Australian Journal of Basic and Applied Sciences, 5(5): 894-909, 2011
ISSN 1991-8178

A Recurrent Neural Network for Nonlinear Convex Optimization with Application to a Class of Variational Inequalities Problems

[1]Mohsen Alipour, [2]Davood Rostamy, [3]Alaeddin Malek

[1]Member of young research club, Islamic Azad University, Sari branch, P.O. Box 48164-194, Sari, Iran.
[2]Department of Mathematics, Imam Khomeini International University, P.O. Box 34149-16818, Ghazvin, Iran.
[3]Department of Applied Mathematics, Faculty of Mathematical Sciences, Tarbiat Modares University, P.O. Box 14115-134, Tehran, Iran.

Abstract: A novel recurrent artificial neural network is proposed in this paper for solving nonlinear convex programming with nonlinear inequality and linear equality constraints. We prove here that the proposed neural network is stable in the sense of Lyapunov and globally convergent to an optimal solution. As an application, we extend the proposed neural network for solving a class of variational inequality problems with nonlinear equality and inequality constraints. This neural network model has no adjustable parameter. The proposed neural network has a simpler structure and lower complexity for implementation than the existing neural networks for solving such problems. Compared with the existing convergence results, the present results do not require Lipschitz continuity condition on the objective function. Finally, some examples demonstrated to show the applicability of the proposed neural networks.

Key words: Neural networks, convex nonlinear programming, variational inequality problem, Lyapunov stable.

INTRODUCTION

The dynamical systems approach to solving constrained optimization problems was first proposed by Pyne (1956), and later studied by Rybashow (1965), Karpinskaya (1967) and others recently, due to renewed interest in neural networks.

In many practical optimization problems such as the planning of power systems and routing of telecommunication systems, the numbers of decision variables and constraints are usually very large. It is even more challenging when a large-scale optimization procedure has to be performed in real time to optimize the performance of a dynamical system. Therefore, neural network methods for the solution of optimization problems have been received considerable attention (Kennedy *et al.*, 1988; Maa *et al.*, 1992; Cichocki *et al.*, 1993; Zak *et al.*, 1995 and Xia., 1996). For the primal-dual solution of linear, quadratic and nonlinear programming problems see Malek's models in the references (Malek *et al.,* 2005 and 2007; Oskoei *et al.*, 2007; Yashtini *et al.*, 2007 and 2008).

In this paper, we propose a novel recurrent neural network for solving convex nonlinear programming problem with nonlinear inequality and linear equality constraints. As an application, we extend the proposed neural network for solving a class of variational inequality problems with nonlinear inequality and linear equality constraints. We prove here that the proposed neural network is stable in the sense of Lyapunov and globally convergent to an optimal solution.

The reminder of this paper is arranged as follows. In Section 2, we introduce the basic problem formulation. In Section 3, a neural network model for solving nonlinear convex programming problems is proposed. In Section 4, we consider analysis of the network dynamics and global convergence. In Section 5, we use an extension to the proposed neural network in section 3, for solving a class of variational inequality problems with nonlinear equality and inequality constraints. Finally, In Section 6, several examples are considered to evaluate the power and effectiveness of proposed neural network approach. Some conclusions are summarized in the last section.

Corresponding Author: Mohsen Alipour, Member of young research club, Islamic Azad University, Sari branch, P.O. Box 48164-194, Sari,Iran.
E-mail: m.alipour2323@gmail.com

Problem Formulation:

In this paper, the following constraint optimization problem is considered:

$$\begin{aligned} &\min \ f(x),\\ &s.t. \quad g(x)\ge 0,\\ &\qquad Ax=b,\\ &\qquad x\ge 0, \end{aligned} \tag{1}$$

where $A\in R^{r\times n}$ and $f(x)$ is twice continuously differentiable and convex from R^n to R and $\nabla^2 f$ (Hessian f) is positive definite and $g(x)=[g_1(x),...,g_m(x)]$ such that $g_i(x)$ (i=1,...,m) is twice continuously differentiable and concave from R^n to R. Let $\nabla f(x)$ and $\nabla g(x)$ be the gradient of $f(x)$ and Jacobian $g(x)$, respectively.
Define a Lagrange function of (1) below:

$$L(x,y,z)=f(x)-y^T(g(x))-z^T(Ax-b),$$

where $y\in R_+^m\left(R_+^m=\left\{y\ge 0\,\middle|\, y\in R^m\right\}\right)$, $z\in R^r$ are referred to as the Lagrange multipliers. According to Karush-Kuhn-Tucker (KKT) conditions (Bazaraa *et al.*, 1990), x^* is a solution to (1) if and only if exist $y^*\in R^m$, $z^*\in R^r$ such that (x^*,y^*,z^*) satisfies the following conditions:

$$\begin{cases} \nabla f(x^*)-\nabla g(x^*)^T y^*-A^T z^*\ge 0,\\ x^{*T}(\nabla f(x^*)-\nabla g(x^*)^T y^*-A^T z^*)=0,\\ g(x^*)\ge 0,\\ y^{*T}(g(x^*))=0,\\ Ax^*-b=0,\\ x^*\ge 0,\\ y^*\ge 0. \end{cases} \tag{2}$$

Lemma 1.

Let Ω be a closed convex set of R^k. Then

$$(v-p_\Omega(v))^T(p_\Omega(v)-x)\ge 0, \qquad v\in R^k,\ x\in\Omega,$$
$$\|p_\Omega(u)-p_\Omega(v)\|\le\|u-v\|, \qquad u,v\in R^k,$$

where $\|.\|$ denote 2-norm and the projection operator $p_\Omega(u)$ is defined by

$$p_\Omega(u)=\arg\min_{v\in\Omega}\|u-v\|.$$

Proof:

See (Bertsekas *et al.*, 1989).

Lemma 2:

Let $H:R^k\to R^k$, be a continuous function and Ω is subset of R^k, then u^* satisfies $(u-u^*)^T H(u^*)\ge 0$ for all $u\in\in R^k$ if and only if

$$-u^*+p_\Omega(u^*-H(u^*))=0.$$

Proof:

See (Bertsekas *et al.*, 1989).

Theorem 1:

x^* is a solution to if and only if there exist $y^* \in R_+^m$, $z^* \in R^r$ such that (x^*, y^*, z^*) satisfies

$$\begin{cases} -x^* + (x^* - \nabla f(x^*) + \nabla g(x^*)^T y^* + A^T z^*)^+ = 0, \\ -y^* + (y^* - g(x^*))^+ = 0, \\ b - Ax^* = 0, \end{cases} \tag{3}$$

where $(x)^+ = \left[(x_1)^+, \ldots, (x_n)^+\right]^T$, and $(x_i)_\cdot^+ = \max\{0, x_i\}$

Proof:

According to KKT saddle point theorem (Bazaraa *et al.*, 1990), x^* is a solution of problem (1) if and only if there exist $y^* \in R_+^m$, $z^* \in R^r$such that $(x^*, y^*, z^*)^T$ is a saddle point of $L(x,y,z)$ on $R_+^n \times R_+^m \times R^r$. That is

$$L(x^*, y, z) \le L(x^*, y^*, z^*) \le L(x, y^*, z^*) \qquad \forall (x, y, z)^T \in R_+^n \times R_+^m \times R^r,$$

therefore

$$f(x^*) - y^T (g(x^*)) - z^T (Ax^* - b) \le f(x^*) - y^{*T} (g(x^*)) - z^{*T} (Ax^* - b)$$
$$\le f(x) - y^{*T} (g(x)) - z^{*T} (Ax - b)$$

Then from left hand side we get:

$$(y - y^*)^T (g(x^*)) + (z - z^*)^T (Ax^* - b) \ge 0, \ \forall (y, z)^T \in R_+^m \times R^r.$$

and from right hand side we have:

$$f(x^*) - y^{*T} (g(x^*)) - z^{*T} (Ax^* - b) \le f(x) - y^{*T} (g(x)) - z^{*T} (Ax - b), \quad \forall x \in R_+^n.$$

Let $G(x) = f(x) - y^{*T} (g(x)) - z^{*T} (Ax - b)$, therefore $G(x^*) \le G(x), \forall x \in R_+^n$. Since $f(x)$ is convex and $g_i(x) (i = 1, \ldots, m)$ is concave and $y \in R_+^m$, therefore $G(x)$ is a convex function. Then it implies that (Bazaraa *et al.*, 1990)

$$(x - x^*)^T (\nabla G(x^*)) \ge 0,$$

or

$$(x - x^*)^T (\nabla f(x^*) - \nabla g(x^*)^T y^* - A^T z^*) \ge 0.$$

So

$$\begin{cases} (x - x^*)^T (\nabla f(x^*) - \nabla g(x^*)^T y^* - A^T z^*) \ge 0, \ \forall x \in R_+^n, \\ (y - y^*)^T (g(x^*)) + (z - z^*)^T (Ax^* - b) \ge 0, \ \forall (y, z) \in R_+^m \times R^r. \end{cases} \tag{4}$$

We define $H : R^{n+m+r} \to R^{n+m+r}$ as follows:

$$H(u) = \begin{pmatrix} \nabla f(x) - \nabla g(x)^T y - A^T z \\ g(x) \\ Ax - b \end{pmatrix}, \tag{5}$$

where $u=(x,y,z)^T \in R_+^n \times R_+^m \times R^r$, then (4) is equivalently written as follows:

$$(u-u^*)^T H(u^*) \geq 0,\ \forall u \in R_+^n \times R_+^m \times R^r. \tag{6}$$

then using Lemma 2, $u^*=(x^*,y^*,z^*)^T$ is a solution of (6) if and only if satisfies $-u^*+(u^*-H(u^*))^+=0$.

Definition 1:
A vector $u^*=u(t^*)$ is an equilibrium point, or steady state of the dynamic system $\frac{du(t)}{dt}=H(u(t))$ at time $t^* \in R_+$ if $H(u(t))=0$ for all $t \geq t^*$.

Definition 2:
A set *S* of points in R^k is invariant with respect to system $\frac{du(t)}{dt}=H(u(t))$ if every solution of dynamic system starting in *S* remains in *S* for all $t \geq 0$.

Definition 3:
Let $S \subset R^k$ be an open neighborhood of *u**. An Energy function or Lyapunov function is a continuously differentiable function $E: R^k \to R$, such that satisfies the following conditions:

i) $E(u) \geq 0,\ \forall u \in S. E(u^*)=0$ and $E(u)>0,\ \forall u \in s, u \neq u^*$.

ii) $\frac{dE(u(t))}{dt}=\left(\nabla_{u(t)}E(u(t))\right)^T \frac{du(t)}{dt}=\left(\nabla_{u(t)}E(u(t))\right)^T H(u(t)) \leq 0$ and $\frac{dE(u^*)}{dt}=0$

Definition 4:
An equilibrium point *u** of dynamic system $\frac{du(t)}{dt}=H(u(t))$ is a Lyapunov stable point, if there exist an associated Lyapunov function that satisfies in Definition 3.

Definition 5:
A mapping $F: R^n \to R^n$ is monotone on set *S* if

$$(x-y)^T (F(x)-F(y)) \geq 0,\ \forall x, y \in S.$$

F is strictly monotone on set *S*, if strict inequality holds whenever $x \neq y$.

Neural network model:
Now, let *x*(*t*), *y*(*t*), *z*(*t*) to be time dependent variables. We can describe the neural network model by the following nonlinear dynamic system, for solving (1):

$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -x+(x-\nabla f(x)+\nabla g(x)^T y+A^T z)^+ \\ -y+(y-g(x))^+ \\ b-Ax \end{pmatrix} \tag{7}$$

The dynamic system (7) can be easily realized in a recurrent neural network as shown in Fig. 1.

Lemma 3:
Let $S^*=\left\{(x^*,y^*,z^*) \in R^{n+m+r} \,\middle|\, (x^*,y^*,z^*)\ solve\ (3)\right\}$, then $(x^*,y^*,z^*) \in S^*$ if and only if (x^*,y^*,z^*) is a equilibrium point of network (7).

Proof:
It is a direct consequence of Theorem 1.

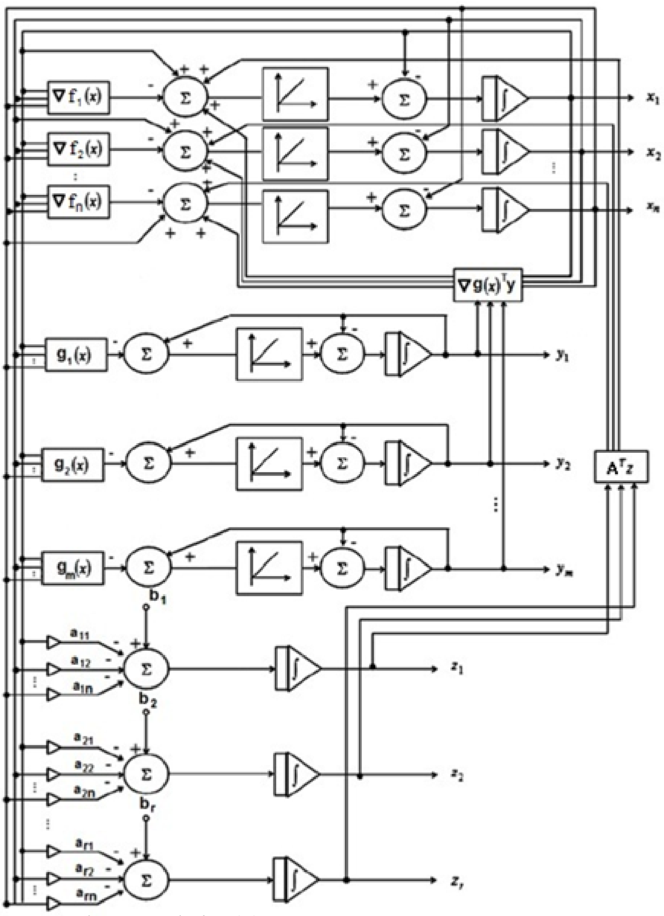

Fig. 1: Architecture of the neural network in (7)

Theorem 2:

For any initial point $u_0 = (x_0, y_0, z_0) \in R_+^n \times R_+^m \times R^r$:

i) there exists a unique continuous solution $u(t) = (x(t), y(t), z(t))$ for (7). Moreover, $x(t) \geq 0,\ y(t) \geq 0.$

ii) If (x^*, y^*, z^*) be the equilibrium point of (7), then x^* solve (1).

Proof i):

From the existence and uniqueness theorem of the initial value problem of a system of differential equations (Miller *et al.,* 1982), we only need to prove that the mapping $V(u) = -u + (u - H(u))^+$ is locally Lipschitz. Now by Lemma 1 we have

$$\|V(u_1) - V(u_2)\| = \|(u_1 - H(u_1))^+ - (u_2 - H(u_2))^+ - (u_1 - u_2)\| \leq$$
$$\|(u_1 - H(u_1))^+ - (u_2 - H(u_2))^+\| + \|u_1 - u_2\| \leq$$
$$2\|u_1 - u_2\| + \|H(u_1) - H(u_2)\|,\ \forall u_1, u_2 \in R_+^{n+m}.$$

Since $H(u)$ is a continuously differentiable mapping, thus $H(u)$ is Lipschitz continuous, that is there exists c>0 such that $\|H(u_1) - H(u_2)\| \leq c\|u_1 - u_2\|$. Hence

$$\|V(u_1)-V(u_2)\| \le (2+c)\|u_1-u_2\|,\ \forall u_1,u_2 \in R_+^{n+m},$$

that is to say $V(u)$ is locally Lipschitz.
Next, note that

$$\frac{dx}{dt} = -x + (x - \nabla f(x) + \nabla g(x)^T y + A^T z)^+,$$

$$\frac{dy}{dt} = -y + (y - g(x))^+.$$

i.e.

$$\frac{dx}{dt} + x = (x - \nabla f(x) + \nabla g(x)^T y + A^T z)^+,$$

$$\frac{dy}{dt} + y = (y - g(x))^+.$$

Multiplying both side of the above equations into e^t and integration with respect to t over interval $[t_0, t]$, we have

$$e^t x(t) - e^{t_0} x(t_0) = \int_{t_0}^{t} e^s (x - \nabla f(x) + \nabla g(x)^T y + A^T z)^+ ds,$$

$$e^t y(t) - e^{t_0} y(t_0) = \int_{t_0}^{t} e^s (y - g(x))^+ ds.$$

Therefore

$$x(t) = e^{(t_0 - t)} x(t_0) + e^{-t} \int_{t_0}^{t} e^s (x - \nabla f(x) + \nabla g(x)^T y + A^T z)^+ ds \ge 0,$$

$$y(t) = e^{(t_0 - t)} y(t_0) + e^{-t} \int_{t_0}^{t} e^s (y - g(x))^+ ds \ge 0.$$

Since two terms of the right hand side are nonnegative.

ii) By Theorem 1 and Lemma 3.

Analysis of the Network Dynamics and Global Convergence:

In this section, we shall study the dynamics of network (7) for solving problem (1).

Lemma 4:

Let S be a nonempty open convex set in R^n and $f : S \to R$ be differentiable on S, then f is convex if and only if

$$(x_1 - x_2)^T (\nabla f(x_1) - \nabla f(x_2)) \ge 0,\ \forall x_1, x_2 \in S.$$

Proof:

See (Bazaraa *et al.*, 1990).

Lemma 5:

Let S be a nonempty open convex set in R^n and $f : S \to R$ be differentiable on S, then f is convex if and only if

$$f(x_1) - f(x_2) - (x_1 - x_2)^T \nabla f(x_2) \ge 0,\ \forall x_1, x_2 \in S.$$

Proof:

See (Bazaraa *et al.*, 1990).

Lemma 6:
$H(u)$ in (5) is a monotone mapping on $\Omega = R_+^n \times R_+^m \times R^r$.

Proof:

$\forall u_1, u_2 \in R_+^{n+m}$, we have

$$(u_1 - u_2)^T (H(u_1) - H(u_2)) = (x_1 - x_2)^T (\nabla f(x_1) - \nabla f(x_2)) -$$
$$(x_1 - x_2)^T (\nabla g(x_1)^T y_1 - \nabla g(x_2)^T y_2) - (x_1 - x_2)^T (A^T z_1 - A^T z_2) +$$
$$(y_1 - y_2)^T (g(x_1) - g(x_2)) + (z_1 - z_2)^T (Ax_1 - Ax_2)$$
$$= (x_1 - x_2)^T (\nabla f(x_1) - \nabla f(x_2)) + y_1^T (g(x_1) - g(x_2) - \nabla g(x_1)(x_1 - x_2)) -$$
$$y_2^T (g(x_1) - g(x_2) - \nabla g(x_2)(x_1 - x_2)).$$

Since f and $-g_i \ (i = 1, \ldots, m)$are convex functions on R^n and using Lemma 4 and 5 and since $y_1, y_2 \in R_+^m$, we have $(u_1 - u_2)^T (H(u_1) - H(u_2)) \geq 0$.

Theorem 3:
Let $u^* = (x^*, y^*, z^*)$ is an equilibrium point of (7) where x^* is an optimal solution of (1). The proposed neural network of (7) with the initial point $u_0 = (x_0, y_0, z_0) \in \Omega$ is stable in the Lyapunov sense and globally convergent to u^*.

Proof:
First, from Theorem 2 it follows that $u(t) \in \Omega, \forall t \geq t_0$.

We define $E : \Omega \to R$ as follows:

$$E(u) = -H(u)^T V(u) - \frac{1}{2}\|V(u)\|^2 + \frac{1}{2}\|u - u^*\|^2,$$

where $V(u) = -u + (u - H(u))^+$. We show that $E(u)$ is a suitable Lypunov function for dynamic system (7). From Gao *et al.* (2003) we have

$$\nabla E(u) = H(u) - (\nabla H(u) - I)V(u) + u - u^*$$

where $\nabla H(u)$ denotes the Jacobian of H, and

$$\nabla H(u) = \begin{pmatrix} \nabla^2 f(x) & -\nabla g(x)^T & -A^T \\ \nabla g(x) & 0 & 0 \\ A & 0 & 0 \end{pmatrix}.$$

Therefore, $\nabla H(u)$ is positive semi-definite. Then

$$\frac{dE(u)}{dt} = \nabla E(u)^T \frac{du}{dt} = \left(H(u) - (\nabla H(u) - I)V(u) + u - u^*\right)^T V(u) = \quad (8)$$
$$\left(H(u) + u - u^*\right)^T V(u) + \|V(u)\|^2 - V(u)^T \nabla H(u) V(u).$$

By the results given in Pang (1987) we know that

$$(V(u) + u - u^*)^T (-V(u) - H(u)) \geq 0,$$
$$H(u)^T V(u) \leq -\|V(u)\|^2.$$

Therefore

$$\left(H(u)+u-u^*\right)^T V(u) \le -H(u)^T (u-u^*) - \|V(u)\|^2, \tag{9}$$

$$-H(u)^T V(u) - \frac{1}{2}\|V(u)\|^2 \ge 0. \tag{10}$$

From (10) it is obvious that $E(u)$ is nonnegative and $E(u^*)=0$. In the following, we show that $\frac{dE(u)}{dt} \le 0$. From (8) and (9) we have

$$\begin{aligned}\frac{dE(u)}{dt} &\le -H(u)^T (u-u^*) - \|V(u)\|^2 + \|V(u)\|^2 - V(u)^T \nabla H(u) V(u) = \\ &-H(u)^T (u-u^*) - V(u)^T \nabla H(u) V(u),\end{aligned} \tag{11}$$

Now it is enough to show that both terms in the right hand side are negative values. The second term is negative since $\nabla H(u)$ is positive semi-definite. From Lemma 6 we know

$$(u-u^*)^T (H(u)-H(u^*)) \ge 0$$

i.e.

$$(u-u^*)^T (H(u)-H(u^*)) = (u-u^*)^T H(u) - (u-u^*)^T H(u^*) \ge 0$$

therefore

$$(u-u^*)^T H(u) \ge (u-u^*)^T H(u^*)$$

from (4) , we have $(u-u^*)^T H(u^*) \ge 0,\ \forall u \in R_+^n \times R_+^m \times R^r$. thus, it implies that

$$(u-u^*)^T H(u) \ge 0,\ \forall u \in \Omega.$$

Therefore, we have

$\frac{dE(u)}{dt} \le 0,\ \forall u \in \Omega$ .Thus function $E(u)$ is monotonically non-increasing for all $t \ge t_0$, therefore trajectory $u(t)$ is bounded, since

$$E(u(t_0)) \ge E(u(t)) \ge \frac{1}{2}\|u(t)-u^*\|^2, \quad \forall t \ge t_0.$$

Thus the set $\mathrm{N} = \{u(t) \in \Omega \,|\, E(u(t)) \le E(u(t_0)),\ \forall t \ge t_0\}$ is bounded.

Now, we show the solution trajectories of the neural network (7) is globally convergent to u^*. By the invariant set Theorem in Golden (1996), trajectories $u(t)$ of neural network (7) converge to a largest invariant set in $\Psi = \left\{u \in \mathrm{N} \,\middle|\, \frac{dE(u)}{dt} = 0\right\}$. We now prove that $\frac{dE(u)}{dt} = 0 \Leftrightarrow \frac{du}{dt} = 0.$ Clearly, if $\frac{du}{dt} = 0,$ then $\frac{dE(u)}{dt} = \nabla E(u)^T \frac{du}{dt} = 0.$ To prove converse, let $\tilde{u} = (\tilde{x}, \tilde{y}, \tilde{z}) \in \Psi$, thus $\frac{dE(\tilde{u})}{dt} = 0.$ From (11) we have

$$H(\tilde{u})^T (\tilde{u}-u^*) + V(\tilde{u})^T \nabla H(\tilde{u}) V(\tilde{u}) = 0.$$

Since $\nabla H(\tilde{u})$ is positive semi-definite and $H(\tilde{u})^T (\tilde{u}-u^*) \ge 0, (12)$ implies that

$V(\tilde{u})^T \nabla H(\tilde{u})V(\tilde{u}) = 0$ and $H(\tilde{u})^T(\tilde{u}-u^*) = 0$, also $(H(\tilde{u}) - H(u^*))^T(\tilde{u}-u^*) = 0.$ Now, we have

$$V(\tilde{u})^T \nabla H(\tilde{u})V(\tilde{u}) = ((\tilde{x} - \nabla f(\tilde{x}) + \nabla g(\tilde{x})^T \tilde{y} + A^T \tilde{z})^+ - \tilde{x})^T \nabla^2 f(\tilde{x})$$
$$((\tilde{x} - \nabla f(\tilde{x}) + \nabla g(\tilde{x})^T \tilde{y} + A^T \tilde{z})^+ - \tilde{x}) = 0.$$

The positive definiteness of $\nabla^2 f(\tilde{x})$ implies

$(\tilde{x} - \nabla f(\tilde{x}) + \nabla g(\tilde{x})^T \tilde{y} + A^T \tilde{z})^+ - \tilde{x} = 0,$ therefore $\frac{d\tilde{x}}{dt} = 0.$ On the other hand,

$(H(\tilde{u}) - H(u^*))^T(\tilde{u}-u^*) = 0$ and using proof of Lemma 6, we have

$(\nabla f(\tilde{x}) - \nabla f(x^*))^T(\tilde{x}-x^*) = 0.$ Therefore we have

$$(\nabla f(\tilde{x}) - \nabla f(x^*))^T(\tilde{x}-x^*) = (\tilde{x}-x^*)^T \nabla^2 f(x_\lambda)(\tilde{x}-x^*) = 0,$$

where $x_\lambda = \lambda\tilde{x} + (1-\lambda)x^*$ and $0 \le \lambda \le 1$. The positive definiteness of $\nabla^2 f(\tilde{x})$ implies that $\tilde{x} = x^*$, then $b - A\tilde{x} = b - Ax^* = 0,$ Therefore we have $\frac{d\tilde{z}}{dt} = 0.$

Since $H(\tilde{u})^T(\tilde{u}-u^*) = 0,$ therefore

$$H(\tilde{u})^T(\tilde{u}-u^*) = (\nabla f(\tilde{x}) - \nabla g(\tilde{x})^T \tilde{y} - A^T \tilde{z})^T(\tilde{x}-x^*) + g(\tilde{x})^T(\tilde{y}-y^*) + (A\tilde{x}-b)^T \tilde{z} = 0.$$

Since $\tilde{x} = x^*$ we can write, $g(\tilde{x})^T(\tilde{y}-y^*) = 0.$ According to the KKT conditions in (2) we have $g(x^*)^T y^* = 0,$ thus $g(\tilde{x})^T \tilde{y} = g(\tilde{x})^T y^* = g(x^*)^T y^* = 0.$ Therefore from $\tilde{y} \ge 0,$ $g(\tilde{x}) = g(x^*) \ge 0$ and $g(\tilde{x})^T \tilde{y} = 0,$ we have $-\tilde{y} + (\tilde{y} - g(\tilde{x}))^+ = 0,$ i.e. $\frac{d\tilde{y}}{dt} = 0.$ So $\frac{du}{dt} = 0 \Leftrightarrow \frac{dE(u)}{dt} = 0,$ and the proposed neural network in (7) is globally convergent to the optimal solution of (1). This completes proof.

The Neural Network for Variational Inequalities Problem:

Consider the following variational inequality problem

$$(x - x^*)^T F(x^*) \ge 0, \ \forall x \in \Omega, \tag{13}$$

where $\Omega = \left\{ x \in R^n \,\middle|\, g(x) \ge 0, \ Ax = b, \ x \ge 0 \right\}$ and $F : R^n \to R^n$ is a monotone mapping, continuously differentiable and ∇F (Jacobian F) is positive definite on R_+^n , and $g(x) = [g_1(x), \ldots, g_m(x)]$ such that $g_i(x)\,(i = 1,\ldots,m)$ is twice continuously differentiable and concave from R^n to R.

By the KKT conditions for (13) we know that $x^* \in R_+^n$ is a solution of (13) if and only if there exists $(y,z)^T \in R_+^m \times R^r$ such that $u^* = (x^*, y^*, z^*)^T$ is a solution of the following variational inequalities problem

$$(u - u^*)^T G(u^*) \ge 0, \ \forall u \in \Omega, \tag{14}$$

where

$$G(u)=\begin{pmatrix} F(x)-\nabla g(x)^T y - A_Z^T \\ g(x) \\ Ax-b \end{pmatrix}.$$

According to Lemma 2, $u^*\in\Omega$ is solution (14) if and only if u^* satisfies in the following equation

$$-u^* + (u^* - G(u^*))^+ = 0. \tag{15}$$

We can equivalently write (15) as below

$$\begin{cases} -x^* + (x^* - F(x^*) + \nabla g(x^*)^T y^* + A^T z)^+ = 0, \\ -y^* + (y^* - g(x^*))^+ = 0, \\ b - Ax^* = 0. \end{cases} \tag{16}$$

Thus, as extension of the proposed neural network in (7) we have the following neural network model for solving (13)

$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -x + (x - F(x) + \nabla g(x)^T y + A^T z)^+ \\ -y + (y - g(x))^+ \\ b - Ax \end{pmatrix} \tag{17}$$

Theorem 4:
Let $u^* = (x^*, y^*, z^*)$ is an equilibrium point of (17) where x^* is an optimal solution of (13). The proposed neural network of (17) with the initial point $u_0 = (x_0, y_0, z_0) \in \Omega$ is stable in the Lyapunov sense and globally convergent to u^*.

Proof:
Its proof is similar to proof of Theorem 3.

Simulation examples:
In this section, to demonstrate the behavior and properties of the proposed neural networks in (7) and (17), several numerical examples are discussed. The simulation is conducted in MATLAB. In examples 1-4, $\nabla^2 f(x)$ and in examples 5-6, $\nabla F(x)$ are positive definite on R_+^n.

Example 1:
Consider the following nonlinear convex programming problem:

$$\begin{aligned} \min\ & f(x) = \frac{1}{4}x_1^4 + 0.5x_1^2 + \frac{1}{4}x_2^4 + 0.5x_2^2 - 0.9x_1x_2, \\ s.t.\ & -x_1 - x_2 \ge -2, \\ & x_1 - x_2 \ge -2, \\ & -x_1^2 - x_2 \ge -3, \\ & x_1 - 3x_2 = -2, \\ & x \ge 0. \end{aligned}$$

This problem has an optimal solution $x^* = (0.3461, 0.7820)^T$. We use the proposed neural network in (7) to solve the above problem. Theorem 3 guarantees that neural network (7) converges to exact solution x^*. Fig. 2 displays the convergence behavior with five different initial points and Fig. 3 displays the transient behavior of $x(t)$ with four initial points $A(3,-3)$, $B(3,3)$, $C(-3,2)$ and $D(-3,-2)$, where $y=(0,0,0)^T$ and $z=0$ are fixed values.

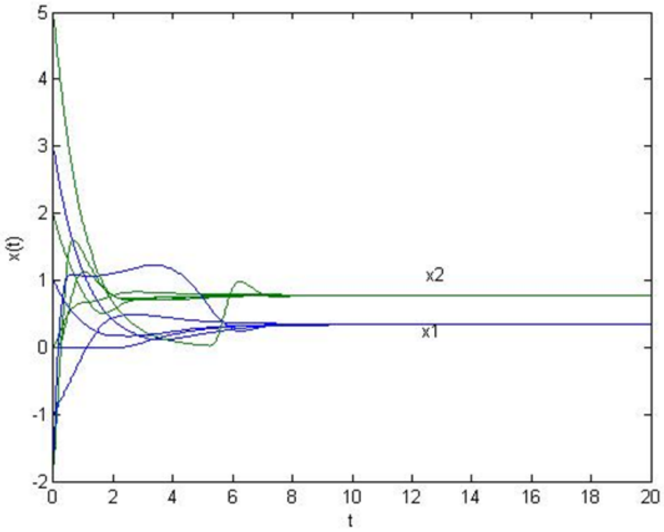

Fig. 2: Trajectories of the neural network (7) for five different initial points in Example 1.

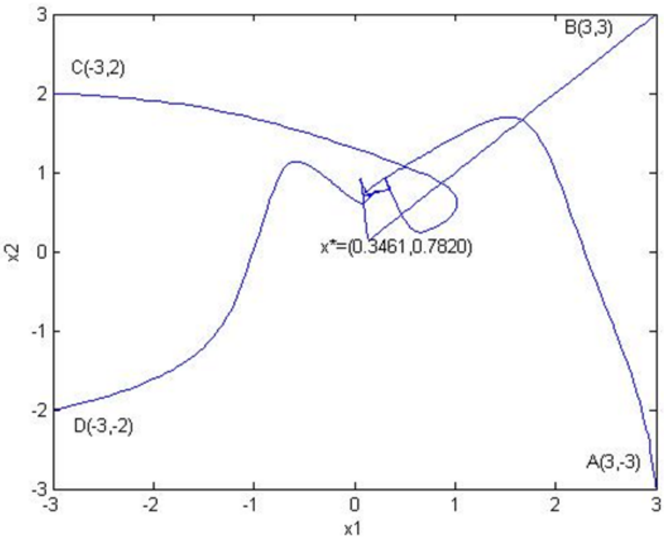

Fig. 3: Simulation results for the neural network (7) using four different initial points for Example 1.

Example 2:

Consider the following nonlinear convex programming problem:

$$\min\ f(x) = 0.4x_1 + x_1^2 + x_2^2 - x_1x_2 + 0.5x_3^2 + 0.5x_4^2 + \frac{x_1^3}{30}$$

$$\begin{aligned} s.t. \quad & -0.5x_1 - x_2 + x_4 \geq -0.5, \\ & -3x_1^2 + 2x_1x_2 \geq -1, \\ & x_1 + 0.5x_2 - x_3 = 0.4, \\ & x \geq 0. \end{aligned}$$

This problem has an optimal solution $x^* = (0.2567, 0.2867, 0, 0)^T$. To observe the convergent behavior of the proposed network in (7), we generated five different initial points and we see results in Fig. 4. Theorem 3 guarantees that neural network (7) converges to exact solution x^*.

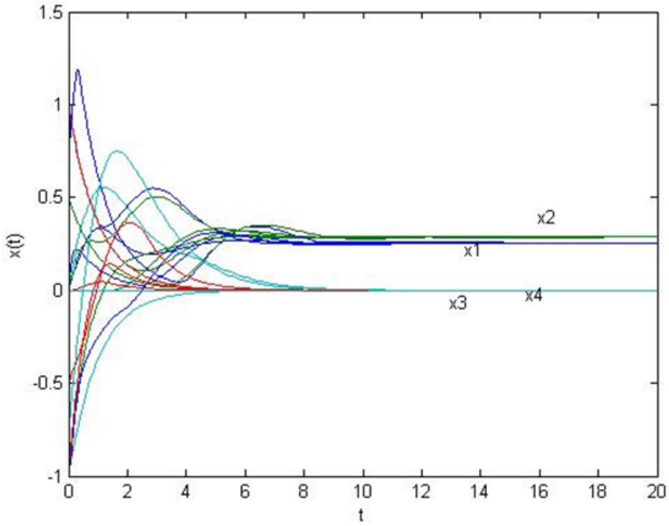

Fig. 4: Trajectories of the neural network (7) for five different initial points in Example 2.

Example 3:

Consider the following constraint optimization problem:

$$\min\ f(x) = x_1^2 + x_2^2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7x_8^2 + 2x_9^2 + (x_{10} - 7)^2 + x_1 x_2 - 14x_1 - 16x_2,$$

$$\begin{aligned} s.t.\quad & 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 \le 120, \\ & 5x_1^2 + 4(x_3 - 6)^2 + 8x_2 - 2x_4 \le 40, \\ & (x_1 - 8)^2 / 2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 \le 30, \\ & x_1^2 + 2(x_2 - 2)^2 - 2x_1 x_2 + 14x_5 - 6x_6 \le 0, \\ & 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 105, \\ & 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0, \\ & 12(x_9 - 8)^2 - 3x_1 + 6x_2 - 7x_{10} \le 0, \\ & -8x_1 + 2x_2 + 5x_9 - 2x_{10} \le 12, \\ & x \ge 0. \end{aligned}$$

The nonlinear programming is a convex optimization and it can be solved by the proposed neural network in (7). Theorem 3 guarantees that neural network (7) converges to exact solution $x^* = (1.8389, 3.3026, 5.1275, 1.4294, 0, 0, 6.0187, 8.7721)^T$. Fig. 5 displays the convergence behavior with four different initial points.

Example 4:

Consider the following nonlinear convex programming with nonlinear constraints:

$$\min\ f(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6 x_7 - 10x_6 - 8x_7,$$

$$\begin{aligned} s.t.\quad & -2x_1^2 - 3x_2^4 - x_3 - 4x_4^2 - 5x_5 + 127 \ge 0, \\ & -7x_1 - 3x_2 - 10x_3^2 - x_4 + x_5 + 282 \ge 0, \\ & -23x_1 - x_2^2 - 6x_6^2 + 8x_7 + 196 \ge 0, \\ & -4x_1^2 - x_2^2 - 2x_3^2 + 3x_1 x_2 - 5x_6 + 11x_7 \ge 0, \\ & x \ge 0. \end{aligned}$$

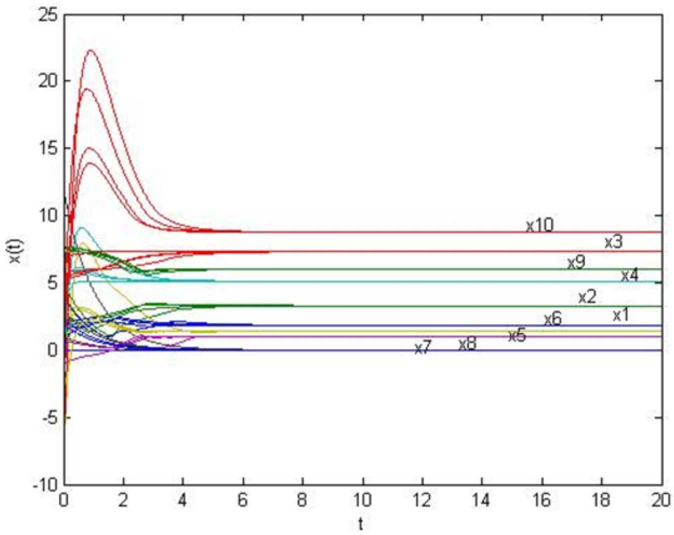

Fig. 5: Trajectories of the neural network (7) for four different initial points in Example 3.

This problem has an optimal solution $x^* = (1.3288, 2.4413, 0, 2.0559, 0, 1.1323, 1.4632)^T$. We apply the proposed neural network in (7) to solve above problem. Theorem 3 guarantees that neural network (7) converges to exact solution. Fig. 6 displays the convergence behavior with four different initial points.

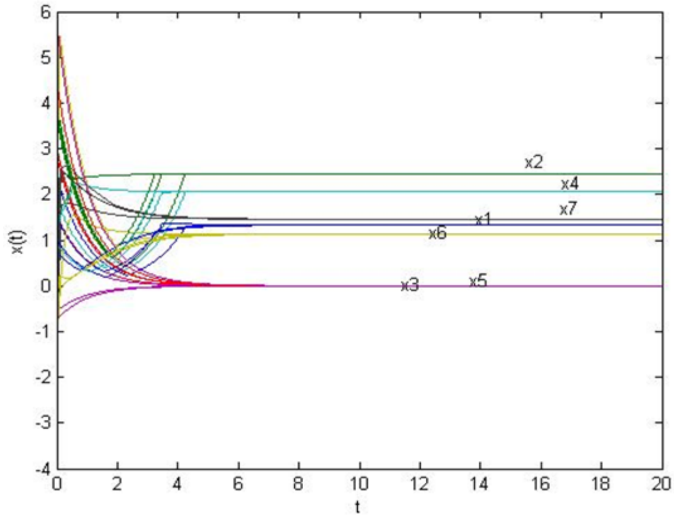

Fig. 6: Trajectories of the neural network (7) for four different initial points in Example 4.

Example 5:

Consider the following nonlinear variational inequality problem with the mapping F and the constraint set Ω:

$$F(x) = \begin{bmatrix} 3x_1 - 1/x_1 + 3x_2 - 2 \\ 3x_1 + 3x_2 \\ 4x_3 + 4x_4 \\ 4x_3 + 4x_4 - 1/x_4 - 3 \end{bmatrix}, \quad \Omega = \left\{ x \in R^4 \,\middle|\, x_3^2 + x_4 \geq 0,\ x_1 + x_2 = 1,\ x \geq 0 \right\}.$$

This problem has one solution $x^* = (1, 0, 0, 1)^T$. We use the proposed neural network in (17) to solve the above problem. Theorem 4 guarantees that neural network (17) converges to exact solution x^*. Fig. 7 and 8 displays the convergence behavior with the feasible initial value (0.75, 0.25, 1, 2, 0, 0) and the infeasible initial value (-2, 2, 1, -4, 0, 0), respectively.

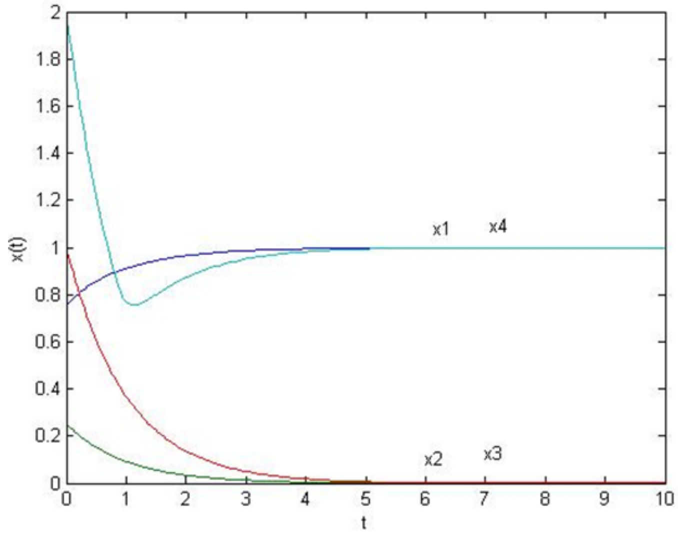

Fig. 7: Numerical results of the neural network (17) in Example 5 for the feasible initial point $(0.75, 0.25, 1, 2, 0, 0)^T$.

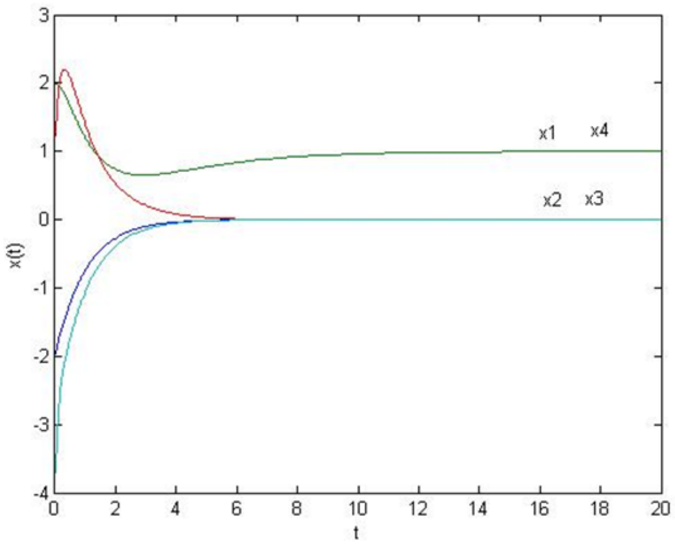

Fig. 8: Numerical results of the neural network (17) in Example 5 for the infeasible initial point $(-2, 2, 1, -4, 0, 0)^T$.

Example 6:

Consider the variational inequality problem (13), where:

$$F(x) = \begin{bmatrix} 10x_1 - 2x_2 + 2x_3 \\ -2x_1 + 26x_2 + 6x_3 \\ 2x_1 + 6x_2 + 2x_3 \end{bmatrix},$$

$$\Omega = \left\{ x \in R^3 \,\middle|\, -x_1^3 + 6x_2 + 4x_3 \geq -3 \text{ , } x_1 + x_2 + x_3 = 1, \ x \geq 0 \right\}.$$

This problem has an optimal solution $x^* = (0, 0 ,1)^T$. To observe the convergent behavior of the proposed network in (17), we generated four different initial points and see results in Fig. 9. Theorem 4 guarantees that neural network (17) converges to exact solution x^*. Also Fig. 10. depict four trajectories starting with fixed values y=1 and z=-1 and various x.

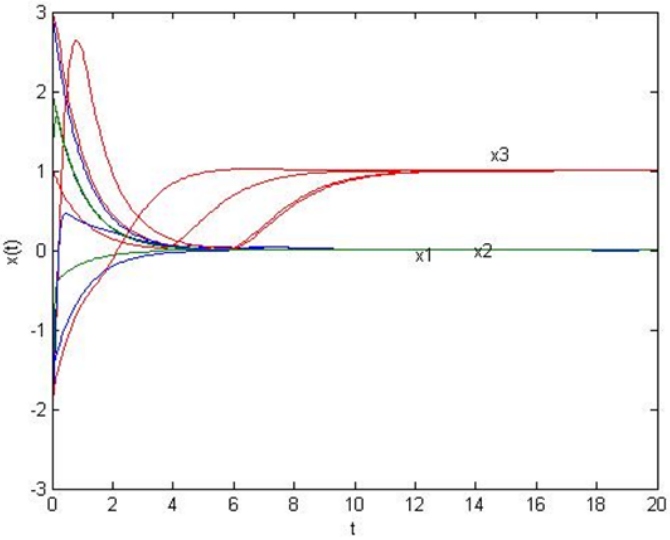

Fig. 9: Trajectories of the neural network (17) for four different initial points in Example 6.

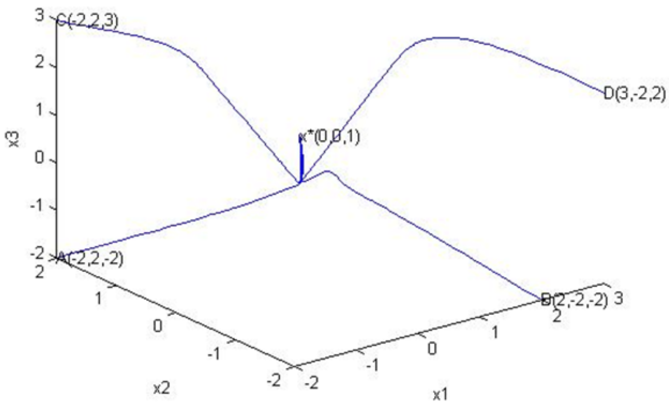

Fig. 10: Simulation results for the neural network (17) using four different initial points for Example 6.

Conclusions:

In this paper, we have presented a new recurrent neural network for solving nonlinear convex programming problem with nonlinear inequality and linear equality constraints and have demonstrated both analytically and by simulation results the stability of the solution of the problem. Also we extend the proposed neural network for solving a class of variational inequality problems with nonlinear equality and inequality constraints.

REFERENCES

Bazaraa, M.S., C.M. Shetty, 1990. Nonlinear Programming: Theory and Algorithms, John Wiley and Sons Inc., New York: Wiley.

Bertsekas, D.P., J.N. Tsitsiklis, 1989. Parallel and Distributed Computation: Numerical Methods, Prentice-Hall, Englewood Cliffs, NJ.

Cichocki, A., R. Unbehauen, 1993. Neural Networks for Optimization and Signal Processing, Wiley, New York.

Gao, X., L.Z. Liao, W. Xue, 2003. A neural network for monotone variational inequalities with linear constraints, Phys. Lett. A. 307: 118-128.

Golden, R.M., 1996. Mathematical Methods for Neural Network Analysis and design. Cambidge, MA: MIT Press.

Karpinskaya, N.N., 1967. Method of penalty functions and the foundations of Pyne's method, Automation and Remote Contr, 28(1): 124-129.

Kennedy, M.P., L.O. Chua, 1988. Neural networks for nonlinear programming, IEEE Trans. Circuits Syst, 35(5): 554-562.

Maa, C.Y., M.A. Shanblatt, 1992. Linear and quadratic programming neural network analysis, IEEE Trans. Neural Netw, 3(4): 580-594.

Malek, A., A. Yari, 2005. Primal-dual solution for the linear programming problems using neural Networks, Appl. Math. Comput., 167: 198-211.

Malek, A., H.G. Oskoei, 2005. Numerical solutions for constrained quadratic problems using high-performance neural networks, Appl. Math. Comput., 169: 451-471.

Malek, A., M. Alipour, 2007. Numerical solution for linear and quadratic programming problems using a recurrent neural network, Appl. Math. Comput., 192: 27-39.

Miller, R.K., A.N. Michel, 1982. Ordinary Differential Equations, Academic Press.

Oskoei, H.G., A. Malek, A. Ahmadi, 2007. Novel artificial neural network with simulation aspects for solving linear and quadratic programming problems, Appl. Math. Comput, 53: 1439-1454.

Pang, J.S., 1987. A posteriori error bounds for the linearly-constrained variational inequality problem, Math. Oper. Res., 12: 474-484.

Pyne, I.B., 1956. Linear programming on an electronic analogue computer, Trans. Am. Ins. Elect. Eng, 75: 139-143.

Rybashov, M.V., 1965. The gradient method of solving convex programming problems on electronic analog computers, Automation and Remote Contr, 26(11): 1886-1898.

Rybashov, M.V., 1965. Gradient method of solving linear and quadratic programming problems on electronic analog computers, Automation and Remote Contr, 26(12): 2079-2089.

Xia, Y.S, 1996. A new neural network for solving linear and quadratic programming problems, IEEE Trans. Neural Netw, 7(6): 1544-1547.

Yashtini, M., A. Malek, 2007. Solving complementarity and variational inequalities problems using neural networks, Appl. Math. Comput., 190(1): 216-230.

Yashtini, M., A. Malek, 2008. A discrete-time neural network for solving nonlinear convex problems witrh hybrid constraints, Appl. Math. Comput., 195: 576-584.

Zak, S.H., V. Upatising, S. Hui, 1995. Solving linear programming problems with neural networks: Acomparative study, IEEE Trans. Neural Netw, 6(1): 94-104.