

A one-layer recurrent neural network for constrained pseudoconvex optimization and its application for dynamic portfolio optimization[☆]

Qingshan Liu^a, Zhishan Guo^b, Jun Wang^{c,*}

^a School of Automation, Southeast University, Nanjing 210096, China

^b Department of Computer Science, University of North Carolina - Chapel Hill, Chapel Hill, NC 27599, USA

^c Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong

ARTICLE INFO

Article history:

Received 26 June 2011

Accepted 1 September 2011

Keywords:

Recurrent neural networks
Pseudoconvex optimization
Differential inclusion
Convergence
Lyapunov function

ABSTRACT

In this paper, a one-layer recurrent neural network is proposed for solving pseudoconvex optimization problems subject to linear equality and bound constraints. Compared with the existing neural networks for optimization (e.g., the projection neural networks), the proposed neural network is capable of solving more general pseudoconvex optimization problems with equality and bound constraints. Moreover, it is capable of solving constrained fractional programming problems as a special case. The convergence of the state variables of the proposed neural network to achieve solution optimality is guaranteed as long as the designed parameters in the model are larger than the derived lower bounds. Numerical examples with simulation results illustrate the effectiveness and characteristics of the proposed neural network. In addition, an application for dynamic portfolio optimization is discussed.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

Constrained optimization problems arise in widespread scientific and engineering applications, such as filter design, signal processing, system identification, robot control, and power system planning.

In the past decades, many recurrent neural networks were developed for solving constrained optimization problems (e.g., see Tank & Hopfield, 1986; Wang, 1993; Kennedy & Chua, 1988; Xia, H, & J, 2002; Forti, Nistri, & Quincampoix, 2004, and references therein). Among them, based on the penalty functions and gradient methods, constrained optimization problems were first converted approximately or exactly to unconstrained optimization problems, then some gradient-based neural network models were constructed to compute the approximate or exact optimal solutions. Two classical recurrent neural network models for optimization are the Hopfield neural network proposed by Tank and Hopfield (1986) for linear programming, and that by Kennedy and Chua (1988) for nonlinear programming. The gradient method was widely used in the neural network design. Based on the l_2 -norm

penalty function and gradient method, a neural network model was developed for constrained optimization (Lillo, Loh, Hui, & Zak, 1993). The l_p -norm ($1 \leq p \leq \infty$) penalty functions were used to derive a class of neural networks for linear programming (Chong, Hui, & Zak, 1999).

In addition to the gradient method, other methods have been developed for neurodynamic optimization. Based on the Lagrangian function and Lagrangian optimality conditions, the Lagrangian network (Zhang & Constantinides, 1992) was proposed for solving the general optimization problems, in which the local convergence was guaranteed. For convex optimization, the global convergence of the Lagrangian network was analyzed and proven by Xia (2003). The deterministic annealing neural networks were developed for solving linear and nonlinear convex programming problems by Wang (1993, 1994), which were utilized to solve the assignment problems (Wang, 1997) and shortest path problems (Wang, 1998).

In recent years, based on the Karush–Kuhn–Tucker optimality conditions, the primal–dual network (Xia, 1996), dual network (Xia, Feng, & Wang, 2004) and simplified dual network (Liu & Wang, 2006) were developed for solving convex optimization problems. Based on the projection method (e.g., see Xia et al., 2002; Hu & Wang, 2007; Liu & Cao, 2010, and references therein), optimality conditions for constrained optimization problems can be written in the form of linear (or nonlinear) variational inequalities, and transformed into projection equations. Then neural networks based on the projection equations were constructed for solving the constrained optimization problems. Moreover, for convex optimization problems, the global convergence of the projection neural

[☆] The work described in the paper was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Grants CUHK417209E and CUHK417608E, and by the National Natural Science Foundation of China under Grant 61105060.

* Corresponding author.

E-mail addresses: qslu@seu.edu.cn (Q. Liu), zsguo@cs.unc.edu (Z. Guo), jwang@mae.cuhk.edu.hk (J. Wang).

networks can be guaranteed for the global optimal solutions (Xia & Wang, 2004b; Hu & Wang, 2007a; Barbarosou & Maratos, 2008). To reduce the model complexity, some one-layer recurrent neural networks were proposed for solving linear and quadratic programming problems (Xia & Wang, 2004a; Liu & Wang, 2008a, 2008b).

Apart from the recurrent neural networks for solving smooth constrained optimization problems, neurodynamic approaches to nonsmooth constrained optimization were investigated by some researchers recently. The generalized nonlinear programming circuit for solving nonsmooth nonconvex optimization problems were proposed by Forti et al. (2004); Forti, Nistri, and Quincampoix (2006). We proposed some neural networks with simple model complexity for solving the constrained nonsmooth convex optimization problems (Liu & Wang, 2006a, 2009). The subgradient-based neural networks (Xue & Bian, 2008; Bian and Xue, 2009) were proposed for solving nonsmooth optimization problems with more general constraints than that by Forti et al. (2004). Inspired by the work in Forti et al. (2004); Forti et al. (2006), Xue and Bian (2008) and Bian and Xue (2009), this paper contributes to present a one-layer recurrent neural network with discontinuous activation functions for solving pseudoconvex optimization problems subject to linear equality and bound constraints.

While most neural network approaches to optimization focus on convex optimization, nonconvex optimization is rarely investigated. In particular, among nonconvex optimization problems, pseudoconvex optimization has many applications, such as fractional programming, computer vision (Olsson, Eriksson, & Kahl, 2007), production planning, financial and corporate planning, health-care and hospital planning. Hu and Wang (2006) extended the projection neural network for optimization with differentiable pseudoconvex objective functions and bound constraints. However, for more general pseudoconvex optimization problems subject to linear equality and bound constraints, the projection neural network is not applicable for solving these problems due to its convergence conditions.

In this paper, a recurrent neural network will be presented for solving pseudoconvex optimization problems with linear equality and bound constraints. The remainder of this paper is organized as follows. Section 2 discusses some preliminaries. In Section 3, the problem formulation and neural network model are described. The theoretical analysis of the proposed neural network is shown in Section 4. Illustrative Examples are given to show the effectiveness and performance of the proposed neural network in Section 5. Next, in Section 6, the proposed neural network is utilized for dynamic portfolio optimization. Finally, Section 7 concludes this paper.

2. Preliminaries

For the convenience of later discussions, we present some definitions and properties concerning set-valued map, nonsmooth analysis and pseudoconvex functions in this section. We refer readers to Clarke (1983), Aubin and Cellina (1984), Filippov (1988) and Forti et al. (2006) for more thorough discussions. Throughout this paper, $\|\cdot\|_1$ and $\|\cdot\|_2$ denote the l_1 and l_2 norms of a vector (or matrix) in \mathbb{R}^n (or $\mathbb{R}^{m \times n}$), respectively.

2.1. Nonsmooth analysis

Definition 1. Suppose $E \subset \mathbb{R}^n$. $F : x \mapsto F(x)$ is called a set-valued map from $E \subset \mathbb{R}^n$ to \mathbb{R}^m , if to each point x of the set E , there corresponds a nonempty closed set $F(x) \subset \mathbb{R}^m$.

Definition 2. A function $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be Lipschitz near $x \in \mathbb{R}^n$ if there exist $\varepsilon, \delta > 0$, such that for any $x', x'' \in \mathbb{R}^n$ satisfying $\|x' - x\|_2 < \delta$ and $\|x'' - x\|_2 < \delta$, we have $|\varphi(x') - \varphi(x'')| \leq \varepsilon \|x' - x''\|_2$. If φ is Lipschitz near any point $x \in \mathbb{R}^n$, then φ is also said to be locally Lipschitz in \mathbb{R}^n .

Assume that φ is Lipschitz near x . The generalized directional derivative of φ at x in the direction $v \in \mathbb{R}^n$ is given by

$$\varphi^0(x; v) = \limsup_{\substack{y \rightarrow x \\ s \rightarrow 0^+}} \frac{\varphi(y + sv) - \varphi(y)}{s}.$$

The Clarke's generalized gradient of f is defined as

$$\partial\varphi(x) = \{y \in \mathbb{R}^n : \varphi^0(x; v) \geq y^T v, \forall v \in \mathbb{R}^n\}.$$

When φ is locally Lipschitz in \mathbb{R}^n , φ is differentiable for almost all (a.a.) $x \in \mathbb{R}^n$ (in the sense of Lebesgue measure). Then, the Clarke's generalized gradient of φ at $x \in \mathbb{R}^n$ is equivalent to

$$\partial\varphi(x) = K \left\{ \lim_{n \rightarrow \infty} \nabla\varphi(x_n) : x_n \rightarrow x, x_n \notin \mathcal{N}, x_n \notin E \right\},$$

where $K(\cdot)$ denotes the closure of the convex hull, $\mathcal{N} \subset \mathbb{R}^n$ is an arbitrary set with measure zero, and $E \subset \mathbb{R}^n$ is the set of points where φ is not differentiable.

Definition 3. A function $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}$, which is locally Lipschitz near $x \in \mathbb{R}^n$, is said to be regular at x if there exists the one-sided directional derivative for any direction $v \in \mathbb{R}^n$ which is given by

$$\varphi'(x; v) = \lim_{\xi \rightarrow 0^+} \frac{\varphi(x + \xi v) - \varphi(x)}{\xi},$$

and we have $\varphi^0(x; v) = \varphi'(x; v)$. The function φ is said to be regular in \mathbb{R}^n if it is regular for any $x \in \mathbb{R}^n$.

Regular functions are very important in the Lyapunov approach and nonsmooth analysis used in this paper, which has been studied in the literature (e.g., see Clarke, 1983; Filippov, 1988). In particular, a nonsmooth convex function on \mathbb{R}^n is regular at any $x \in \mathbb{R}^n$. For a finite family of functions $\varphi_i (i = 1, 2, \dots, n)$, which are regular at x , we have $\partial(\sum_{i=1}^n \varphi_i)(x) = \sum_{i=1}^n \partial\varphi_i(x)$.

Consider the following ordinary differential equation (ODE):

$$\frac{dx}{dt} = \psi(x), \quad x(t_0) = x_0. \quad (1)$$

A set-valued map defined as

$$\phi(x) = \bigcap_{\varepsilon > 0} \bigcap_{\mu(\mathcal{N})=0} K[\psi(B(x, \varepsilon) - \mathcal{N})],$$

where $\mu(\mathcal{N})$ is the Lebesgue measure of set \mathcal{N} , $B(x, \varepsilon) = \{y : \|y - x\|_2 \leq \varepsilon\}$. A solution of (1) is an absolutely continuous function $x(t)$ defined on an interval $[t_0, t_1] (t_0 \leq t_1 \leq +\infty)$, which satisfies $x(t_0) = x_0$ and differential inclusion:

$$\frac{dx}{dt} \in \phi(x), \quad \text{a.a. } t \in [t_0, t_1].$$

In the regular case, the following chain rule is of key importance in the Lyapunov approach used in this paper.

Lemma 1 (Chain Rule Clarke (1983)). If $V : \mathbb{R}^n \rightarrow \mathbb{R}$ is regular at $x(t)$ and $x(t) : \mathbb{R} \rightarrow \mathbb{R}^n$ is differentiable at t and Lipschitz near t , then

$$\frac{d}{dt} V(x(t)) = \xi^T \dot{x}, \quad \forall \xi \in \partial V(x(t)).$$

2.2. Normal cone

Definition 4. Suppose that $E \subset \mathbb{R}^n$ is a nonempty closed convex set. The normal cone to the set E as $x \in E$ is defined as

$$N_E(x) = \{v \in \mathbb{R}^n : v^T(x - y) \geq 0, \forall y \in E\}.$$

Lemma 2 (Clarke, 1983). If $E_1, E_2 \subset \mathbb{R}^n$ are closed convex sets and satisfy $0 \in \text{int}(E_1 - E_2)$, then for any $x \in E_1 \cap E_2$, $N_{E_1 \cap E_2}(x) = N_{E_1}(x) + N_{E_2}(x)$, where $\text{int}(\cdot)$ denotes the interior of the set.

Lemma 3 (Clarke, 1983). Suppose that f is Lipschitz near x and attains a minimum over \mathcal{S} at x , then $0 \in \partial f(x) + N_{\mathcal{S}}(x)$.

2.3. Pseudoconvex and pseudomonotone

Definition 5 (Penot & Quang, 1997). Let $E \subset \mathbb{R}^n$ be a nonempty convex set. A function $f : E \rightarrow \mathbb{R}$ is said to be pseudoconvex on E if, for every pair of distinct points $x, y \in E$

$$\exists \eta \in \partial f(x) : \eta^T(y - x) \geq 0 \implies f(y) \geq f(x).$$

Definition 6 (Penot & Quang, 1997). Let $E \subset \mathbb{R}^n$ be a nonempty convex set. A set-valued map $F : E \rightarrow \mathbb{R}^m$ is said to be pseudomonotone on E if, for every pair of distinct points $x, y \in E$

$$\exists \eta_x \in F(x) : \eta_x^T(y - x) \geq 0 \implies \forall \eta_y \in F(y) : \eta_y^T(y - x) \geq 0.$$

It is shown in Penot and Quang (1997) that a continuous function $f(x)$ is pseudoconvex if and only if its generalized gradient $\partial f(x)$ is a pseudomonotone mapping.

3. Problem formulation and model description

In this paper, we are concerned with the following nonlinear programming problem:

$$\begin{aligned} & \text{minimize} && f(x), \\ & \text{subject to} && Ax = b, \\ & && u \leq x \leq v, \end{aligned} \quad (2)$$

where $x = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$, $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is assumed to be locally Lipschitz continuous, $A \in \mathbb{R}^{m \times n}$ which is full row-rank (i.e., $\text{rank}(A) = m \leq n$), $b \in \mathbb{R}^m$, $u = (u_1, u_2, \dots, u_n)^T \in \mathbb{R}^n$ and $v = (v_1, v_2, \dots, v_n)^T \in \mathbb{R}^n$.

For a given $x \in \mathbb{R}^n$, let $\mathcal{D}(x) = \sum_{i=1}^n d(x_i)$ with $d(x_i)$ ($i = 1, 2, \dots, n$) defined as

$$d(x_i) = \begin{cases} x_i - v_i, & x_i \geq v_i, \\ 0, & u_i < x_i < v_i, \\ -x_i + u_i, & x_i \leq u_i. \end{cases} \quad (3)$$

Then $d(x_i) \geq 0$ and $\mathcal{D}(x) \geq 0$. We define that

$$I_+(x) = \{i \in \{1, 2, \dots, n\} : d(x_i) > 0\},$$

$$I_0(x) = \{i \in \{1, 2, \dots, n\} : d(x_i) = 0\}.$$

The region where the constraints are satisfied (feasible region) is defined as $\mathcal{S} = \{x \in \mathbb{R}^n : Ax = b, u \leq x \leq v\}$. Moreover, we define that $\mathcal{C} = \{x \in \mathbb{R}^n : Ax = b\}$. Then, it is clear that $\mathcal{S} = \mathcal{C} \cap [u, v]$, where $[u, v] = \{x \in \mathbb{R}^n : u \leq x \leq v\}$.

Throughout this paper, the objective function $f(x)$ is not necessary to be convex or smooth, whereas the following two assumptions on the optimization problem (2) are needed.

Assumption 1. There exist $\hat{x} \in \mathbb{R}^n$ and $r > 0$ such that $\hat{x} \in \text{int}([u, v]) \cap \mathcal{C}$ and $[u, v] \subset B(\hat{x}, r)$, where $B(\hat{x}, r) = \{x \in \mathbb{R}^n : \|x - \hat{x}\|_2 \leq r\}$ is the r neighborhood of \hat{x} .

Assumption 2. The objective function $f(x)$ of problem (2) is pseudoconvex and regular on \mathcal{S} , and Lipschitz bounded on $B(\hat{x}, r)$.

Remark 1. In Assumption 2, the objective function is assumed to be pseudoconvex and regular on \mathcal{S} . For a pseudoconvex function,

several classes of them are regular. For example, let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be Lipschitz on \mathcal{S} , then, (i) if f is strictly differentiable on \mathcal{S} , it is regular on \mathcal{S} ; (ii) if f is convex on \mathcal{S} , it is regular on \mathcal{S} . Consequently, problem (2) includes the pseudoconvex optimization problems investigated by Hu and Wang (2006) as its special case, and also includes the smooth and nonsmooth convex optimization problems investigated by Liu and Wang (2009) and Xia and Wang (2005) as its special cases. Therefore, the optimization problem (2) under the above assumptions includes a larger part of optimization problems.

The dynamic equation of the proposed recurrent neural network model for solving (2) is described in the following differential inclusion:

$$\epsilon \frac{dx}{dt} \in -\partial f(x) - \mu g_{[u,v]}(x) - \sigma A^T g_0(Ax - b), \quad (4)$$

where ϵ is a positive scaling constant, σ and μ are nonnegative constants, $\partial f(x)$ is the generalized gradient of $f(x)$, $g_{[u,v]}$ is a discontinuous activation function with its components are defined as

$$g_{[u,v]}(y) = \begin{cases} 1, & y > v_i, \\ [0, 1], & y = v_i, \\ 0, & u_i < y < v_i, \\ [-1, 0], & y = u_i, \\ -1, & y < u_i, \end{cases} \quad (i = 1, 2, \dots, n) \quad (5)$$

and its special case for $u = v = 0$ is defined as

$$g_0(y) = \begin{cases} 1, & y > 0, \\ [-1, 1], & y = 0, \\ -1, & y < 0. \end{cases} \quad (6)$$

Note that $g_0(y)$ and $g_{[u,v]}(y)$ are discontinuous at $y = 0$ and $y = u_i$ or v_i , respectively.

Remark 2. One of the important classes of pseudoconvex optimization problems is the quadratic fractional programming problems in the following form:

$$\begin{aligned} & \text{minimize} && f(x) = \frac{x^T Q x + a^T x + a_0}{c^T x + c_0}, \\ & \text{subject to} && Ax = b, \\ & && u \leq x \leq v, \end{aligned} \quad (7)$$

where $Q \in \mathbb{R}^{n \times n}$ is a symmetric matrix, $a, c \in \mathbb{R}^n$, $a_0, c_0 \in \mathbb{R}$, A, b, u and v are defined in (2). It is well known that f is pseudoconvex on $X = \{x \in \mathbb{R}^n : c^T x + c_0 > 0\}$ if Q is positive semidefinite. Specially, when $c = 0$, (7) is reduced to the classic quadratic programming problem, and when $Q = 0$, it is reduced to the so called linear fractional problem, which is, of course, pseudoconvex on X .

4. Theoretical analysis

In this section, the convergence and optimality conditions of the proposed neural network are investigated.

Definition 7. \bar{x} is said to be an equilibrium point of system (4) if

$$0 \in \partial f(\bar{x}) + \sigma A^T K[g_0(A\bar{x} - b)] + \mu K[g_{[u,v]}(\bar{x})]; \quad (8)$$

i.e., if there exist $\bar{\eta} \in \partial f(\bar{x})$, $\bar{\gamma} \in K[g_0(A\bar{x} - b)]$, and $\bar{\xi} \in K[g_{[u,v]}(\bar{x})]$ such that

$$\bar{\eta} + \sigma A^T \bar{\gamma} + \mu \bar{\xi} = 0. \quad (9)$$

4.1. Boundedness of the state vector $x(t)$

In this subsection, the boundedness of the state vector of neural network (4) is proven. According to Assumption 2, $f(x)$ is Lipschitz

bounded on $B(\hat{x}, r)$. Throughout this paper, we denote l_f as an upper bound of Lipschitz constant of $f(x)$ on $B(\hat{x}, r)$. First, it is easy to get the following lemma.

Lemma 4. For any $x \in \mathbb{R}^n$, $Ax = b$ if and only if $Px = q$, where $P = A^T(AA^T)^{-1}A$ and $q = A^T(AA^T)^{-1}b$.

Next, inspired by the work in Xue and Bian (2008) and Bian and Xue (2009), the following two lemmas are given.

Lemma 5. Suppose Assumption 1 holds. For any $x \in \mathbb{R}^n \setminus [u, v]$ and $\xi \in K[g_{[u,v]}(x)]$, $(x - \hat{x})^T \xi > \omega$, where $\omega = \min_{1 \leq i \leq n} \{v_i - \hat{x}_i, \hat{x}_i - u_i\}$, and \hat{x}_i is the i th element of \hat{x} .

Proof. From the definition of $g_{[u,v]}(x)$, for any $x \in \mathbb{R}^n$ and $\xi \in K[g_{[u,v]}(x)]$, we have $(x_i - \hat{x}_i)\xi_i \geq 0$ ($i = 1, 2, \dots, n$), where x_i, \hat{x}_i and ξ_i are the i th elements of x, \hat{x} and ξ respectively. For any $i \in I_+(x)$, one gets that $(x_i - \hat{x}_i)\xi_i > \min\{v_i - \hat{x}_i, \hat{x}_i - u_i\}$. Since for any $x \in \mathbb{R}^n \setminus [u, v]$, we have $I_+(x) \neq \emptyset$, then $(x - \hat{x})^T \xi = \sum_{i=1}^n (x_i - \hat{x}_i)\xi_i \geq \sum_{i \in I_+(x)} (x_i - \hat{x}_i)\xi_i > \min_{1 \leq i \leq n} \{v_i - \hat{x}_i, \hat{x}_i - u_i\}$. \square

Lemma 6. Suppose Assumption 1 holds. For any $x \in (B(\hat{x}, r) \cap \mathcal{C}) \setminus [u, v]$ and $\xi \in K[g_{[u,v]}(x)]$, $\|(I - P)\xi\|_2 > \omega/r$, where I is the identity matrix and ω is defined in Lemma 5.

Proof. From Lemma 4, for any $x \in \mathcal{C}$, $Px = P\hat{x} = q$. According to Lemma 5, for any $x \in (B(\hat{x}, r) \cap \mathcal{C}) \setminus [u, v]$, we have $(x - \hat{x})^T(I - P)\xi = (x - \hat{x})^T \xi > \omega$, where $\xi \in K[g_{[u,v]}(x)]$. Since $(x - \hat{x})^T(I - P)\xi \leq \|x - \hat{x}\|_2 \|(I - P)\xi\|_2 \leq r\|(I - P)\xi\|_2$, it follows that $\|(I - P)\xi\|_2 > \omega/r$. \square

The boundedness of the state vector of neural network (4) is stated as the following theorem.

Theorem 1. Suppose that Assumptions 1 and 2 hold. For any $x_0 \in B(\hat{x}, r)$, the state vector of neural network (4) satisfies $x(t) \in B(\hat{x}, r)$ if $\mu > r l_f / \omega$.

Proof. Let $\rho(x(t)) = \epsilon \|x(t) - \hat{x}\|_2^2 / 2$. For any $x_0 \in B(\hat{x}, r)$, there exist $\eta \in \partial f(x)$, $\gamma \in K[g_0(Ax - b)]$ and $\xi \in K[g_{[u,v]}(x)]$ such that

$$\begin{aligned} \frac{d}{dt} \rho(x(t)) &= \epsilon (x(t) - \hat{x})^T \dot{x}(t) \\ &= (x(t) - \hat{x})^T (-\eta - \sigma A^T \gamma - \mu \xi) \\ &= (x(t) - \hat{x})^T (-\eta - \mu \xi) - \sigma (Ax(t) - b)^T \gamma, \end{aligned}$$

in which the last equality holds since $A\hat{x} = b$. From the definition of g in (6), $(Ax(t) - b)^T \gamma \geq 0$ for any $\gamma \in K[g_0(Ax - b)]$. Then we have

$$\begin{aligned} \frac{d}{dt} \rho(x(t)) &\leq (x(t) - \hat{x})^T (-\eta - \mu \xi) \\ &\leq \|x(t) - \hat{x}\|_2 \|\eta\|_2 - \mu (x(t) - \hat{x})^T \xi. \end{aligned}$$

If $x(t) \notin [u, v]$, according to Lemma 5, one gets that $(x(t) - \hat{x})^T \xi > \omega$. Then we have

$$\frac{d}{dt} \rho(x(t)) < \|x(t) - \hat{x}\|_2 \|\eta\|_2 - \mu \omega. \quad (10)$$

If $\mu > r l_f / \omega$, we say that $x(t) \in B(\hat{x}, r)$. If not so, the state $x(t)$ leaves $B(\hat{x}, r)$ at time t_1 , and we have $\|x(t_1) - \hat{x}\|_2 = r$. Then we have $d\rho(x(t))/dt|_{t=t_1} \geq 0$. From (10), combining that $\|\eta\|_2 \leq l_f$, we have

$$\left. \frac{d}{dt} \rho(x(t)) \right|_{t=t_1} < r l_f - \mu \omega < 0,$$

which is a contradiction.

If $x(t) \in [u, v]$, then we have $x(t) \in B(\hat{x}, r)$ directly.

Consequently, for any $x_0 \in B(\hat{x}, r)$, the state vector of neural network (4) satisfies $x(t) \in B(\hat{x}, r)$ if $\mu > r l_f / \omega$. \square

4.2. Finite-time convergence to \mathcal{C}

In this subsection, we prove that the state vector of the neural network (4) reaches the equality feasible region \mathcal{C} in finite time and stays there thereafter.

Theorem 2. Suppose that Assumptions 1 and 2 hold. For any $x_0 \in B(\hat{x}, r)$, the state vector of neural network (4) reaches the equality feasible region \mathcal{C} in finite time and stays there thereafter, if $\sigma > (l_f + \sqrt{n}\mu) / \sqrt{\lambda_{\min}(AA^T)}$ and $\mu > r l_f / \omega$, where λ_{\min} is the minimum eigenvalue of the matrix.

Proof. Let $\mathcal{B}(x) = \epsilon \|Ax - b\|_1$. According to the chain rule, we have

$$\frac{d}{dt} \mathcal{B}(x) = \zeta^T \dot{x}(t), \quad \forall \zeta \in \partial \mathcal{B}(x(t)).$$

From Theorem 1, for any $x_0 \in B(\hat{x}, r)$, $x(t) \in B(\hat{x}, r)$ if $\mu > r l_f / \omega$. When $x \in B(\hat{x}, r) \setminus \mathcal{C}$, there exist $\eta \in \partial f(x)$, $\gamma \in K[g_0(Ax - b)]$ and $\xi \in K[g_{[u,v]}(x)]$ such that

$$\begin{aligned} \frac{d}{dt} \mathcal{B}(x) &= \gamma^T A (-\eta - \sigma A^T \gamma - \mu \xi) \\ &\leq \|A^T \gamma\|_2 (\|\eta\|_2 + \mu \|\xi\|_2 - \sigma \|A^T \gamma\|_2) \end{aligned}$$

For any $\eta \in \partial f(x)$, we have $\|\eta\|_2 \leq l_f$. For any $\xi \in K[g_{[u,v]}(x)]$, from the definition of $g_{[u,v]}(x)$, it follows that $\|\xi\|_2 \leq \sqrt{n}$. For any $x \in B(\hat{x}, r) \setminus \mathcal{C}$ and $\gamma \in K[g_0(Ax - b)]$, since $\epsilon A^T \gamma \in \partial \mathcal{B}(x)$, we have $A^T \gamma \neq 0$ and at least one of the components of γ is -1 or 1 . On one hand, since A has full row-rank, AA^T is invertible. It follows that

$$\|(AA^T)^{-1} AA^T \gamma\|_2 = \|\gamma\|_2 \geq 1.$$

On the other hand, we have

$$\|(AA^T)^{-1} AA^T \gamma\|_2 \leq \|(AA^T)^{-1} A\|_2 \|A^T \gamma\|_2.$$

Since AA^T is positive definite, we have

$$\begin{aligned} \|(AA^T)^{-1} A\|_2 &= \sqrt{\lambda_{\max}[(AA^T)^{-1} A (AA^T)^{-1} A^T]} \\ &= \sqrt{\lambda_{\max}((AA^T)^{-1})} \\ &= \frac{1}{\sqrt{\lambda_{\min}(AA^T)}} > 0, \end{aligned}$$

where λ_{\min} and λ_{\max} are the minimum and maximum eigenvalues of the matrices respectively. It follows that

$$\|A^T \gamma\|_2 \geq \sqrt{\lambda_{\min}(AA^T)}.$$

Let $\alpha = \sqrt{\lambda_{\min}(AA^T)}$. If $\sigma > (l_f + \sqrt{n}\mu) / \alpha$, we have

$$\frac{d}{dt} \mathcal{B}(x) \leq \alpha (l_f + \sqrt{n}\mu - \sigma \alpha) < 0.$$

Denote $k = \alpha(\sigma \alpha - l_f - \sqrt{n}\mu)$. Then $k > 0$ and

$$\frac{d}{dt} \mathcal{B}(x) \leq -k. \quad (11)$$

Integrating both sides of (11) from $t_0 = 0$ to t , we have

$$\mathcal{B}(x(t)) \leq \mathcal{B}(x(t_0)) - kt.$$

Thus, when $t = \mathcal{B}(x(t_0)) / k$, $\mathcal{B}(x(t)) = 0$. Therefore, the state of neural network (4) reaches the equality feasible region \mathcal{C} in finite time by an upper bound of the hit time $t = \mathcal{B}(x(t_0)) / k$.

Next we prove that when $t \geq \mathcal{B}(x(t_0)) / k$, the state vector of neural network (4) remains inside \mathcal{C} thereafter. If not so, we suppose that the state trajectory leaves \mathcal{C} at time t_1 and stays outside of \mathcal{C} for almost all $t \in (t_1, t_2)$, where $t_1 < t_2$. Then,

$\mathcal{B}(x(t_1)) = 0$, and from the above analysis, $\mathcal{B}(x(t)) < 0$ for almost all $t \in (t_1, t_2)$. By the definition of $\mathcal{B}(x)$, we have $\mathcal{B}(x(t)) \geq 0$ for any $t \in [t_0, \infty)$, which contradicts the result above. That is the state vector of neural network (4) reaches the equality feasible region \mathcal{C} by $t = \mathcal{B}(x(t_0))/k$ and stays there thereafter. \square

4.3. Finite-time convergence to \mathcal{S}

In this subsection, neural network (4) is further proved to be convergent to the feasible region \mathcal{S} in finite time.

Theorem 3. Suppose that Assumptions 1 and 2 hold. For any $x_0 \in B(\hat{x}, r)$, the state vector of neural network (4) reaches the feasible region \mathcal{S} in finite time and stays there thereafter, if $\sigma > (l_f + \sqrt{n}\mu)/\sqrt{\lambda_{\min}(AA^T)}$ and $\mu > rl_f/\omega$.

Proof. According to Theorem 2, the trajectory of $x(t)$ reaches the equality feasible region \mathcal{C} in finite time and stays there thereafter. It remains to show that once in the set \mathcal{C} , the trajectory reaches the set $[u, v]$ in finite time and stays there thereafter.

According to its definition, $\mathcal{D}(x)$ in Section 3 is convex in \mathbb{R}^n . By the chain rule, we have

$$\frac{d}{dt}\mathcal{D}(x) = \zeta^T \frac{dx(t)}{dt}, \quad \forall \zeta \in \partial\mathcal{D}(x(t)).$$

From Theorem 1, for any $x_0 \in B(\hat{x}, r)$, we have $x(t) \in B(\hat{x}, r)$. Since $x \in \mathcal{C}$, from Lemma 4, we have $Px = q$. Thus $P\dot{x} = 0$. Since x can be written as $x = Px + (I - P)x$, then $\dot{x} = (I - P)\dot{x}$. From (4), combining that $(I - P)A^T = 0$, we have

$$\epsilon \frac{dx}{dt} \in -(I - P)(\partial f(x) + \mu K[g_{[u,v]}(x)]). \quad (12)$$

Then, for any $x \in (B(\hat{x}, r) \cap \mathcal{C}) \setminus [u, v]$, there exist $\eta \in \partial f(x)$ and $\xi \in K[g_{[u,v]}(x)]$ such that

$$\begin{aligned} \epsilon \frac{d}{dt}\mathcal{D}(x) &= \epsilon \xi^T \dot{x}(t) \\ &= -\xi^T (I - P)(\eta + \mu \xi) \\ &= -\xi^T (I - P)\eta - \mu \xi^T (I - P)\xi \\ &\leq \|(I - P)\xi\|_2 (\|\eta\|_2 - \mu \|(I - P)\xi\|_2), \end{aligned}$$

in which the last inequality holds since $(I - P)^2 = I - P$.

For any $\eta \in \partial f(x)$, we have $\|\eta\|_2 \leq l_f$. By Lemma 6, for any $\xi \in K[g_{[u,v]}(x)]$, $\|(I - P)\xi\|_2 > \omega/r$. Then, if $\mu > rl_f/\omega$, we have

$$\epsilon \frac{d}{dt}\mathcal{D}(x) < \frac{\omega}{r} \left(l_f - \frac{\mu\omega}{r} \right) < 0.$$

Denote $s = \omega(\mu\omega/r - l_f)/(\epsilon r)$. Then $s > 0$ and

$$\frac{d}{dt}\mathcal{D}(x) < -s. \quad (13)$$

Integrating (13) from t_c to t , we have

$$\mathcal{D}(x(t)) \leq \mathcal{D}(x(t_c)) - s(t - t_c),$$

where t_c is the time that $x(t)$ reaches to \mathcal{C} . Thus, when $t \geq \mathcal{D}(x(t_c))/s + t_c$, $\mathcal{D}(x(t)) \leq 0$. Therefore, the state of neural network (4) reaches \mathcal{S} in finite time.

Similar to the proof of Theorem 2, we can prove that the state vector of neural network (4) stays in \mathcal{S} thereafter. \square

4.4. Optimality analysis

Theorem 4. Suppose that Assumptions 1 and 2 hold. Any equilibrium point of neural network (4) is an optimal solution to problem (2) and vice versa, if $\sigma > (l_f + \sqrt{n}\mu)/\sqrt{\lambda_{\min}(AA^T)}$ and $\mu > rl_f/\omega$.

Proof. Let x^* be an optimal solution of problem (2), then $x^* \in \mathcal{S}$. Since x^* is a minimum point of $f(x)$ over the feasible region \mathcal{S} , according to Lemma 3, we get that

$$0 \in \partial f(x^*) + N_{\mathcal{S}}(x^*),$$

where $N_{\mathcal{S}}(x^*)$ is the normal cone to the set \mathcal{S} at x^* . Since $\text{int}([u, v]) \cap \mathcal{C} \neq \emptyset$, we get that $0 \in \text{int}([u, v] - \mathcal{C})$. From Lemma 2, it follows that $N_{\mathcal{S}} = N_{[u,v]} + N_{\mathcal{C}}$. Then

$$0 \in \partial f(x^*) + N_{\mathcal{C}}(x^*) + N_{[u,v]}(x^*).$$

Suppose $x^* \in \text{bd}([u, v]) \cap \mathcal{C}$, where $\text{bd}(\cdot)$ denotes the boundary of the set. It follows that there exists $\eta \in N_{\mathcal{S}}(x^*)$ such that $\eta \in -\partial f(x^*)$ and $\|\eta\|_2 \leq l_f$. By simple derivation, we get that $N_{\mathcal{C}}(x^*) = \{vA^T\gamma : v \geq 0, -1 \leq \gamma_i \leq 1 \text{ and at least one } \gamma_i = 1 \text{ or } -1\}$ and $N_{[u,v]}(x^*) = \{v\xi : v \geq 0, \xi \in K[g_{[u,v]}(x^*)]\}$ and at least one $\xi_i = 1$ or -1 . Then, there exist $\alpha \geq 0, \beta \geq 0, \gamma \in \mathbb{R}^m$ and $\xi \in \mathbb{R}^n$ such that $\alpha A^T\gamma \in N_{\mathcal{C}}(x^*), \beta \xi \in N_{[u,v]}(x^*)$ and

$$\eta = \alpha A^T\gamma + \beta \xi.$$

Then we prove that $\eta \in \sigma A^T K[g_0(Ax^* - b)] + \mu K[g_{[u,v]}(x^*)]$. We say that $\beta \leq \mu$. If not, we have $\beta > \mu$. Similar to the proof of Lemma 5, we have $(x^* - \hat{x})^T \xi = \sum_{i=1}^n (x_i^* - \hat{x}_i) \xi_i \geq \omega$, where ω is defined in Lemma 5. Since $x^*, \hat{x} \in \mathcal{C}, Ax^* = A\hat{x} = b$. Then we have $(x^* - \hat{x})^T \eta = \alpha (x^* - \hat{x})^T A^T \gamma + \beta (x^* - \hat{x})^T \xi \geq \beta \omega > \mu \omega$. Thus $\|\eta\|_2 > \mu \omega / \|x^* - \hat{x}\|_2 \geq \mu \omega / r$. If $\mu > rl_f/\omega$, we have $\|\eta\|_2 > l_f$, which contradicts $\|\eta\|_2 \leq l_f$. Consequently $\beta \leq \mu$ and it follows that $\beta \xi \in \mu K[g_{[u,v]}(x^*)]$.

We say that $\alpha \leq \sigma$. If not, then $\alpha > \sigma$. We have $\|\eta\|_2 = \|\alpha A^T\gamma + \beta \xi\|_2 \geq \alpha \|A^T\gamma\|_2 - \beta \|\xi\|_2$. According to the proof of Theorem 2, $\|A^T\gamma\|_2 \geq \sqrt{\lambda_{\min}(AA^T)}$. Combining that $\|\xi\|_2 \leq \sqrt{n}$ and $\beta \leq \mu$, we have $\|\eta\|_2 \geq \sigma \sqrt{\lambda_{\min}(AA^T)} - \sqrt{n}\mu$. If $\sigma > (l_f + \sqrt{n}\mu)/\sqrt{\lambda_{\min}(AA^T)}$, we have $\|\eta\|_2 > l_f$, which contradicts $\|\eta\|_2 \leq l_f$. Hence, $\alpha \leq \sigma$ and it follows that $\alpha A^T\gamma \in \sigma A^T K[g_0(Ax^* - b)]$. Then we have $\eta \in \sigma A^T K[g_0(Ax^* - b)] + \mu K[g_{[u,v]}(x^*)]$.

Consequently, $0 \in \partial f(x^*) + \sigma A^T K[g_0(Ax^* - b)] + \mu K[g_{[u,v]}(x^*)]$, which means that x^* is an equilibrium point of neural network (4).

Similarly, when $x^* \in \text{int}([u, v]) \cap \mathcal{C}$, the result also holds.

Next, let us prove the reverse side. Let \bar{x} be an equilibrium point of neural network (4), then there exist $\bar{\eta} \in \partial f(\bar{x}), \bar{\gamma} \in K[g_0(A\bar{x} - b)]$ and $\bar{\xi} \in K[g_{[u,v]}(\bar{x})]$ such that

$$\bar{\eta} + \sigma A^T \bar{\gamma} + \mu \bar{\xi} = 0. \quad (14)$$

According to Theorem 3, if the conditions of the theorem hold, $\bar{x} \in [u, v]$. From (14), we get the following projection formulation

$$\bar{x} = \phi_{[u,v]}(\bar{x} - \bar{\eta} - \sigma A^T \bar{\gamma}),$$

where $\phi_{[u,v]}(y) = (\phi_{[u_1, v_1]}(y_1), \phi_{[u_2, v_2]}(y_2), \dots, \phi_{[u_n, v_n]}(y_n))^T$ with $\phi_{[u_i, v_i]}(y_i)$ ($i = 1, 2, \dots, n$) defined as

$$\phi_{[u_i, v_i]}(y_i) = \begin{cases} v_i, & y_i \geq v_i, \\ y_i, & u_i < y_i < v_i, \\ u_i, & y_i \leq u_i. \end{cases}$$

Furthermore, using the well-known projection theorem (Kinderlehrer & Stampacchia, 1982), it is equivalent to the following variational inequality

$$(x - \bar{x})^T (\bar{\eta} + \sigma A^T \bar{\gamma}) \geq 0, \quad \forall x \in [u, v].$$

From Theorem 3, if the conditions of the theorem hold, $\bar{x} \in \mathcal{C}$ and it follows that $A\bar{x} = b$. For any $x \in \mathcal{S}$, we have $Ax = b$. Then, for any $x \in \mathcal{S}$, $(x - \bar{x})^T \bar{\eta} \geq 0$. Since $f(x)$ is pseudoconvex on \mathcal{S} , it follows that $f(x) \geq f(\bar{x})$ for any $x \in \mathcal{S}$. Therefore, \bar{x} is a minimum point of $f(x)$ over \mathcal{S} . \square

4.5. Convergence analysis

In this subsection, the convergence property of neural network (4) is investigated by using the Lyapunov method and differential inclusion theory (e.g., see Aubin & Cellina, 1984; Clarke, 1983; Forti et al., 2004; Lu & Chen, 2006, and references therein).

Theorem 5. Suppose that Assumptions 1 and 2 hold. For any $x_0 \in B(\hat{x}, r)$, the state vector of neural network (4) is convergent to an optimal solution of problem (2) if $\sigma > (l_f + \sqrt{n}\mu)/\sqrt{\lambda_{\min}(AA^T)}$ and $\mu > rl_f/\omega$.

Proof. Let \bar{x} be an equilibrium point of neural network (4). According to Theorem 4, \bar{x} is an optimal solution of problem (2). Thus $A\bar{x} = b$ and $\bar{x} \in [u, v]$. From Theorem 3, we can suppose that $x_0 \in \mathcal{S}$. Let $\psi(x) = f(x) + \sigma \|Ax - b\|_1 + \mu \mathcal{D}(x)$. There exist $\bar{\eta} \in \partial f(\bar{x})$, $\bar{\gamma} \in K[g_0(A\bar{x} - b)]$ and $\bar{\xi} \in K[g_{[u,v]}(\bar{x})]$ such that

$$\bar{\eta} + \sigma A^T \bar{\gamma} + \mu \bar{\xi} = 0. \quad (15)$$

Consider the following Lyapunov function

$$V(x) = \epsilon \left[\psi(x) - \psi(\bar{x}) + \frac{1}{2} \|x - \bar{x}\|_2^2 \right]. \quad (16)$$

Since \bar{x} is a minimum point of $f(x)$ on \mathcal{S} , then for any $x \in \mathcal{S}$, $\psi(x) \geq \psi(\bar{x})$. Thus $V(x) \geq \epsilon \|x - \bar{x}\|_2^2/2$.

We have

$$\partial V(x) = \epsilon [\partial \psi(x) + x - \bar{x}].$$

By using the chain rule, it follows that $V(x(t))$ is differentiable for almost all (a.a.) $t \geq t_0$ and it results in

$$\frac{d}{dt} V(x(t)) = \zeta(t)^T \dot{x}(t), \quad \forall \zeta(t) \in \partial V(x(t)).$$

From (4), $\epsilon \dot{x}(t) \in -\partial \psi(x(t))$, hence by choosing $\zeta(t) = \epsilon [-\epsilon \dot{x}(t) + x - \bar{x}] \in \partial V(x(t))$, we have

$$\begin{aligned} \frac{d}{dt} V(x(t)) &= \epsilon [-\epsilon \dot{x}(t) + x - \bar{x}]^T \dot{x}(t) \\ &= -\|\epsilon \dot{x}(t)\|_2^2 + \epsilon (x - \bar{x})^T \dot{x}(t) \\ &\leq \sup_{\theta \in \partial \psi(x)} (-\|\theta\|_2^2) + \sup_{\theta \in \partial \psi(x)} [-(x - \bar{x})^T \theta] \\ &= -\inf_{\theta \in \partial \psi(x)} \|\theta\|_2^2 - \inf_{\theta \in \partial \psi(x)} (x - \bar{x})^T \theta. \end{aligned} \quad (17)$$

For any $\theta \in \partial \psi(x)$, there exist $\eta \in \partial f(x)$, $\gamma \in K[g_0(Ax - b)]$ and $\xi \in K[g_{[u,v]}(x)]$ such that $\theta = \eta + \sigma A^T \gamma + \mu \xi$. Since $f(x)$ is pseudoconvex on \mathcal{S} , $\partial f(x)$ is pseudomonotone on \mathcal{S} . From the proof of Theorem 4, for any $x \in \mathcal{S}$, $(x - \bar{x})^T \eta \geq 0$. Then it implies that $(x - \bar{x})^T \eta \geq 0$ for any $\eta \in \partial f(x)$. For $x, \bar{x} \in \mathcal{S}$, we have $(x - \bar{x})^T A^T \gamma = 0$ since $Ax = A\bar{x} = b$. From the definition of $g_{[u,v]}(x)$, we have $(x_i - \bar{x}_i) \xi_i \geq 0$ ($i = 1, 2, \dots, n$). Thus $(x - \bar{x})^T \xi \geq 0$. Consequently, for any $\theta \in \partial \psi(x)$, we have $(x - \bar{x})^T \theta \geq 0$. Then,

$$\frac{d}{dt} V(x(t)) \leq -\inf_{\theta \in \partial \psi(x)} \|\theta\|_2^2. \quad (18)$$

Define $\mathcal{H}(x) = \inf_{\theta \in \partial \psi(x)} \|\theta\|_2^2$. It is easy to get that $\mathcal{H}(x) = 0$ if and only if x is an equilibrium point of neural network (4).

Since $x(t) \in \mathcal{S}$ is bounded, from (4), $\|\dot{x}(t)\|_2$ is also bounded, denoted by M . Then, there exists an increasing sequence $\{t_k\}$ with $\lim_{k \rightarrow \infty} t_k = \infty$ and a limit point \bar{x} such that $\lim_{k \rightarrow \infty} x(t_k) = \bar{x}$.

Next, we prove that $\mathcal{H}(\bar{x}) = 0$. If it does not hold, that is $\mathcal{H}(\bar{x}) > 0$. From the definition of $\mathcal{H}(x)$, there exist $\varepsilon > 0$ and $\delta > 0$, such that $\mathcal{H}(x) > \varepsilon$ for all $x \in B(\bar{x}, \delta)$, where $B(\bar{x}, \delta) = \{x \in \mathbb{R}^n : \|x - \bar{x}\|_2 \leq \delta\}$ is the δ neighborhood of \bar{x} . Since $\lim_{k \rightarrow \infty} x(t_k) = \bar{x}$, there exists a positive integer N , such that for all

$k \geq N$, $\|x(t_k) - \bar{x}\|_2 \leq \delta/2$. When $t \in [t_k - \delta/(4M), t_k + \delta/(4M)]$ and $k \geq N$, we have

$$\begin{aligned} \|x(t) - \bar{x}\|_2 &\leq \|x(t) - x(t_k)\|_2 + \|x(t_k) - \bar{x}\|_2 \\ &\leq M|t - t_k| + \frac{\delta}{2} \leq \delta. \end{aligned}$$

It follows that $\mathcal{H}(x(t)) > \varepsilon$ for all $t \in [t_k - \delta/(4M), t_k + \delta/(4M)]$. On one hand, since the Lebesgue measure of the set $t \in \bigcup_{k \geq N} [t_k - \delta/(4M), t_k + \delta/(4M)]$ is infinite, then we have

$$\int_{t_0}^{\infty} \mathcal{H}(x(t)) dt = \infty. \quad (19)$$

On the other hand, by (18), $V(x(t))$ is monotonically nonincreasing and bounded on \mathcal{S} , then, there exists a constant V_0 such that $\lim_{t \rightarrow \infty} V(x(t)) = V_0$. We have

$$\begin{aligned} \int_{t_0}^{\infty} \mathcal{H}(x(t)) dt &= \lim_{s \rightarrow \infty} \int_{t_0}^s \mathcal{H}(x(t)) dt \\ &\leq -\lim_{s \rightarrow \infty} \int_{t_0}^s \dot{V}(x(t)) dt \\ &= -\left[\lim_{s \rightarrow \infty} V(x(s)) - V(x(t_0)) \right] \\ &= -V_0 + V(x(t_0)), \end{aligned}$$

which contradicts (19). Therefore, we have $\mathcal{H}(\bar{x}) = 0$. That is, the limit point \bar{x} is an equilibrium point of neural network (4).

Finally, let us define another Lyapunov function

$$\tilde{V}(x) = \epsilon \left[\psi(x) - \psi(\bar{x}) + \frac{1}{2} \|x - \bar{x}\|_2^2 \right].$$

Similar to the above proof, we have $\tilde{V}(x(t)) \geq \epsilon \|x - \bar{x}\|_2^2/2$ and $\dot{\tilde{V}}(x(t)) \leq 0$. From the continuity of function $\tilde{V}(x(t))$, for any $\tilde{\varepsilon} > 0$, there exists $\tilde{\delta} > 0$ such that $\tilde{V}(x(t)) < \tilde{\varepsilon}^2$ when $\|x - \bar{x}\|_2 \leq \tilde{\delta}$. Since $\tilde{V}(x(t))$ is monotonically nonincreasing on interval $[t_0, \infty)$, there exists a positive integer L , such that when $t \geq t_L$,

$$\epsilon \|x(t) - \bar{x}\|_2^2 \leq 2\tilde{V}(x(t)) \leq 2\tilde{V}(x(t_L)) < 2\tilde{\varepsilon}^2.$$

That is $\lim_{t \rightarrow +\infty} x(t) = \bar{x}$. Then the state vector of neural network (4) is convergent to an equilibrium point. Combining with Theorem 4, this completes the proof. \square

4.6. Further results on finite-time convergence

Recently, the finite-time convergence of recurrent neural networks were investigated for solving linear programming problems (e.g., see Chong et al., 1999; Forti et al., 2004; Liu, Cao, & Chen, 2010). In Marco, Forti, and Grazzini (2006), the robustness of convergence in finite time was investigated for the linear programming neural networks. Here, the convergence to an optimal solution of problem (2) can be achieved in finite time if a mild condition is satisfied. Similar to that in Forti et al. (2004), to achieve the finite-time convergence, another assumption is stated as follows.

Assumption 3. There exists $\alpha > 0$ such that

$$\inf_{x \in \mathcal{S}, \mathcal{M}} \left\{ \inf_{\theta \in \partial \psi(x)} \|\theta\|_2 \right\} > \alpha,$$

where $\psi(x) = f(x) + \sigma \|Ax - b\|_1 + \mu \mathcal{D}(x)$ and \mathcal{M} is the optimal solution set of problem (2).

Theorem 6. Suppose that Assumptions 1–3 hold. For any $x_0 \in B(\hat{x}, r)$, the state of neural network (4) is convergent to an optimal solution of problem (2) in finite time if $\sigma > (l_f + \sqrt{n}\mu)/\sqrt{\lambda_{\min}(AA^T)}$ and $\mu > rl_f/\omega$.

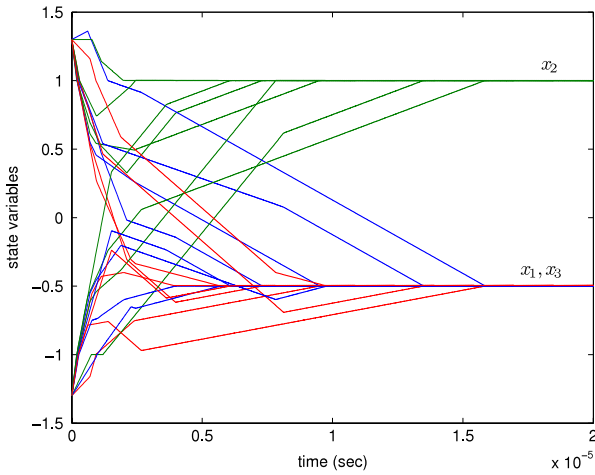


Fig. 1. Transient behaviors of the state variables of neural network (4) with $\sigma = 16$ and $\mu = 12$ in Example 1.

Proof. From (18) and Assumption 3, for $x \in \mathcal{S} \setminus \mathcal{M}$, we have

$$\dot{V}(x(t)) \leq -\alpha^2. \tag{20}$$

Integrating both sides of (20) from t_δ to t , one gets that

$$V(x(t)) \leq V(x(t_\delta)) - \alpha^2(t - t_\delta),$$

where t_δ is the minimum time that $x(t)$ reaches \mathcal{S} . Then $V(x(t)) \leq 0$ when $t \geq V(x(t_\delta))/\alpha^2 + t_\delta$. From (16), we have $V(x(t)) \geq \epsilon \|x - \bar{x}\|_2^2/2$, where \bar{x} is the optimal solution which $x(t)$ converges to. Consequently, $x(t) = \bar{x}$ when $t \geq V(x(t_\delta))/\alpha^2 + t_\delta$. That is, $x(t)$ is convergent to an optimal solution of problem (2) in finite time. \square

5. Illustrative examples

In this section, three examples of optimization problems are solved using the proposed neural network.

Example 1. Consider a nonsmooth optimization problem as follows:

$$\begin{aligned} \text{minimize} \quad & f(x) = |x_1 - x_2 - 2x_3 + 1| + |x_1 + 2x_2 - x_3 \\ & \quad - 2| + |x_1 + x_2 - 1| - |x_3 - 1|, \\ \text{subject to} \quad & x_1 + x_2 + x_3 = 0, \\ & -1 \leq x_1, x_2, x_3 \leq 1. \end{aligned} \tag{21}$$

In this problem, the objective function $f(x)$ is nonsmooth and nonconvex. If we substitute $x_1 + x_2 = -x_3$ into $f(x)$, we get that $f(x)$ is convex on the feasible region. Consequently, the proposed neural network in (4) is capable of solving this problem. Let $\hat{x} = (0, 0, 0)^T \in \text{int}([u, v]) \cap \mathcal{C}$, then we have $\omega = 1$. Moreover, the restricted region $[u, v] \subset B(\hat{x}, r)$ with $r = \sqrt{3}$. An upper bound of the Lipschitz constant of $f(x)$ on $B(\hat{x}, r)$ is estimated as $l_f = 6.5$. Then the design parameters are estimated as $\sigma > 15.02$ and $\mu > 11.26$. Let $\epsilon = 10^{-5}$, $\sigma = 16$ and $\mu = 12$ in the simulations. Fig. 1 shows the transient behaviors of the state variables of the neural network with 8 random initial points. Fig. 2 depicts the phase plot of $(x_1, x_2, x_3)^T$ from 8 random initial points, which shows that the state variables converge to the unique optimal solution $x^* = (-0.5, 1, -0.5)^T$.

Example 2. We now use the proposed neural network to solve a pseudoconvex optimization problem (Hu & Wang, 2006). Consider

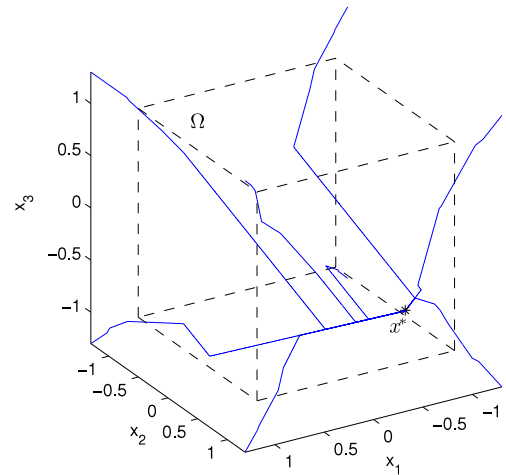


Fig. 2. Three-dimensional phase plot of the trajectories of (x_1, x_2, x_3) of neural network (4) with $\sigma = 16$ and $\mu = 12$ in Example 1.

the quadratic fractional programming problem in (7) with

$$Q = \begin{pmatrix} 5 & -1 & 2 & 0 \\ -1 & 5 & -1 & 3 \\ 2 & -1 & 3 & 0 \\ 0 & 3 & 0 & 5 \end{pmatrix}, \quad a = \begin{pmatrix} 1 \\ -2 \\ -2 \\ 1 \end{pmatrix}, \quad a_0 = -2,$$

and

$$c = (2, 1, 1, 0)^T, \quad c_0 = 4.$$

It is easily verified that Q is symmetric and positive definite, and consequently is pseudoconvex on $X = \{x \in \mathbb{R}^4 : c^T x + c_0 > 0\}$. This problem was solved by Hu and Wang (2006) with the constraints $x \in [u, v] = \{x \in \mathbb{R}^4 : 1 \leq x_i \leq 10, i = 1, 2, 3, 4\}$, the objective function $f(x)$ is obviously pseudoconvex on $[u, v]$. Thus the projection neural network is suitable for solving the problem in this case (Hu & Wang, 2006). The proposed neural network herein is capable of solving this problem too. Let $\hat{x} = (5.5, 5.5, 5.5, 5.5)^T \in \text{int}([u, v])$, then we have $\omega = 4.5$. Moreover, the restricted region $[u, v] \subset B(\hat{x}, r)$ with $r = 9$. An upper bound of the Lipschitz constant of $f(x)$ on $B(\hat{x}, r)$ is estimated as $l_f = 9$. Then the design parameter is estimated as $\mu > 18$. Let $\epsilon = 10^{-5}$ and $\mu = 19$ in the simulations. Fig. 3 depicts that the state variables of the neural network are convergent to the unique optimal solution $x^* = (1, 1, 1, 1)^T$ with 10 random initial values.

Next, we consider this problem with both equality and bound constraints. We assume that the feasible region $\mathcal{S} = \{x \in \mathbb{R}^4 : Ax = b, 5 \leq x_i \leq 10, i = 1, 2, 3, 4\}$, where

$$A = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

The neural network (4) is still capable of solving this problem. Let $\hat{x} = (8, 7, 8.5, 6.5)^T \in \text{int}([u, v]) \cap \mathcal{C}$, then we have $\omega = 1.5$. Moreover, the restricted region $[u, v] \subset B(\hat{x}, r)$ with $r = 6.52$. An upper bound of the Lipschitz constant of $f(x)$ on $B(\hat{x}, r)$ is estimated as $l_f = 7.34$. Then the designed parameters are estimated as $\sigma > 50.32$ and $\mu > 31.91$ respectively. Let $\epsilon = 10^{-5}$, $\sigma = 51$ and $\mu = 32$ in the simulations. Fig. 4 shows that the state vector of the neural network is convergent to the unique optimal solution $x^* = (6, 5, 7, 5)^T$ with 10 random initial values.

Example 3. Consider the quadratic fractional programming problem in (7) with

$$Q = \begin{pmatrix} -1 & 0.5 & 1 & 0 \\ 0.5 & 5.5 & -1 & -0.5 \\ 1 & -1 & 1 & 0 \\ 0 & -0.5 & 0 & 0 \end{pmatrix}, \quad a = \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix},$$

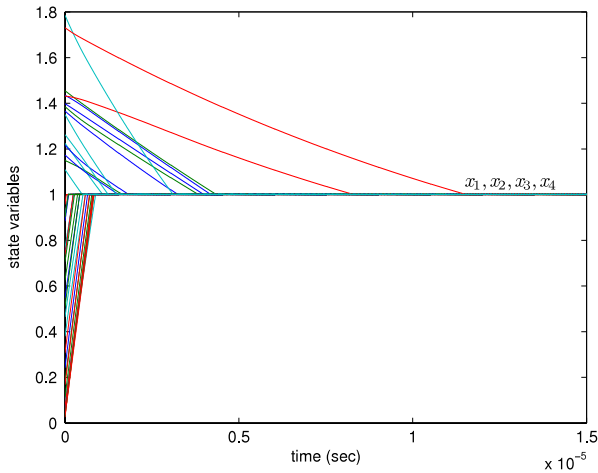


Fig. 3. Transient behaviors of the state variables of neural network (4) with $\mu = 19$ in Example 2.

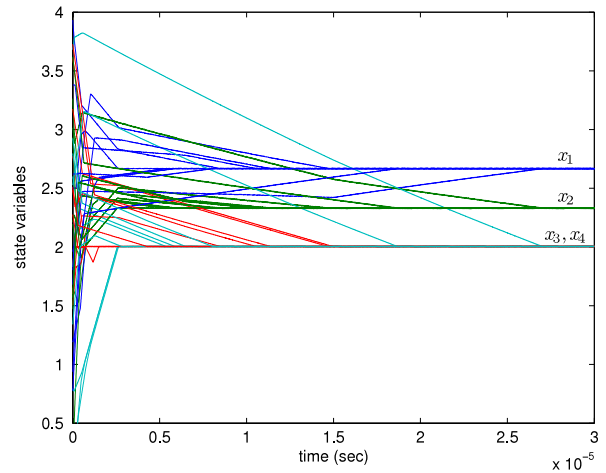


Fig. 5. Transient behaviors of the state variables of neural network (4) with $\sigma = 10$ and $\mu = 7$ in Example 3.

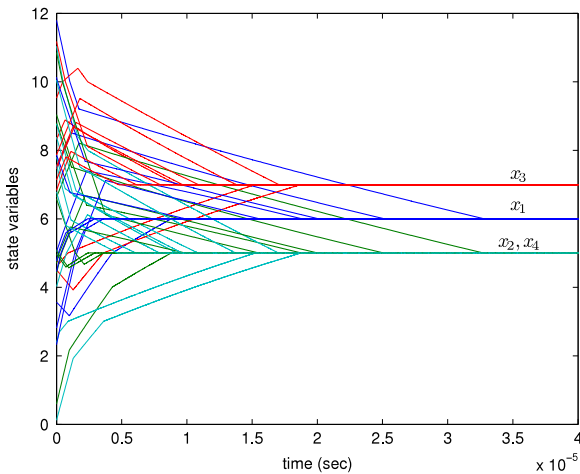


Fig. 4. Transient behaviors of the state variables of neural network (4) with $\sigma = 51$ and $\mu = 32$ in Example 2.

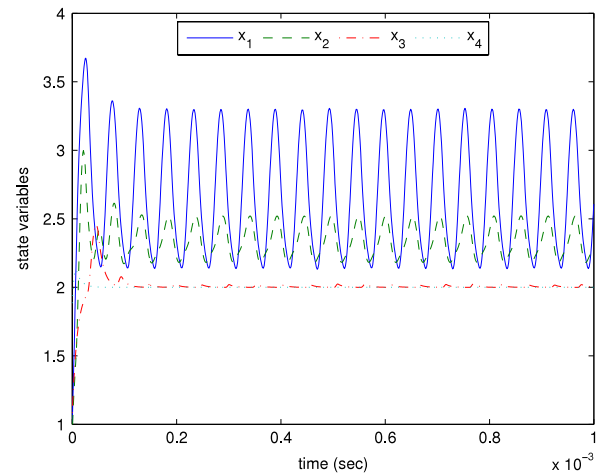


Fig. 6. Transient behaviors of state variables of the projection neural network from a random initial point in Example 3.

$$a_0 = -2,$$

$$c = (1, 1, 1, -1)^T, \quad c_0 = 6,$$

$$A = \begin{pmatrix} 1 & 1 & -1 & 0 \\ 1 & -2 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 3 \\ 0 \end{pmatrix},$$

and

$$x \in \{\mathbb{R}^4 : 2 \leq x_1, x_2, x_3, x_4 \leq 4\}.$$

As Q is not positive definite, the objective function $f(x)$ is not pseudoconvex on $X = \{x \in \mathbb{R}^4 : c^T x + c_0 > 0\}$. However, if we substitute $x_3 = x_1 + x_2 - 3$ and $x_4 = -x_1 + 2x_2$ into the objective function, the objective function can be written as $\tilde{f}(x_1, x_2) = (2x_1^2 + 2.5x_2^2 + 4x_1x_2 - 13x_1 + 10)/(c^T x + c_0)$, which is pseudoconvex on X . Furthermore, $f(x)$ is pseudoconvex on the feasible region \mathcal{S} . Then the neural network (4) is capable of solving this problem. Let $\hat{x} = (3, 3, 3, 3)^T \in \text{int}([u, v]) \cap \mathcal{C}$, then we have $\omega = 1$. Moreover, the restricted region $[u, v] \subset B(\hat{x}, r)$ with $r = 2$. An upper bound of the Lipschitz constant of $f(x)$ on $B(\hat{x}, r)$ is estimated as $l_f = 3.28$. Then the design parameters are estimated as $\sigma > 9.99$ and $\mu > 6.56$ respectively. Let $\epsilon = 10^{-5}$, $\sigma = 10$ and $\mu = 7$ in the simulations. Fig. 5 shows that the state vector of the neural network is convergent to the unique optimal solution $x^* = (8/3, 7/3, 2, 2)^T$ with 10 random initial values. Fig. 6 depicts the simulation results of the projection neural network (Xia et al.,

2002; Xia & Wang, 2005) for solving this problem, which show that the state vector is not convergent from a random initial point.

6. Dynamic portfolio optimization

In this section, the proposed neurodynamic optimization approach is applied for dynamic portfolio optimization. First, the objective function of portfolio optimization in a given portfolio selection model is proven to be pseudoconvex in the feasible region. Then the setting of initial points is discussed, which guarantees that the recurrent neural network will yield an optimal portfolio. Finally, simulation results are discussed and compared based on the given portfolio selection model.

Portfolio optimization (Markowitz, 1991) is a means to optimize a set of financial instruments held to achieve high expected returns by spreading the risk of possible loss due to low expected performance. A good portfolio is not just a long list of good stocks and bonds, but also a balanced whole that provides protections and opportunities with respect to a wide range of contingencies.

Since Markowitz's pioneering work of Mean-Variance (MV) model in portfolio investment (Markovitz, 1952), many studies have been done to enhance the model. In particular, a portfolio model to maximize the probability that the rate of return is no less than an expected one is proposed (e.g., see Liu, Ku, & C, 1993; Liang & Tang, 2001; Tang, Wang, & Liang, 2002, and references therein).

As the market conditions vary, dynamic portfolio optimization is both necessary and rewarding. The proposed recurrent neural network can serve as a parallel computing mechanism for real-time portfolio optimization.

6.1. Portfolio optimization

For n securities, suppose that the rate of return is a random vector $\xi = (\xi_1, \xi_2, \dots, \xi_n)^T$ with normal distribution; i.e., $\xi \sim N(a, Q)$. Here $a = (a_1, a_2, \dots, a_n)^T > 0$ is the mean vector of ξ , and $Q \in \mathbb{R}^{n \times n}$ is the positive definite covariance matrix of ξ , which is usually considered as a measurement of risk. Let $x = (x_1, x_2, \dots, x_n)^T$ be the investment ratio vector, such that $\sum_i x_i = 1$. Thus, the total rate of return is $\eta = x^T \xi$, $\eta \sim N(a^T x, x^T Q x)$, and $(\eta - a^T x) / \sqrt{x^T Q x} \sim N(0, 1)$. Thus the optimization model of portfolio investment with probability criterion is

$$\begin{aligned} & \text{maximize} \quad P(\eta \geq r) = \Phi\left(\frac{a^T x - r}{\sqrt{x^T Q x}}\right), \\ & \text{subject to} \quad \sum_{i=1}^n x_i = 1, \\ & \quad \quad \quad 0 \leq x \leq 1, \end{aligned} \tag{22}$$

where $r \geq 0$ is the expected rate of return, $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x \exp(-\frac{t^2}{2}) dt$ is the standard normal distribution function. Since $\Phi(\cdot)$ is monotone increasing on \mathbb{R} , the following equivalent optimization problem can be formulated

$$\begin{aligned} & \text{minimize} \quad f(x) = \frac{r - a^T x}{\sqrt{x^T Q x}}, \\ & \text{subject to} \quad \sum_{i=1}^n x_i = 1, \\ & \quad \quad \quad 0 \leq x \leq 1. \end{aligned} \tag{23}$$

For an expected rate of return r , $\Phi(-f(x^*))$ gives the investors the maximum probability of the portfolio investment on current n securities with respect to the expected rate of return r , where x^* is an optimal solution of problem (23).

6.2. Theoretical results

Theorem 7. If $w : \mathcal{X} \rightarrow \mathbb{R}$ and $v : \mathcal{X} \rightarrow \mathbb{R}$ are convex, then $f(x) = w(x)/v(x)$ is pseudoconvex on $\mathcal{X} = \{x : w(x) < 0, v(x) > 0\}$.

Proof. By setting $(x_1 - x_2)^T \nabla f(x_2) \geq 0$, we have

$$\frac{1}{v(x_2)} \left(\nabla w(x_2) - \frac{w(x_2)}{v(x_2)} \nabla v(x_2) \right)^T (x_1 - x_2) \geq 0,$$

where $\nabla(\cdot)$ is the gradient of a function.

Since w and v are both convex, $\forall x_1, x_2 \in \mathcal{X}$, we have $w(x_1) - w(x_2) \geq (x_1 - x_2)^T \nabla w(x_2)$ and $v(x_1) - v(x_2) \geq (x_1 - x_2)^T \nabla v(x_2)$, and $w(x_2)/v(x_2) < 0$. Then

$$\begin{aligned} 0 & \leq (x_1 - x_2)^T \nabla w(x_2) - \frac{w(x_2)}{v(x_2)} (x_1 - x_2)^T \nabla v(x_2) \\ & \leq w(x_1) - w(x_2) - \frac{w(x_2)}{v(x_2)} (v(x_1) - v(x_2)) \\ & = w(x_1) - \frac{w(x_2)}{v(x_2)} v(x_1). \end{aligned}$$

Since $v(x_1) > 0$, $w(x_1)/v(x_1) \geq w(x_2)/v(x_2)$ follows directly which indicates that $f(x) = w(x)/v(x)$ is pseudoconvex on \mathcal{X} . \square

According to Theorem 7, the objective function $f(x)$ in (23) is pseudoconvex on $\mathcal{X}_0 = \{x : r - a^T x < 0\}$.

Theorem 8. For all $r < \max_i a_i$, the state vector of neural network (4) will converge to the global optimal solution of (23) for any $x_0 \in \mathcal{X} = \{x : r - a^T x < 0, \sum_i x_i = 1, 0 \leq x \leq 1\}$ and sufficiently large parameters σ and μ in the neural network (4).

Proof. First, the set \mathcal{X} is not empty for any $r < \max_i a_i$ since the vector with the j th element of 1 and the rest 0; i.e., $e_j = (0^1, \dots, 0^{j-1}, 1^j, 0^{j+1}, \dots, 0^n)^T$ is a feasible state, where j is the index of $\max_i a_i$. Thus we can always find a feasible initial state x_0 .

Second, since the denominator of the objective function is positive for all $x \in \mathbb{R}^n \setminus \{0\}$, $f(x_1) < f(x_2)$ always holds if $r - a^T x_1 < 0$ and $r - a^T x_2 \geq 0$. As \mathcal{X} is not empty, it is clear that the optimal solution of problem (23) under condition $r - a^T x < 0$ is the optimal solution of the original problem (23).

Finally, from the theorems in Section 4, we know that for any $x_0 \in \mathcal{X}$, the state vector of neural network (4) will remain in the feasible region forever and converge to the optimal solution of problem (23) under condition $r - a^T x < 0$, which is also the optimal solution to problem (23) as just stated. \square

6.3. Simulation results

Example 4. Here, we use the proposed neural network to solve a portfolio optimization problem. The numerical example is generated randomly in the following steps. First, the expected rate of return to a particular security varies over time. Thus the function $a_i(t) = r_i + k_i t + R_i \sin(t/T_i + \omega_i)$ ($i = 1, 2, \dots, 5$) is used to describe the mean of expected rate of return at time t , where $k_i \sim U(2 \times 10^{-4}, 6 \times 10^{-4})$, $R_i \sim U(0.3, 1)$, $T_i \sim U(2, 5)$, and $\omega_i \sim U(0, 2\pi)$ are all randomly generated from uniform distributions. Fig. 7 shows the statistically expected rate of return a over a time period.

The covariance matrix $Q = Q_1 + Q_2$ is randomly generated and fixed over time to compare the results directly, where $Q_1 = 0.01U^T U \geq 0$, $u_{ij} \sim U(-1, 1)$ are i.i.d (independent identically distributed) variables, and $Q_2 = \text{diag}\{R_1, R_2, \dots, R_5\}$:

$$Q = \begin{pmatrix} 1.849 & 0.548 & 0.089 & 0.437 & 0.207 \\ 0.548 & 2.159 & -0.011 & -1.155 & 0.004 \\ 0.089 & -0.011 & 1.697 & -0.010 & -0.547 \\ 0.437 & -1.155 & -0.010 & 3.252 & 0.372 \\ 0.207 & 0.004 & -0.547 & 0.372 & 1.523 \end{pmatrix}.$$

Then, based on the statistic results of the expected return rates and their covariance matrices, the probabilities of expected returning rate are maximized based on the proposed neurodynamic optimization model. Fig. 8 shows the transient states of the one-layer recurrent neural network (4) for portfolio optimization. Fig. 9 shows the optimal selection scheme of the five securities over time. The probabilities that rate of return greater or equal than an expected value ($r = 0.1$) is calculated as $P(\xi^T x^* \geq r) = \Phi((a^T x - r) / \sqrt{x^T Q x})$ and marked in dotted line with ‘+’ in the figure.

From Figs. 7 and 9, we can see that the selection rate x_i^* of a particular security i is higher if its expected rate of return a_i is larger and also with less fluctuation (small q_{ii}). On one hand, though a_4 is much larger than a_3 when $t > 25$, its optimal selection rate x_4^* is still quite low since its return rate fluctuates more than others (with larger q_{33}). On the other hand, though $a_3 < a_1$ all the time, $x_3^* > x_1^*$ sometimes since $q_{33} < q_{11}$. For some time (i.e., $5 < t < 25$), most of the rates of return for the securities are not large enough (< 0.14). As a result, the probability $P(\xi^T x^* \geq r)$ of earnings of 10 percent more ($r = 0.1$) is smaller than the ones on other time.

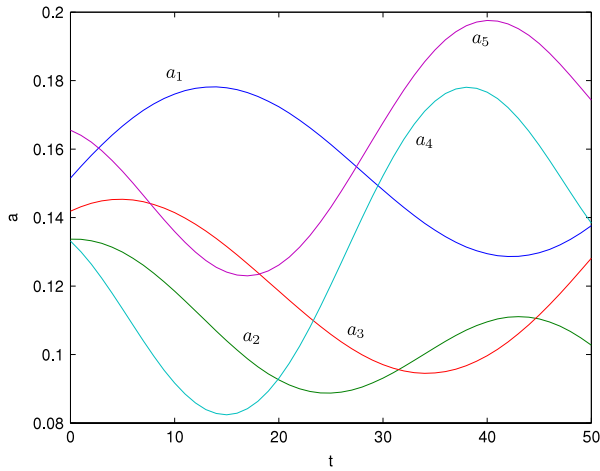


Fig. 7. Mean of expected rate of return a in Example 4.

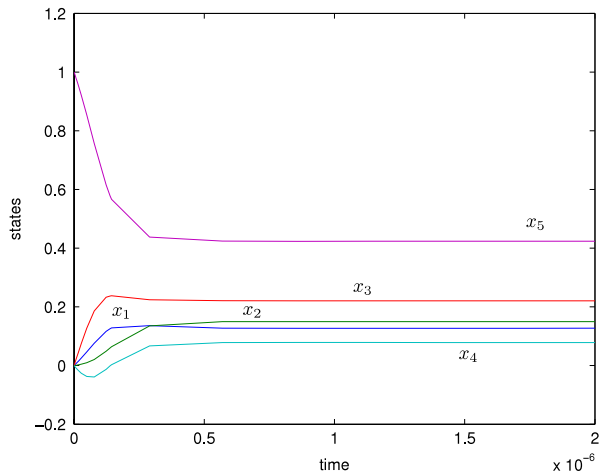


Fig. 8. Transient states of the neural network (4) in portfolio selection in Example 4.

Figs. 10 and 11 show the optimal portfolios as well as the possibilities $P(\xi^T x^* \geq r)$ for $r = 0.12$ and $r = 0.08$, respectively.

By comparing Figs. 9–11, we can see that even under the same condition (i.e., exactly the same securities), different expectations of returning rate lead to different optimal selections. If one is more greedy expecting higher rate of return (r), the possibility to achieve the goal is smaller. In one words, high rate of return usually comes along with high risk.

7. Conclusions

This paper presents a one-layer recurrent neural network with a simple structure for solving constrained pseudoconvex optimization problems. By properly setting two gain parameters beyond derived lower bounds, the finite-time convergence of any equilibrium point of the proposed neural network to solution feasibility is guaranteed. Furthermore, by using the Lyapunov method and differential inclusion theory, the convergence of the neural network to the solution optimality of pseudoconvex optimization problems is guaranteed subject to the same condition for feasibility. In contrast to existing neural networks for constrained optimization, the proposed neural network is capable of solving more general problems of pseudoconvex optimization with linear equality and bound constraints. Simulation results on numerical examples are given to illustrate the effectiveness and performance of the proposed neural network. In addition, the proposed recurrent neural network is

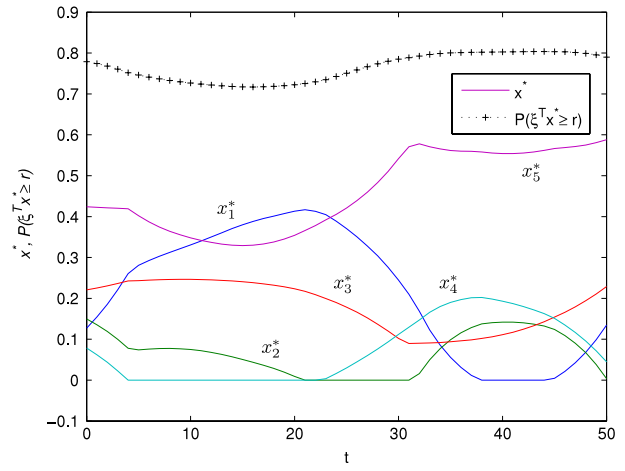


Fig. 9. Optimal portfolio selected by the neural network and the resulting probability when the minimum expected rate of return r is 0.1 in Example 4.

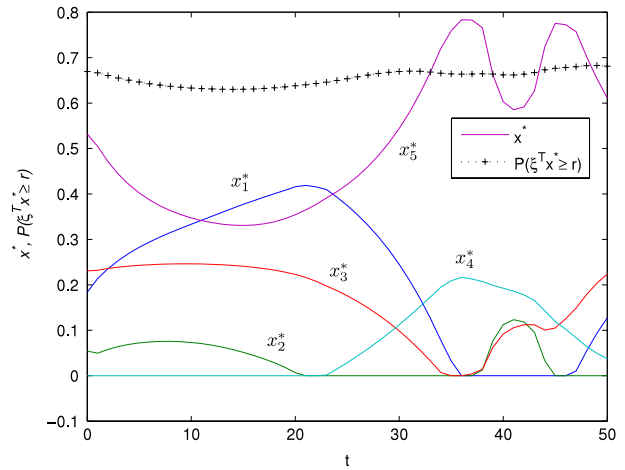


Fig. 10. Optimal portfolio selected by the neural network and the resulting probability when the minimum expected rate of return r is 0.12 in Example 4.

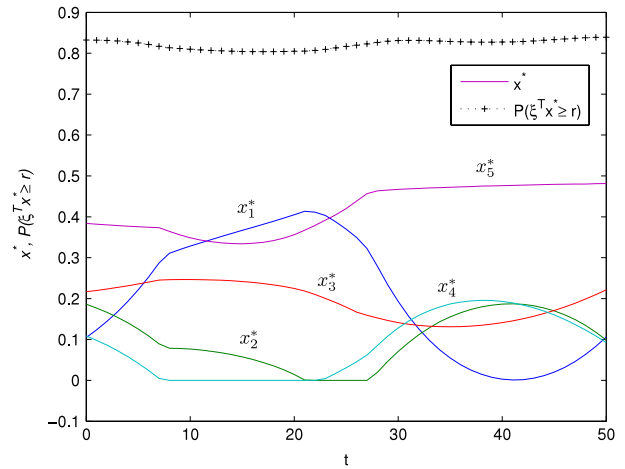


Fig. 11. Optimal portfolio selected by the neural network and the resulting probability when the minimum expected rate of return r is 0.08 in Example 4.

shown to be useful for dynamic portfolio optimization based on a stochastic portfolio model.

References

Aubin, J., & Cellina, A. (1984). *Differential inclusions: set-valued maps and viability theory*. New York: Springer-Verlag.

- Barbarosou, M., & Maratos, N. (2008). A nonfeasible gradient projection recurrent neural network for equality-constrained optimization problems. *IEEE Transactions on Neural Networks*, 19(10), 1665–1677.
- Bian, W., & Xue, X. (2009). Subgradient-based neural networks for nonsmooth nonconvex optimization problems. *IEEE Transactions on Neural Networks*, 20(6), 1024–1038.
- Chong, E., Hui, S., & Zak, S. (1999). An analysis of a class of neural networks for solving linear programming problems. *IEEE Transactions on Automatic Control*, 44(11), 1995–2006.
- Clarke, F. (1983). *Optimization and nonsmooth analysis*. New York: Wiley.
- Filippov, A. (1988). *Differential equations with discontinuous right hand sides. Mathematics and its applications (Soviet series)*. Boston: Kluwer Academic Publishers.
- Forti, M., Nistri, P., & Quincampoix, M. (2004). Generalized neural network for nonsmooth nonlinear programming problems. *IEEE Transactions on Circuits and Systems-I*, 51(9), 1741–1754.
- Forti, M., Nistri, P., & Quincampoix, M. (2006). Convergence of neural networks for programming problems via a nonsmooth Łojasiewicz inequality. *IEEE Transactions on Neural Networks*, 17(6), 1471–1486.
- Hu, X., & Wang, J. (2007b). A recurrent neural network for solving a class of general variational inequalities. *IEEE Transactions on Systems, Man and Cybernetics-B*, 37(3), 528–539.
- Hu, X., & Wang, J. (2007a). Design of general projection neural networks for solving monotone linear variational inequalities and linear and quadratic optimization problems. *IEEE Transactions on Systems, Man and Cybernetics-B*, 37(5), 1414–1421.
- Hu, X., & Wang, J. (2006). Solving pseudomonotone variational inequalities and pseudoconvex optimization problems using the projection neural network. *IEEE Transactions on Neural Networks*, 17(6), 1487–1499.
- Kennedy, M., & Chua, L. (1988). Neural networks for nonlinear programming. *IEEE Transactions on Circuits and Systems*, 35(5), 554–562.
- Kinderlehrer, D., & Stampacchia, G. (1982). *An introduction to variational inequalities and their applications*. New York: Academic.
- Liang, J., & Tang, W. (2001). Probability criterion in portfolio investment model with commission. *Systems Engineering*, 19(2), 5–10.
- Lillo, W., Loh, M., Hui, S., & Zak, S. (1993). On solving constrained optimization problems with neural networks: a penalty method approach. *IEEE Transactions on Neural Networks*, 4(6), 931–940.
- Liu, Q., & Cao, J. (2010). A recurrent neural network based on projection operator for extended general variational inequalities. *IEEE Transactions on Systems, Man and Cybernetics-B*, 40(3), 928–938.
- Liu, Q., Cao, J., & Chen, G. (2010). A novel recurrent neural network with finite-time convergence for linear programming. *Neural Computation*, 22(11), 2962–2978.
- Liu, B., & Ku, C. (1993). Probability criterion in inventory systems. *Journal of Systems Science and Mathematical Science*, 13(1), 70–75.
- Liu, S., & Wang, J. (2006b). A simplified dual neural network for quadratic programming with its kwta application. *IEEE Transactions on Neural Networks*, 17(6), 1500–1510.
- Liu, Q., & Wang, J. (2008a). A one-layer recurrent neural network with a discontinuous activation function for linear programming. *Neural Computation*, 20(5), 1366–1383.
- Liu, Q., & Wang, J. (2008b). A one-layer recurrent neural network with a discontinuous hard-limiting activation function for quadratic programming. *IEEE Transactions on Neural Networks*, 19(4), 558–570.
- Liu, Q., & Wang, J. (2006a). A recurrent neural network for non-smooth convex programming subject to linear equality and bound constraints. In *LNCS: Vol. 4233. Proc. 13th Int. conference on neural information processing* (pp. 1004–1013) Springer.
- Liu, Q., & Wang, J. (2009). A one-layer recurrent neural network for non-smooth convex optimization subject to linear equality constraints. In *LNCS: Vol. 5507. Proc. 15th int. conference on neural information processing (ICONIP2008). Part II* (pp. 1003–1010). Springer.
- Lu, W., & Chen, T. (2006). Dynamical behaviors of delayed neural network systems with discontinuous activation functions. *Neural Computation*, 18, 683–708.
- Marco, M., Forti, M., & Grazzini, M. (2006). Robustness of convergence in finite time for linear programming neural networks. *International Journal of Circuit Theory and Applications*, 34(3), 307–316.
- Markovitz, H. (1952). Portfolio selection. *Journal of Finance*, 7(1), 77–91.
- Markovitz, H. (1991). *Portfolio selection: efficient diversification of investments*. Cambridge, MA: B. Blackwell.
- Olsson, C., Eriksson, A., & Kahl, F. (2007). Efficient optimization for l_∞ -problems using pseudoconvexity. In *Proc. IEEE international conference on computer vision* (pp. 1–8).
- Penot, J., & Quang, P. (1997). Generalized convexity of functions and generalized monotonicity of set-valued maps. *Journal of Optimization Theory and Applications*, 92(2), 343–356.
- Tang, W., Wang, Y., & Liang, J. (2002). Fractional programming model for portfolio with probability criterion. In *Proc. IEEE international conference on systems, man and cybernetics* (pp. 516–519). Vol. 6.
- Tank, D., & Hopfield, J. (1986). 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.
- Wang, J. (1993). Analysis and design of a recurrent neural network for linear programming. *IEEE Transactions on Circuits and Systems-I*, 40(9), 613–618.
- Wang, J. (1994). A deterministic annealing neural network for convex programming. *Neural Networks*, 7(4), 629–641.
- Wang, J. (1997). Primal and dual assignment networks. *IEEE Transactions on Neural Networks*, 8(3), 784–790.
- Wang, J. (1998). Primal and dual neural networks for shortest-path routing. *IEEE Transactions on Systems, Man, and Cybernetics-A*, 28(6), 864–869.
- Xia, Y. (2003). Global convergence analysis of Lagrangian networks. *IEEE Transactions on Circuits and Systems-I*, 50(6), 818–822.
- Xia, Y. (1996). A new neural network for solving linear and quadratic programming problems. *IEEE Transactions on Neural Networks*, 7(6), 1544–1548.
- Xia, Y., Feng, G., & Wang, J. (2004). A recurrent neural network with exponential convergence for solving convex quadratic program and related linear piecewise equations. *Neural Networks*, 17(7), 1003–1015.
- Xia, Y., H., Leung, G., & J. Wang (2002). A projection neural network and its application to constrained optimization problems. *IEEE Transactions Circuits and Systems-I*, 49(4), 447–458.
- Xia, Y., & Wang, J. (2004b). A recurrent neural network for nonlinear convex optimization subject to nonlinear inequality constraints. *IEEE Transactions on Circuits and Systems-I*, 51(7), 1385–1394.
- Xia, Y., & Wang, J. (2004a). A one-layer recurrent neural network for support vector machine learning. *IEEE Transactions on Systems, Man and Cybernetics-B*, 34(2), 1261–1269.
- Xia, Y., & Wang, J. (2005). A recurrent neural network for solving nonlinear convex programs subject to linear constraints. *IEEE Transactions on Neural Networks*, 16(2), 379–386.
- Xue, X., & Bian, W. (2008). Subgradient-based neural networks for nonsmooth convex optimization problems. *IEEE Transactions on Circuits and Systems-I*, 55(8), 2378–2391.
- Zhang, S., & Constantinides, A. (1992). Lagrange programming neural networks. *IEEE Transactions on Circuits and Systems-II*, 39(7), 441–452.