)
10. Mestari, M., Namir, A.: Switched capacitor neural networks for optimal control of nonlinear dynamic systems: design and stability analysis. Syst. Anal. Model. Simul. **41**(3), 11–20 (2001)
11. Malek, A., Alipour, M.: Numerical solution for linear and quadratic programming problems using a recurrent neural network. Appl. Math. Comput. **192**(1), 27–39 (2007)
12. Liu, Q., Wang, J.: A one-layer recurrent neural network with a discontinuous activation function for linear programming. Neural Comput. **20**(5), 1366–1383 (2008)
13. 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 Trans. Neural Netw. **19** (12), 2022–2031 (2008)
14. Xia, Y., Han, Y.W.: A mixed-binary convex quadratic reformulation for box-constrained nonconvex quadratic integer program. Mathematics **10**(12), 7897–7905 (2014)
15. Li, S., Chen, S.: Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks. Neurocomputing **91**(9), 1–10 (2012)
16. Cheng, L., Hou, Z.: A Simplified Neural Network for Linear Matrix Inequality Problems. Neural Process. Lett. **29**(3), 213–230 (2009)
17. Hu, X., Zhang, B.: A simplified dual neural network for quadratic programming with its KWTA application. IEEE Trans. Neural Netw. **17**(6), 1500–1510 (2006)
18. Xia, Y., Feng, G., Wang, J.: A novel recurrent neural network for solving nonlinear optimization problems with inequality constraints. IEEE Trans. Neural Netw. **19**, 1340–1353 (2008)
19. Tymoshchuk, P.: A discrete-time dynamic K-winners-take-all neural circuit. Neurocomputing **72**(13–15), 3191–3202 (2009)
20. Xiao, Y., Liu, Y.: Analysis on the convergence time of dual neural network-based KWTA. IEEE Trans. Neural Netw. Learn. Syst. **23**(4), 676–682 (2012)
21. Hu, F., Zhang, Z.: Contraction theory-based adaptive dynamic surface control for a class of nonlinear systems. Control Decis. **31**(5), 769–775 (2016)
22. Wang, W., Slotine, J.J.E.: Contraction analysis of time-delayed communications and group cooperation. IEEE Trans. Autom. Control **51**(4), 712–717 (2006)