Neurocomputing 143 (2014) 80–89

Contents lists available at ScienceDirect

Neurocomputing

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

Letters

Finite time dual neural networks with a tunable activation function for solving quadratic programming problems and its application

Peng Miao [a], Yanjun Shen [b,*], Xiaohua Xia [c]

[a] College of Science, China Three Gorges University, Yichang, Hubei 443002, China
[b] College of Electrical Engineering and New Energy, China Three Gorges University, Yichang, Hubei 443002, China
[c] Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa

ARTICLE INFO

Article history:
Received 14 December 2013
Received in revised form
17 May 2014
Accepted 4 June 2014
Communicated by Long Cheng
Available online 21 June 2014

Keywords:
Recurrent neural networks
Finite-time stability
Tunable activation function
Quadratic programming problem

ABSTRACT

In this paper, finite time dual neural networks with a new activation function are presented to solve quadratic programming problems. The activation function has two tunable parameters, which give more flexibility to design the neural networks. By Lyapunov theorem, finite-time stability can be derived for the proposed neural networks, and the actual optimal solutions of the quadratic programming problems can be obtained in finite time interval. Different from the existing recurrent neural networks for solving the quadratic programming problems, the neural networks of this paper have a faster convergent speed, at the same time, they can reduce oscillation when delay appears, and have less sensitivity to additive noise with careful selection of the parameters. Simulations are presented to evaluate the performance of the neural networks with the tunable activation function. In addition, the proposed neural networks are applied to estimate parameters for an energy model of belt conveyors. The effectiveness of our methods are validated by theoretical analysis and numerical simulations.

1. Introduction

Recently, recurrent neural networks have made great development. They are widely applied in scientific and engineering field, for example, optimization [1,2], control of chaos [3], pattern classification [4,5], signal processing [6], robotics [7], solving time-varying Sylvester equation [8], the winners-take-all competition [9–11], etc.

With the development of recurrent neural networks, remarkable advances have been made in the field of online optimization. For example, by removing the explicit constraints and by introducing a penalty term into the cost function, recurrent neural networks are designed to solve the constrained optimization problem in [12–14]. However, the designed neural networks only converge to the optimal solution asymptotically and the convergence time is infinite. In order to obtain the accurate solution, in [15] and [16], dynamic Lagrange multipliers are introduced to regulate the constraints and the optimal solution can be obtained in finite time. However, the number of neurons in the neural networks is increased. The reason is that extra neurons are required for the dynamics of the Lagrange multipliers. It is well known that the complexity and cost of its hardware implementation are relevant to the number of neurons in neural networks. Then, research on reduction of neuron number without losing efficiency and accuracy receives some researchers' attention [17–33]. For examples, Zhang investigated and analyzed the performance of gradient neural network applied to time-varying quadratic minimization and quadratic programming problems in [20]. Wang presented a k-winners-take-all (kWTA) neural network and proved its global stability and finite-time convergence in [23]. Bian and Chen proposed a smoothing neural network with a differential equation which could be implemented easily in [32]. There are also some neural network models for solving optimization problems, which can efficiently deal with time delays [34–38]. For instance, Liu and Cao proposed a delayed neural network which could effectively solve a class of linear projection equations and some quadratic programming problems in [34]. In order to deal with the convex optimization problem in finite time, the authors firstly presented a recurrent neural network with a continuous function, $|x|^r \operatorname{sign}(x)(0<r<1)$ in [39]. The activation function was also applied to a dual neural network model in [25]. The finite-time convergence property and the optimality of the proposed neural network for solving the quadratic programming problem are proven. The parameter r has an effect on the convergence time. The neural network has a faster convergent speed with a smaller r. However, chattering phenomenon will happen, especially in the case when time delay appears. On the other hand, the neural network with a smaller r is less sensitive to

* Corresponding author.
E-mail address: shenyj@ctgu.edu.cn (Y. Shen).

http://dx.doi.org/10.1016/j.neucom.2014.06.018
0925-2312/

additive noise. Therefore, it is worth while to study finite-time dual neural network for solving quadratic programming problems with a relative high robustness against time delay and noise.

In the paper, our main contribution is to present recurrent neural networks with a tunable activation function to solve quadratic programming problems. The tunable activation function is $k_1|x|^r \operatorname{sign}(x)+k_2x$, where k_1, k_2 are tunable positive parameters. These parameters are not only helpful to accelerate convergence speed, but also helpful to improve robustness of the neural networks with appearance of time delay and noise. The motivation comes from the idea of Gao and Hung in [43]. Adding the term $k_2x(t)$ to the activation function is a standard technique in sliding mode control. It can be used to suppress chattering [43].

The paper is organized as follows. In Section 2, finite-time criteria and upper bounds of the convergence time are reviewed. In Section 3, we present finite-time recurrent neural networks with a tunable activation function for solving quadratic programming problems. In Section 4, numerical simulations are given to show the effectiveness of our methods. Section 5 concludes the paper.

2. Preliminaries

Consider the following system:

$$\dot{x}(t)=f(x(t)),\quad f(0)=0,\quad x\in\mathcal{R}^n,\quad x(0)=x_0, \tag{1}$$

where $f:\mathcal{D}\rightarrow\mathcal{R}^n$ is continuous on an open neighborhood $\mathcal{D}$ of the origin $x=0$.

Definition 1 (*Bhat and Bernstein [40]*). The equilibrium $x=0$ of (1) is finite-time convergent if there are an open neighborhood $\mathcal{U}$ of the origin and a function $T_x:\mathcal{U}\backslash\{0\}\rightarrow(0,\infty)$, such that every solution trajectory $x(t,x_0)$ of (1) starting from the initial point $x_0\in\mathcal{U}\backslash\{0\}$ is well-defined and unique in forward time for $t\in[0,T_x(x_0))$, and $\lim_{t\rightarrow T_x(x_0)}x(t,x_0)=0$. Then, $T_x(x_0)$ is called the convergence time (of the initial state x_0). The equilibrium of (1) is finite-time stable if it is Lyapunov stable and finite-time convergent. If $U=\mathcal{D}=\mathcal{R}^n$, the origin is a globally finite-time stable equilibrium.

The following Lemmas provide sufficient conditions for the origin of the system (1) to be a finite-time stable equilibrium.

Lemma 1 (*Bhat and Bernstein [40]*). *Suppose there are a C^1 positive definite function $V(x)$ defined on a neighborhood $\mathcal{U}\subset\mathcal{R}^n$ of the origin, and real numbers $k_1>0$ and $0<r<1$, such that*

$$\dot{V}(x)|_{(1)}\le -k_1V(x)^r,\quad \forall x\in\mathcal{U}. \tag{2}$$

Then, the origin of the system (1) is locally finite-time stable. The convergence time T_1, depending on the initial state x_0, satisfies

$$T_1(x_0)\le\frac{V(x_0)^{1-r}}{k_1(1-r)}, \tag{3}$$

for all $x_0\in\mathcal{U}$. Further, if $\mathcal{U}=\mathcal{R}^n$ and $V(x)$ is radially unbounded (that is $V(x)\rightarrow+\infty$ as $\|x\|\rightarrow+\infty$), the origin of system (1) is globally finite-time stable.

Lemma 2 (*Shen and Xia [41], Shen and Huang [42]*). *If there are a C^1 positive definite function $V(x)$ defined on a neighborhood $\mathcal{U}\subset\mathcal{R}^n$ of the origin, and real numbers $k_1,k_2>0$ and $0<r<1$, such that*

$$\dot{V}(x)|_{(1)}\le -k_1V(x)^r-k_2V(x),\quad \forall x\in\mathcal{U}. \tag{4}$$

Then, the origin of system (1) is finite-time stable. The convergence time T_2 satisfies

$$T_2(x_0)\le\frac{\ln\left[1+\dfrac{k_2}{k_1}V(x_0)^{1-r}\right]}{k_2(1-r)}, \tag{5}$$

for all $x_0\in\mathcal{U}$. If $\mathcal{U}=\mathcal{R}^n$ and $V(x)$ is radially unbounded, the origin of system (1) is globally finite-time stable.

Remark 1. From Lemmas 1 and 2, we can see the upper bound of the convergence time is relevant to r. It decreases with decrease of r. When r is greater than 0 but sufficiently close to 0, the term $|x|^r \operatorname{sign}(x)$ is very close to $\operatorname{sign}(x)$ for x with small absolute values. Therefore, it may yield chattering phenomenon. For example, consider the following scalar differential equation:

$$\dot{x}(t)=-|x(t)|^r \operatorname{sign}(x(t)),\quad 0<r<1. \tag{6}$$

The trajectories of (6) with different value of r are given in Fig. 1.

From Fig. 1, we can see that the trajectory of (6) with $r=0.2$ has the fastest convergence speed, but the chatting phenomenon happens also. To overcome the problem, we can select a small value of k_1 and a large value of k_2 for the following scalar differential equation:

$$\dot{x}(t)=-k_1|x(t)|^r \operatorname{sign}(x(t))-k_2x(t),\quad 0<r<1. \tag{7}$$

Fig. 2 shows the trajectory of (7). From Fig. 2, we can see the chattering phenomenon disappears.

In sliding mode control, the introduction of k_2x, called reaching control, to suppress chattering is the idea of Gao and Hung in [43].

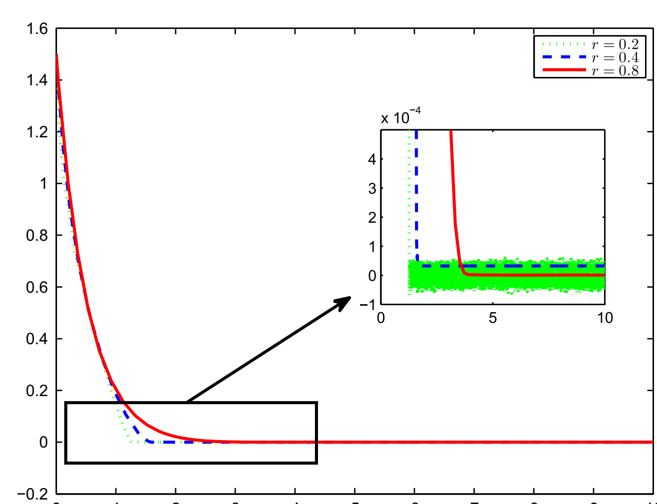

Fig. 1. The trajectories of (6) with different value of r.

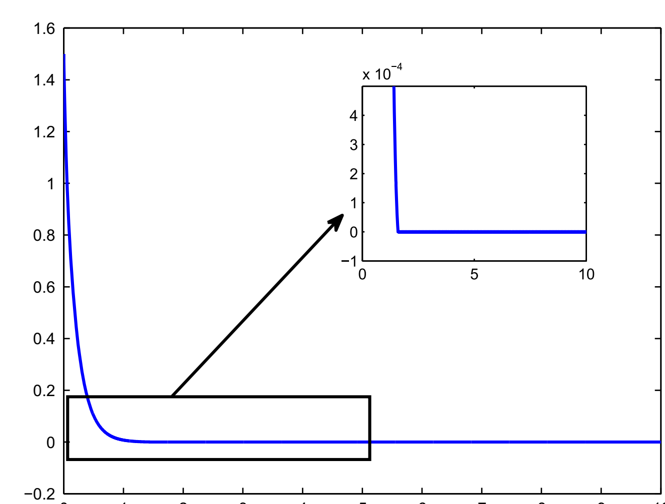

Fig. 2. The trajectory of (7) with $r=0.2$, $k_1=0.0001$, $k_2=15$.