ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN EFFICIENT ONE-LAYER RECURRENT NEURAL NETWORK FOR SOLVING A CLASS OF NONSMOOTH PSEUDOCONVEX OPTIMIZATION PROBLEMS

M. J. EBADI, M. M. HOSSEINI, S. M. KARBASSI
Department of Mathematics, Yazd University, P.O. Box 89195-741, Yazd, Iran
E-mail: ebadi@stu.yazd.ac.ir, hosse_m@yazd.ac.ir, smkarbassi@yazd.ac.ir

ABSTRACT

In this paper, an efficient one-layer recurrent neural network model which is differential inclusion-based is proposed for solving nonsmooth pseudoconvex optimization problems subject to linear equality constraints. The optimal solution of the original optimization problem is proven to be equivalent with the equilibrium point of the proposed neural network. In addition, the stability of the proposed neural network in the Lyapunov sense and globally convergence to an optimal solution are proven. Some illustrative examples are given to show the effectiveness of the proposed neural network. In addition, an application for condition number optimization is discussed.

Keywords: *Differential inclusion-based method, Nonsmooth optimization, Recurrent neural network, Lyapunov stability.*

1. INTRODUCTION

Consider the following constrained nonlinear optimization problem:

$$\min f(x)$$
$$\text{subject to } Ax=b, \qquad (1)$$

where $x=(x_1,x_2,...,x_n)^T \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. The objective function f is not necessary to be convex or smooth on the feasible region $\Omega=\{x \in \mathbb{R}^n \mid Ax=b\}$. Constrained optimization problems have many applications in science and engineering, such as robot control, signal processing, manufacturing system design, and pattern recognition [1] and [2]. Various types of neural network models have been proposed for solving optimization problems since 1960's. The first neural network model is proposed by Tank and Hopfield [3] which was introduced to solve linear programming problems. Other kinds of neural networks for solving linear and nonlinear optimization problems have been inspired by this neural network. For instance, a neural network for solving nonlinear programming problems presented in [4]. The structure of the proposed neural network model is based on Newton-type descent and finite penalty parameters used in this model. A significant problem arises for such neural network, when the penalty parameter is very large. Lagrangian networks are another type of neural networks which are constructed to solve nonlinear programming problems with equality constraints [5] and [6]. Projection-type neural network models belong to a class of neural network models which are designed to solve nonlinear convex programming problems (for example see [7]). Some projection neural networks which are proposed to solve general convex and nonlinear programming problems are globally convergent to an exact optimal solution [8] and [9]. Most existing neural network models are designed to solve convex optimization problems whereas, there are not many models to solve nonconvex ones, unless for problems with very limiting properties. In recent years, some efforts have been made to design models to solve nonsmooth nonconvex optimization problems, such as the model which is proposed by Liu and Wang [10]. This model proposed to solve a particular case of nonconvex problems which has a complicated structure and cannot be applied in engineering applications. Moreover, Guo et al. introduced some models which are applicable to solve nonsmooth nonconvex problems with pseudoconvex objective functions. The first model was designed to solve problems with linear equality constraints [11]. Then, they extended this model for solving problems containing linear equality and bound constraints [12]. The structure of all presented neural networks are based on a differential inclusion with some penalty parameters. Among nonconvex optimization problems, ones with pseudoconvex objective functions have many applications, such as computer vision [13], production planning, financial and corporate planning [14], fractional programming [15] and [16], healthcare and hospital planning, and frictionless contact analysis [17]. Solving nonsmooth optimization problems is difficult, even in unconstrained cases. In recent years many researchers proposed recurrent neural networks models for solving nonsmooth convex optimization problems (for example see [18], [19], [20], [21], [22], [23] and [24]); however, there are not many articles for solving nonsmooth pseudoconvex optimization problems. Most of existing models for solving nonsmooth pseudoconvex optimization problems are based on penalty parameters or penalty functions, such as proposed models in [10], [12], [22], and [36]. In fact, the effectiveness of these methods depends on the exact penalty parameter. Obviously, calculation of the penalty parameters is difficult in real utilizations. In spite of the lower complexity of the model in some cases, the speed of convergence is less than usual, for example in the case of model in [12]. In [31], in order to guarantee the convergence of the model for differentiable pseudoconvex optimization problems, the initial points must be chosen from inside the feasible region in advance. This may be computationally costly depending on the problem's structure. To overcome the above difficulties, we extend the model proposed in [11] and [31] to nonsmooth case which is not penalty-based, i.e. a one-layer recurrent neural network is presented for solving nonsmooth pseudoconvex optimization problems with linear equality constraints. The model is global, that is, there is no need to choose the initial point from inside the feasible region. We prove the global convergence of the proposed neural network and also its stability in the sense of Lyapunov for nonsmooth pseudoconvex optimization problems. The reminder of paper is organized as follow. Some definitions and relevant preliminaries are discussed in Section 2. In Section 3, we construct the neural network model. The stability and global convergence of the proposed neural network are analyzed and proved in Section 4. Section 5 presents some illustrative examples including quadratic fractional programming problem and condition number optimization problem to show the effectiveness and performance of the proposed neural network. Finally, in Section 6, some conclusions are presented.

2. PRELIMINARIES

In the section, some definitions and lemmas are presented for the convenience of later discussions. Note that l_1 and l_2 norms of a vector in $\mathbb{R}^n$ are presented by $\|\cdot\|_1$ and $\|\cdot\|_2$, respectively.

Definition 1. [25] Suppose that X and Y be two sets. A set valued map F from X to Y is a map that associates a subset $F(x)$ of Y to any point $x \in X$.

Definition 2. [25] A set valued map F with nonempty values is said to be upper semicontinuous (U.S.C.) at $x^0 \in X$ if for any open set N containing $F(x^0)$ there exists a neighborhood M of x^0, such that $F(M) \subseteq N$. Also, F is U.S.C. if and only if it is U.S.C. at every $x^0 \in X$.

Definition 3. [25] For a function $f : \mathbb{R}^n \to \mathbb{R}$, if there exists $\delta > 0$ for any given $\epsilon > 0$, such that for any $x_1, x_2 \in \mathbb{R}^n$, satisfying $\|x_1 - x\|_2 < \delta$ and $\|x_2 - x\|_2 < \delta$, we have $|f(x_1) - f(x_2)| \le \epsilon \|x_1 - x_2\|_2$, then f is said to be Lipschitz near $x \in \mathbb{R}^n$. The function f is said to be locally Lipschitz in $\mathbb{R}^n$, if f is Lipschitz near any point $x \in \mathbb{R}^n$.

Definition 4. [26] Suppose that f is Lipschitz near $x \in \mathbb{R}^n$. Then $f^0(x;v)$ is said to be the generalized directional derivative of f at x in the direction of any vector $v \in \mathbb{R}^n$ which is defined as $f^0(x;v) = \limsup_{y \to x, t \downarrow 0} \frac{f(y+tv) - f(y)}{t}$, and we define the Clarke's generalized gradient of f at x as $\partial f(x) = \{y \in \mathbb{R}^n : f^0(x;v) \ge y^T v, \forall v \in \mathbb{R}^n\}$.

Definition 5. [26] A function f is said to be regular at x if for all $v \in \mathbb{R}^n$:

(1) there exists the usual one-sided directional derivative $f'(x;v)$ which is given by

$$f'(x;v) = \lim_{\xi \to 0^+} \frac{f(y+\xi v) - f(y)}{\xi};$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

(2) $f'(x;v) = f^0(x;v)$. Clearly, any convex function is regular.

Lemma 6. (Chain Rule Clarke [26]) If $W: \mathbb{R}^n \to \mathbb{R}$ is regular at $x(t)$ and $x(t): \mathbb{R} \to \mathbb{R}^n$ is differentiable at t, and Lipschitz near t, , then

$$\frac{dW(x(t))}{dt} = \xi^T \dot{x}, \forall \xi \in \partial W(x(t))$$
$$for\ a.e.\ t \in [0, \infty),$$

Definition 7. [27] Let a set $\Omega \subset \mathbb{R}^n$ is nonempty closed convex set. A function $f: \Omega \to \mathbb{R}$ is said to be pseudoconvex on Ω if, for every pair of distinct points $x_1, x_2 \in \Omega$

$$\exists \xi(x_1) \in \partial f(x_1): \xi(x_1)^T (x_2 - x_1) \ge 0$$
$$\Rightarrow f(x_2) \ge f(x_1).$$

Definition 8. [27] Let $\Omega \subset \mathbb{R}^n$ is a nonempty closed convex set. A set valued map $F: \Omega \to \mathbb{R}^m$ is said to be pseudomonotone on Ω if, for every pair of distinct points $x, y \in \Omega$,

$$\exists \xi_x \in F(x): \xi_x^T (y - x) \ge 0$$
$$\Rightarrow \forall \xi_y \in F(y): \xi_y^T (y - x) \ge 0.$$

Assumption 9. (I) $f: R^n \to R$ is pseudoconvex and regular on the feasible region $\Omega = \{x \in R^n \mid Ax = b\}$.

(II) $A \in R^{m \times n}$ is a full row-rank matrix, i.e. $rank(A) = m \le n$.

(III) There exists at least one optimal solution of problem (1) .

Here we introduce Karush-Kuhn-Tucker (KKT) conditions for nonsmooth nonlinear optimization problem (1) as follows:

Lemma 10. Let Assumption 9. holds, then x^* is an optimal solution of problem (1) if there exists $v^* \in \mathbb{R}^m$ such that (x^*, v^*) satisfies the following equations

$$\begin{cases} 0 \in \partial f(x^*) + A^T v^* \\ Ax^* - b = 0, \\ v^* \text{ free in sign} \end{cases} \quad (2)$$

3. NEURAL NETWORK MODEL

According to Lemma 10. for solving nonlinear programming problem (1), we propose a one-layer recurrent neural network model which is an extension of the model proposed in [11] to nonsmooth pseudoconvex optimization problems, with the following differential inclusion:

$$\frac{dx}{dt} \in -(I - P)\partial f(x) - A^T h(Ax - b), \quad (3)$$

where $P = A^T (AA^T)^{-1} A$, $h = (h(x_1), h(x_2), \ldots, h(x_m))^T$ and $h(x_i)$ is defined as follows

$$h(x_i) = \begin{cases} 1 & \text{if } x_i > 0, \\ [-1,1] & \text{if } x_i = 0, \\ -1 & \text{if } x_i < 0. \end{cases} \quad (4)$$

For $i \in \{1, 2, \cdots, m\}$. Also, $\partial f(x)$ is the Clarke subdifferential of f at x. The neural network (3) can be realized by a generalized circuit. To find more details of the generalized circuit, readers can refer to [24], [28], [29] and [30]. Now, a generalized circuit implementation of neural network (3) is proposed for a simple optimization problem as follows:

$$\min f(x_1, x_2) = |x_1 + x_2| + |x_1|,$$
$$\text{subject to } Ax = b, b \in \mathbb{R}^2. \quad (P)$$

We can simulate the implementation of

Journal of Theoretical and Applied Information Technology
15th April 2018. Vol.96. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

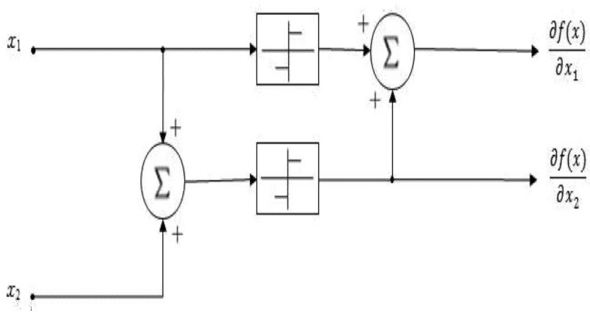

Figure 1: Block diagram of $\partial f(x)$ by circuits

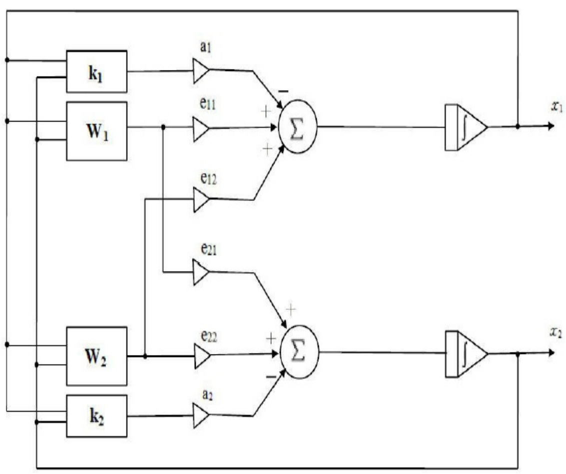

Figure 2: Schematic block diagram of neural

$\partial f(x)$ as the block diagram in Fig. 1, also the architecture of neural network (3) for the optimization problem (P) is shown in Fig. 2, where $(k_1,k_2)^T = h(Ax-b), [e_{ij}]_{2\times 2} = I-P$, and $(W_1,W_2)^T = (\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2})^T$.

Definition 11. x^* is said to be an equilibrium point of neural network (3) if $0 \in (I-P)\xi^* + A^T h(Ax^*-b)$, i.e. there exist $\xi^* \in \partial f(x^*)$ and $\gamma^* \in h(Ax^*-b)$ such that $(I-P)\xi^* + A^T\gamma^* = 0$.

Definition 12. An absolutely continuous function $x(\cdot):[0,T] \to \mathbb{R}^n$ is said to be a solution of neural network (3) on an interval [0, T], which satisfies the initial condition $x(0) = x_0$, and for almost all $t \in [0,T)$:

$\frac{dx(t)}{dt} \in -(I-P)\partial f(x(t)) - A^T h(Ax-b)$. In other words, there exist measurable functions $\xi(t) \in \partial f(x(t))$ and $\gamma(t) \in h(Ax-b)$, such that $\dot{x}(t) = -(I-P)\xi(t) - A^T\gamma(t)$, for almost all $t \in [0,T)$.

Lemma 13. For any $x \in \mathbb{R}^n$, $Ax = b$ if and only if $Px = A^T(AA^T)^{-1}b$, where $P = A^T(AA^T)^{-1}A$.

Proof. It can be easily proved.

4. CONVERGENCE ANALYSIS

In this section, we analyse and prove the global convergence of neural network (3). Firstly, we prove that the state of neural network (3) will be convergent to $\Omega = \{x \in \mathbb{R}^n \mid Ax = b\}$, when the initial point is chosen from outside of the feasible region Ω. Then, the stability of neural network (3) in the sense of Lyapunov, and globally convergence to an optimal solution to problem (1) are proved.

Theorem 14. Let Assumption 9. holds. For neural network (3), there exists at least a local solution $x(t)$ for any initial point $x(0) = x_0 \in \mathbb{R}^n$. $x(t)$ is convergent to the feasible region $\Omega = \{x \in \mathbb{R}^n \mid Ax = b\}$ in finite time by $t_u = \frac{\|Ax(0)-b\|_1}{\lambda_{\max}(AA^T)}$ when the initial point $x(0) = x_0 \notin \Omega$ and will remain there thereafter, where $\lambda_{\max}$ is the maximum eigenvalue of the matrix AA^T Also, $x(t) \in \Omega$ for all $t \geq 0$, when $x_0 \in \Omega$.

Proof. Since the right hand side of (3) is U.S.C. and its values are nonempty convex compact, then for any initial point $x_0 \in \mathbb{R}^n$, according to Theorem 3 on page 98 of [25], there exist at least a local solution x(t) of neural network (3) with $x(0) = x_0 \in \mathbb{R}^n$ and a positive T for $t \in [0,T)$.
Which means that we have

$\dot{x}(t) = -(I-P)\eta(t) - A^T\xi(t), for\ a.e.\ t \in [0,T)$

where $\eta(t) \in \partial f(x(t))$ and $\xi(t) \in h(Ax(t)-b)$ are measurable functions. Note that only one of the following cases is satisfied

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$\begin{cases} T < +\infty \, and \lim_{t \to T^-} \|x(t)\| = +\infty \\ T = +\infty \end{cases} \quad (5)$$

Similar to the proof of Theorem 1 in [10] and [11], suppose that $B(x) = \|Ax - b\|_1$. Clearly, $B(x)$ is regular and convex. By using the chain rule, for a.e. $t \in [0, T)$, we have

$$\frac{d}{dt} B(x(t)) = \xi^T A\dot{x}(t), \quad \forall \xi \in h(Ax(t) - b). \quad (6)$$

Substituting ξ by $\zeta(t) \in h(Ax(t) - b)$ implies that

$$\begin{aligned} &\frac{d}{dt} B(x(t)) = \zeta(t)^T A\dot{x}(t) \\ &= -\zeta(t)^T A(I - P)[\eta(t)] - \zeta(t)^T AA^T \zeta(t) \\ &= -\zeta(t)^T AA^T \zeta(t) \\ &\le -\lambda_{\max}(AA^T)\|\zeta(t)\|^2, \end{aligned} \quad (7)$$

where $\lambda_{\max}(AA^T)$ is maximum eigenvalue of AA^T. In (7), we use the fact that $A(I - P) = A(I - A^T(AA^T)^{-1}A) = A - A = 0$ and $\lambda_{\max}(AA^T)$ is bigger than zero since A is a full row-rank matrix. Suppose that $x(t) \notin \Omega$, $\forall t \in [0, T)$ i.e. $Ax(t) \ne b$. Now

Case I: $T = +\infty$, so according to the definition of h in (4), $\|\zeta(t)\|^2 \ge 1$ which combining with (7) leads to

$$\frac{d}{dt} B(x(t)) \le -\lambda_{\max}(AA^T) < 0. \quad (8)$$

Hence, by integrating from both sides of (8) from 0 to t, we get that

$$B(x(t)) - B(x(0)) \le -\lambda_{\max}(AA^T)t \quad (9)$$

which means that

$$\begin{aligned} 0 &\le \|Ax(t) - b\|_1 \\ &\le \|Ax(0) - b\|_1 - \lambda_{\max}(AA^T)t. \end{aligned}$$

So, when $t = \dfrac{\|Ax(0) - b\|_1}{\lambda_{\max}(AA^T)}$ we have $Ax(t) - b = 0$ which is clearly a contradiction.

Case II: $T < +\infty$ and $\lim_{t \to T^-} \|x(t)\| = +\infty$.

According to the proof of case I. we have

$$\begin{aligned} &\|Ax(t) - b\|_1 \\ &\le \|Ax(0) - b\|_1 - \lambda_{\max}(AA^T)t \\ &\le \|Ax(0) - b\|_1 . \end{aligned} \quad (10)$$

Therefore,

$$\begin{aligned} &\lambda_{max}(AA^T)\|x(t)\|_1^2 - 2\|b^T A\|_1 \|x(t)\|_1 + b^T b \\ &\le \|Ax(t) - b\|_1^2 \le \|Ax(0) - b\|_1^2 < +\infty, \end{aligned} \quad (11)$$

By taking limit on both sides of (11) when $t \to T^-$ we have

$$+\infty = \lim_{t \to T^-} [\lambda_{max}(AA^T)\|x(t)\|_1^2 - 2\|b^T A\|_1 \|x(t)\| + b^T b] < +\infty,$$

which is clearly a contradiction. Then for any $x(0) \notin \Omega$ the state $x(t)$ will reach $\Omega = \{x \in \mathbb{R}^n \mid Ax - b = 0\}$ in finite time and $t_u = \dfrac{\|Ax(0) - b\|_1}{\lambda_{\max}(AA^T)}$ is an upper bound for the hit time. Now we prove that the state $x(t)$ will be stayed inside the feasible region Ω, if $x(t)$ leaves Ω at $t_1 > t_u$, there must exists $t_2 > t_1$ such that $x(t) \notin \Omega$ for all $t \in (t_1, t_2)$ and $\|Ax(t_1) - b\|_1 = 0$. Therefore, according to (7), we have

$$\begin{aligned} &\|Ax(t_2) - b\|_1 \\ &\le \|Ax(t_1) - b\|_1 - \lambda_{\max}(AA^T)(t_2 - t_1) \\ &= -\lambda_{\max}(AA^T)(t_2 - t_1) < 0, \end{aligned} \quad (12)$$

which is clearly contradiction. Hence, $x(t)$ will reach Ω in finite time and will remain there thereafter. If $x_0 \in \Omega$, then $B(x_0) = 0$. So, $B(x(t)) = 0$ for $t \ge 0$ by (9). Therefore, when $x_0 \in \Omega$, $x(t) \in \Omega$ for $t \ge 0$.

Theorem 15. If x^* is an equilibrium point of differential inclusion (3), then x^* is an optimal solution of the optimization problem (1) and vice versa.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Proof.

Let x^* be an equilibrium point of differential inclusion (3) then we have

$$0 \in (I-P)\left(\partial f(x^*)\right) + A^T h(Ax^*-b),$$

according to Lemma 10, we have $Ax^* = b$. By choosing $v^* = (AA^T)^{-1}A(\partial f(x^*)) + h(Ax^*-b)$, we have

$$0 \in (I-P)\left(\partial f(x^*)\right) + A^T h(Ax^*-b)$$
$$= \partial f(x^*) + A^T v^*. \quad (13)$$

Therefore, according to Lemma 10, x^* is an optimal solution of problem (1). To prove the rever se side suppose that x^* is an optimal solution of the optimization problem (1). Therefore, according to Lemma 10, there exist $v^* \in \mathbb{R}^m$ such that the equalities in (1) hold. Therefore, $0 \in (I-P)\left(\partial f(x^*)\right) + A^T v^* = (I-P)[\partial f(x^*)]$.

Moreover, since $0 \in h(Ax^*-b)$, we have

$$0 \in (I-P)\left(\partial f(x^*)\right) + A^T h(Ax^*-b).$$

By combining with the equality in (2), clearly x^* is an equilibrium point of differential inclusion (3).

Theorem 16. Let Assumption 9. holds. For any initial point $x_0 \in \mathbb{R}^n$, the solution $x(t)$ of neural network (3) is stable in the sense of Lyapunov and convergent to an optimal solution of neural network (3).

Proof. Consider $\bar{x}$ as an equilibrium point of neural network (3). With regards to Theorem 15, $\bar{x}$ is also an optimal solution of problem (1). Thus, $\bar{x} \in \Omega = \{x \in \mathbb{R}^n \mid Ax - b = 0\}$. Since $\bar{x}$ is an equilibrium point of neural network (3), there exist measurable functions $\bar{\eta} \in \partial f(\bar{x})$ and $\bar{\gamma} \in h(Ax-b)$ such that

$$A^T \bar{\gamma} + (I-P)\bar{\eta} = 0. \quad (14)$$

Since, $\bar{x} \in \Omega = \{x \in R^n \mid Ax - b = 0\}$ without loss of generality we can choose $h(A\bar{x}-b) = 0$, then (14) leads to

$$(I-P)\bar{\eta} = 0. \quad (15)$$

Substituting (15) into (3) leads

$$\frac{dx}{dt} \in -A^T h(Ax-b) - (I-P)(\partial f(x) - \bar{\eta}). \quad (16)$$

According to Theorem 14. we have the convergence of any $x(t)$ to the feasible region Ω in finite time and remaining there thereafter. So it is sufficient to show the stability of the system with $x(t) \in \Omega$. Since $x \in \Omega$ and $x \neq \bar{x}$, according to Lemma 13. by multiplying $(I-P)$ on both sides of (16), we have $Px = A^T(AA^T)^{-1}b$. Thus, $P\dot{x} = 0$. Then, we have $\dot{x} = (I-P)\dot{x}$ since we can write $x = Px + (I-P)x$. Combining with $(I-P)A^T = 0$ and this fact that $(I-P)^2 = (I-P)$, we

Have $\frac{dx}{dt} \in -(I-P)(\partial f(x) - \bar{\eta})$, i.e. there exists a measurable function $\eta \in \partial f(x)$ such that

$$\frac{dx}{dt} = -(I-P)(\eta - \bar{\eta}), \quad (17)$$

Consider the following Lyapunov function:

$$V(x) = \exp(f(x) - f(\bar{x}) - (x-\bar{x})^T \bar{\eta}$$
$$+ \frac{1}{2}\|Px - \bar{x}\|_2^2) - 1. \quad (18)$$

We have $V(\bar{x}) = 0$ and for $x \neq \bar{x}$, $V(x) > 0$ and $\partial V(x) = [\partial f(x) - \bar{\eta} + x - \bar{x}]\exp(f(x) - f(\bar{x})$
$-(x-\bar{x})^T \bar{\eta} + \frac{1}{2}\|Px - \bar{x}\|_2^2)$.

According to the Chain rule, $V(x(t))$ is differentiable for almost all $t \geq 0$. Hence,

$\frac{d}{dt}V(x(t)) = \zeta(t)^T \dot{x}(t), \ \forall \zeta(t) \in \partial V(x(t))$. From (3), $\eta(t) \in \partial f(x(t))$, hence by choosing

$$\zeta(t) = (\eta(t) - \bar{\eta} + x(t) - \bar{x})\exp(f(x) - f(\bar{x}) - (x-\bar{x})^T \bar{\eta}$$
$$+ \frac{1}{2}\|Px - \bar{x}\|_2^2) \in \partial V(x(t))$$

and according to (17) we have

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$\frac{d}{dt}V(x(t))$$
$$\leq \sup_{\eta\in\partial f(x)} [\eta-\bar{\eta}+x-\bar{x}]^T[-(I-P)(\eta-\bar{\eta})]$$
$$\exp(f(x)-f(\bar{x})-(x-\bar{x})^T\bar{\eta}+\frac{1}{2}\|x-\bar{x}\|_2^2)$$
$$=\sup_{\eta\in\partial f(x)} [-(\eta-\bar{\eta})^T(I-P)(\eta-\bar{\eta})$$
$$-(x-\bar{x})^T(\eta-\bar{\eta})+(x-\bar{x})^TP(\eta-\bar{\eta})]$$
$$\exp(f(x)-f(\bar{x})-(x-\bar{x})^T\bar{\eta}$$
$$+\frac{1}{2}\|x-\bar{x}\|_2^2). \qquad (19)$$

For any $x\in\Omega$, we have

$$(x-\bar{x})^TP$$
$$=[(x-\bar{x})^TA^T](AA^T)^{-1}A$$
$$=[(Ax-A\bar{x})^T](AA^T)^{-1}A$$
$$=0. \qquad (20)$$

Hence,

$$(x-\bar{x})^T\bar{\eta}$$
$$=[(x-\bar{x})^TP+(x-\bar{x})^T(I-P)]\bar{\eta}$$
$$=(x-\bar{x})^T(I-P)\bar{\eta}$$
$$=0. \qquad (21)$$

From pseudoconvexity of $f(x)$ on the feasible region Ω, $\partial f(x)$ is pseudomonotone on Ω. Thus, according to the Definition 7. we have

$$(x-\bar{x})^T\eta\geq 0, \qquad (22)$$

for any $\eta\in\partial f(x)$. So, from (21) and (22) we find that

$$(x-\bar{x})^T(\eta-\bar{\eta})\geq 0. \qquad (23)$$

Consequently, combining (20) and (23) with (19) leads to

$$\frac{d}{dt}V(x(t))$$
$$\leq \sup_{\eta\in\partial f(x)} [-(\eta-\bar{\eta})^T(I-P)(\eta-\bar{\eta})]\exp(f(x)$$
$$-f(\bar{x})-(x-\bar{x})^T\bar{\eta}+\frac{1}{2}\|x-\bar{x}\|_2^2)$$
$$=\sup_{\eta\in\partial f(x)} [-\|(I-P)(\eta-\bar{\eta})\|_2^2]\exp(f(x)$$
$$-f(\bar{x})-(x-\bar{x})^T\bar{\eta}+\frac{1}{2}\|x-\bar{x}\|_2^2) \qquad (24)$$

By substituting (15) into (24) we have

$$\frac{d}{dt}V(x(t))$$
$$\leq \sup_{\eta\in\partial f(x)} [-\|(I-P)\eta\|_2^2]\exp(f(x)$$
$$-f(\bar{x})-(x-\bar{x})^T\bar{\eta}+\frac{1}{2}\|x-\bar{x}\|_2^2)$$
$$=-\inf_{\eta\in\partial f(x)} [\|(I-P)\eta\|_2^2]\exp(f(x)$$
$$-f(\bar{x})-(x-\bar{x})^T\bar{\eta}+\frac{1}{2}\|x-\bar{x}\|_2^2)$$
$$\leq 0. \qquad (25)$$

By the last inequality, the global stability of the neural network (3), in the sense of Lyapunov is proved. Similar to the proof of Theorem 3. in [19] and Theorem 4. in [18] we take $\Gamma(x)=\inf_{\eta\in\partial f(x(t))}\|(I-P)\eta\|_2^2\exp(f(x)-f(\bar{x})$
$-(x-\bar{x})^T\bar{\eta}+\frac{1}{2}\|x-\bar{x}\|_2^2)$

Then it is easy to verify that $\Gamma(x)=0$ and $x\in\Omega$ if and only if x is an equilibrium point of neural network (3). Moreover, according to the construction of $V(x)$, $V(x)$ is radially unbounded since $V(x)\geq\frac{1}{2}\|x(t)-x^*\|_2^2$, also, by taking integral on both sides of $\frac{dV}{dt}\leq 0$, we have the following inequality

$$0\leq V(x(t))\leq V(x(0))<\infty, \qquad (26)$$

clearly, we can conclude that $x(t)$ is bounded. We have the boundedness of $\|\dot{x}(t)\|_2^2$ according to (3), and we choose a sufficiently large constant M as an upper bound. However, a convergent subsequence exists as follow

$$\{x(t_k) \mid 0 \le t_1 \le t_2 \le \ldots\}, \quad and \quad \lim_{k\to+\infty} t_k = +\infty, \qquad (27)$$

such that $\lim_{k\to+\infty} x(t_k) = \bar{x}$. Clearly, $\bar{x} \in \Omega$. To show that $\bar{x}$ is an equilibrium point of neural network (3), we prove $\Gamma(\bar{x}) = 0$. Suppose that $\Gamma(\bar{x}) \ne 0$, that is $\Gamma(\bar{x}) > 0$. By Theorem 5 on page 52 in [25], we have the lower semicontinuity of $\Gamma(x)$ i.e. there exists $\delta > 0$ for any $\varepsilon > 0$ such that $\Gamma(x) > \varepsilon$ for all $x \in B(\bar{x}, \delta)$, where $B(\bar{x},\delta) = \{x \in R^n \, \|x - \bar{x}\|_2 < \delta\}$ is the δ neighborhood of $\bar{x}$ as follows

$$\Gamma(x) > \varepsilon, \quad \forall x \in B(\bar{x}, \delta) = \{x \in R^n \, \|x - \bar{x}\|_2 < \delta\}. \qquad (28)$$

Since $\lim_{k\to+\infty} x(t_k) = \bar{x}$, a positive integer N exists, such that $\|x(t_k) - \bar{x}\|_2 \le \delta/2$ for all $k \ge N$. When $t \in [t_k - \frac{\delta}{4M}, t_k + \frac{\delta}{4M}]$ and $k \ge N$, we have

$$\begin{aligned} &\|x(t) - \bar{x}\|_2 \\ &\le \|x(t) - x(t_k)\|_2 + \|x(t_k) - \bar{x}\|_2 \\ &\le M\left|t - t_k\right| + \frac{\delta}{2} \le \delta, \end{aligned} \qquad (29)$$

combining (29) with (28) leads to $\Gamma(x(t)) > \varepsilon$ for all $t \in [t_k - \frac{\delta}{4M}, t_k + \frac{\delta}{4M}]$ and $k \ge N$. Therefore, by choosing $\Lambda = \cup_{k \ge N} [t_k - \frac{\delta}{4M}, t_k + \frac{\delta}{4M}]$ and this fact that the Lebesgue measure of the set Λ is infinite we have

$$\begin{aligned} &\int_0^\infty \Gamma(x(t))dt \\ &\ge \int_\Lambda \Gamma(x(t))dt \\ &= +\infty \end{aligned} \qquad (30)$$

On the other hand, according to (25) and (26), $V(x(t))$ is monotonically decreasing and bounded on Ω. Then, a constant V_0 exists such that $\lim_{t\to+\infty} V(x(t)) = V_0$. Therefore,

$$\begin{aligned} &\int_0^\infty \Gamma(x(t))dt = \lim_{s\to+\infty} \int_0^s \Gamma(x(t))dt \\ &\le -\lim_{s\to+\infty} \int_0^s \dot{V}(x(t))dt \\ &= -\lim_{s\to+\infty} [V(x(t)) - V(x(0))] \\ &= -V_0 + V(x(0)) < +\infty, \end{aligned} \qquad (31)$$

which contradicts (30). Therefore, $\Gamma(\bar{x}) = 0$, which means that $\bar{x}$ is an equilibrium point of neural network (3) and by Theorem 15. we have $\bar{x}$ as the optimality solution problem (1). At the end, we want to prove the convergence of the state vector of neural network (3) to an equilibrium point i.e. $\lim_{t\to+\infty} x(t) = \tilde{x}$. We define another Lyapunov function

$$\begin{aligned} W(x) &= \exp(f(x) - f(\tilde{x}) - (x - \tilde{x})^T \tilde{\eta} \\ &+ \frac{1}{2}\|x - \tilde{x}\|_2^2) - 1. \end{aligned} \qquad (32)$$

Obviously, $W(\tilde{x}) = 0$. Similar to the previous proof, we can prove that

$$\begin{aligned} &\frac{dW(x(t))}{dt} \le 0 \quad and \\ &W(x(t)) \ge \frac{1}{2}\|x(t) - \tilde{x}\|_2^2. \end{aligned} \qquad (33)$$

Since the function $W(x)$ is continuous, there exists $\delta > 0$ such that for any $\varepsilon > 0$ we have

$$W(x) = |W(x) - W(\tilde{x})| < \varepsilon \qquad (34)$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

for any $x \in \mathbb{R}^n$, when $\|x - \tilde{x}\|_2^2 < \delta$. Moreover, since $\lim_{t_k \to +\infty} x(t_k) = \tilde{x}$ there exists t_N such that for all $t > t_N$, we have

$$\|x(t_N) - \tilde{x}\|_2^2 < \delta. \qquad (35)$$

Clearly, by (33), (34) and (35) we conclude that

$$\|x(t) - \tilde{x}\|_2^2 \le 2W(x(t)) \le 2W(x(t_N)) < 2\varepsilon. \qquad (36)$$

Which means that $\lim_{t \to +\infty} x(t) = \tilde{x}$. This completes the proof.

Remark 1. In comparison with the proposed neural network in [31], the neural network (3) in this paper has some benefits. For instance, the initial point can be chosen from inside or outside the feasible region, while in [31] the initial point must be chosen from inside the feasible region. Also, the objective function may be nonsmooth in this paper, which extends the domain of the pseudoconvex optimization problem (1) in [31]. In addition, the Lyapunov stability and the global convergence to an optimal solution of the neural network (3) in this paper are proved, by Theorem 16. with any initial point $x_0 \in \mathbb{R}^n$, while these properties are proved in [31] just when the initial point is chosen from inside the feasible region.

5. NUMEAICAL EXAMPLES

In this section, two examples are presented to illustrate the effectiveness of the proposed neural network (3) for solving the nonsmooth pseudoconvex optimization problem (1). We know that pseudoconvexity is a weaker condition than convexity. If we use neural network (3) for solving pseudoconvex optimization problem, we can solve the convex problem as a special case; Therefore, Example 1. is a convex optimization problem solved by neural network (3). In all examples, the differential inclusion defined by (3) is solved using MATLAB R2015b, on a 2.4 GHZ Intel Core(TM) i5 Qrad PC running Windows 7 Ultimate with 4.00 GB main memory.

Example 1. Consider the following nonsmooth convex optimization problem with one linear equality constraint.

$$\begin{aligned} &\min f(x) = |x_1| + x_2^2 + 2 \\ &\text{subject to } x_1 + x_2 - 1 = 0 \end{aligned} \qquad (37)$$

This problem was discussed and solved in [32]. The only optimal solution of this problem is $x^* = (0.5, 0.5)$. We solve this problem by using the proposed neural network (3). The subgradient of $f(x)$ at point x is given by

$$\partial f(x) = \begin{cases} (1, 2x_2)^T, & x_1 > 0 \\ ([-1, 1], 2x_2)^T, & x_1 = 0 \\ (-1, 2x_2)^T, & x_1 < 0 \end{cases} \qquad (38)$$

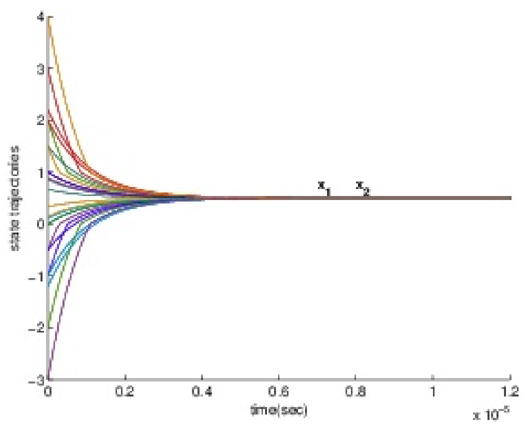

Figure 3: Transient behaviors of the neural network (3) with 10 random initial points in Example 1.

Fig.3 shows that the trajectories of neural network (3) with 10 random initial points, will converge to the optimal solution $x^* = (0.5, 0.5)$. Comparing to [32], we do not use any penalty parameter in the structure of the model, but in [32] to solve the problem we need to choose penalty parameters which may lead to wrong solution or degeneracy.

5.1. NONSMOOTH NONCONVEX OPTIMIZATION PROBLEMS

Example 2. Consider the following pseudoconvex optimization problem with linear equality

Journal of Theoretical and Applied Information Technology
15th April 2018. Vol.96. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

constraints:

$$\min \ f(x) = (x_1 - x_2)^2 + (x_2 - x_3)^2 + (x_3 - x_4)^2 + (x_4 - x_5)^2,$$

$$subject\ to\ Ax = b,$$

$$where\ A = \begin{bmatrix} 1 & 2 & 3 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 \\ 0 & 0 & 1 & 2 & 3 \end{bmatrix}, b = \begin{bmatrix} 6 \\ 6 \\ 6 \end{bmatrix}.$$

This example was solved in [31], but in that paper the initial points must be chosen from inside the feasible region Ω, while by the proposed neural network (3) no needs the initial points to be chosen from inside the feasible region. The only optimal solution of this problem is $x^* = (1, 1, 1, 1, 1)$. We solve this problem by using the proposed neural network (3). Fig. 4 shows that the trajectories of the neural network (3) with 10 random initial points will converge to the optimal solution x^*.

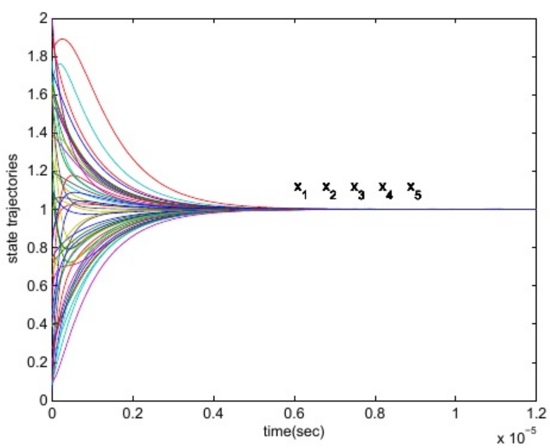

Figure 4: Transient behavior of the neural network (3) with 10 random initial points in Example 2.

Example 3. Consider the following nonsmooth nonconvex optimization problem

$$\min f(x) = \frac{x_1^2 + x_3}{x_2} + |2 + x_2| + |x_1 + x_2 + 1|$$

$$subject\ to\ \sum_{i=1}^{3} x_i = 1,\ x_2 \geq \frac{1}{5}.$$

The objective function is nonsmooth and pseudoconvex on the feasible region and the neural network (3) is capable of solving the problem. The optimal solution of this problem is x^*= (0.1727, 0.6547, 0.1727)^T and the optimal value of the objective function is 4.7913. Fig. 5 shows that all trajectories with 10 random initial points are convergent to the optimal solution x^* in finite time.

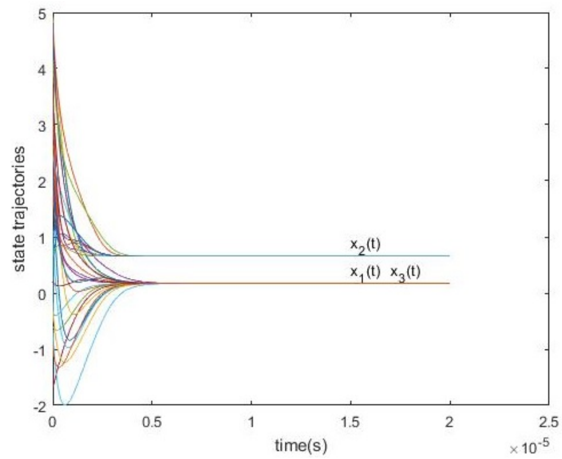

Figure 5: Transient behavior of the neural network (3) with 10 random initial points for solving Example 3.

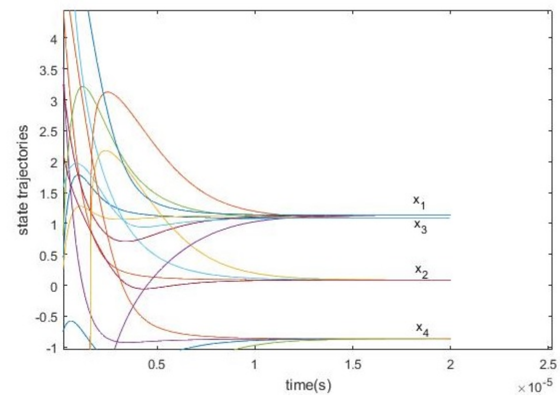

Figure 6: Transient behavior of the neural network (3) to solve Example 4, with 4 random initial points of the Table A-1. State trajectories converge to (1.0865, 0.0865, 1.1327, -0.8673)^T where state trajectory coordinate is limited to [-1, 4].

Example 4. Consider the following quadratic fractional programming problem

$$\min \ f(x) = \frac{x^T Qx + a^T x + a_0}{c^T x + c_0}$$

$$s.t. \quad Ax = b$$

With

Journal of Theoretical and Applied Information Technology
15th April 2018. Vol.96. No 7
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

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

$$c=\begin{pmatrix} 2 & 1 & 1 & 0 \end{pmatrix}^T,\ c_0=4$$

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

It is clear to show that the objective function f is pseudoconvex on $\{x \in \Re^4 : c^T x + c_0 > 0\}$, as Q is symmetric and positive definite. We use three methods for solving the problem. Our new proposed model in this paper, a penalty-based method [12] and the interior-point method [33] ("fmincon" Matlab code) are used for solving the problem. We choose four random initial points from outside of the feasible region. Two penalty values (σ = 0.5, 52) are used for the penalty-based method. Note that finding exact penalty values for the model in [12] is not easy. The optimal solution and optimal value of the problem are x^*= (1.0865,

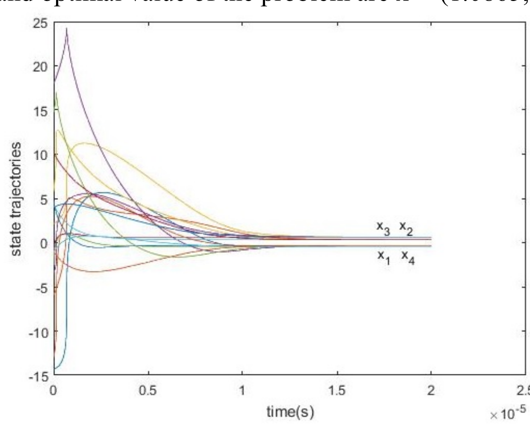

Figure 7: Transient behavior of a penalty-based model [12] with the penalty value σ = 0.5 to solve Example 4, with 4 random initial points of the Table A-1. State trajectories converge to a non-optimal and infeasible solution.

0.0865, 1.1327, -0.8673)^T and 1.8153 respectively. The state behavior of the presented method is shown in Fig. 6. The quality of convergence of aforementioned methods is shown in Table A-1 in Appendix A. Comparing to other two methods our proposed method has the following benefits:

1-Non-penalty based structure: As it can be clearly observed from the Table. A-1, the solution accuracy and convergence of the penalty based method is completely dependent on choosing the appropriate penalty values. In the problem, with the penalty value σ = 0.5 all trajectory solutions are convergent to (-0.4275, 0.3277, 0.6257, -0.3172)^T which is not an optimal or equilibrium point of the problem. It can be seen that all corresponding solutions are infeasible and all state trajectories converge to a wrong solution (see Fig. 7) whereas the trajectory solutions of the problem with penalty value σ = 52 are convergent to the optimal solution. The merit of our new proposed model is that it does not need to use any penalty parameter or penalty function in its structure prior to solving the problem.

2- Running time: In the penalty methods, the convergence of trajectories to the optimal solution depends on the penalty parameter value or the value of the penalty function. However, our new proposed model is not depend on such values and the method is very simple. At first, the trajectories will converge to the feasible region and then they will converge to the optimal solution. From Table. A-1, it is inferred that our method has better performance in running time in comparison with penalty based method and interior point method.

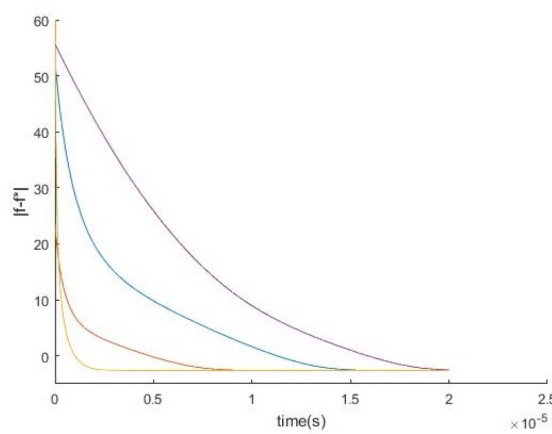

Figure 8: Error function values for a penalty-based model [12] with the penalty value σ = 0.5 to solve Example 4, with 4 random initial points of the Table A-1.

Although the penalty based method is convergent to optimal solution for penalty value σ = 52 but running time is considerably high and the use of this method is not cost-effective.

3- Accuracy: From Table. A-1, it can be clearly seen that the interior point method is divergent for all random initial points in the Table and the penalty based method with penalty value σ = 0.5

converges to an infeasible point which is not optimal (see the corresponding error |f-f*| in Fig. 8). While the penalty method with σ =52 converges to the optimal and feasible solution, the CPU run time is much higher than our proposed model. It is obvious from Table. A-1, that the convergence of our proposed method is faster than the convergence of other methods. The corresponding error |f-f*| of our method is shown in Fig. 9. Also, Fig. 10 shows the objective value f(x(t)) along the solution of neural network (3) with four different random initial points of the Table A-1, which are convergent to the optimal value of the objective function f (x*) =1.8153. From the above explanations, it can be inferred that choosing inappropriate penalty parameter may lead to wrong solution or degeneracy. As it is mentioned above, our new model shows better performance and effectiveness in comparison with some other NN models and a traditional method (interior point method).

5.2. MINIMIZING CONDITION NUMBER

In this section, the proposed neural network is applied to minimize the condition number. Consider the following optimization problem

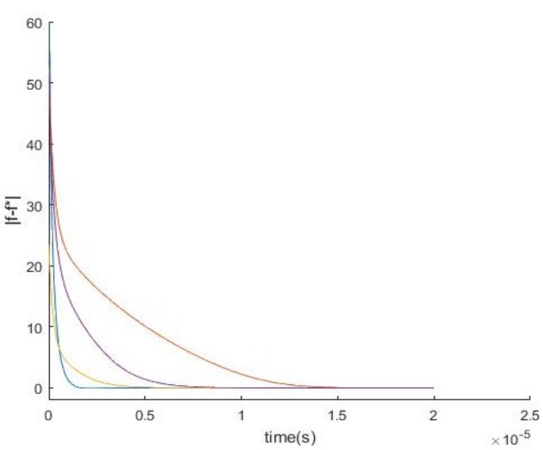

Figure 9: Error function values for neural network (3) with the penalty value σ = 0.5 to solve Example 4, with four random initial points of the Table A-1.

$$\begin{aligned} &\min \quad \kappa(S) \\ &\textit{subject to } S \in \Lambda, \qquad (39) \end{aligned}$$

where Λ is a compact convex subset of S_n^+, the cone of symmetric positive semidefinite $n \times n$ matrices, and Λ indicated to condition number of S. If the eigenvalues of S denote, in decreasing order by $\lambda_1(S),\ldots,\lambda_n(S)$, then the condition number function $\kappa(S)$ is defined by

$$\kappa(S)=\begin{cases} \lambda_1(S)/\lambda_n(S), & \text{if} \quad \lambda_n(S)>0, \\ \infty, & \text{if } \lambda_n(S)=0 \text{ and } \lambda_1(S)>0, \\ 0, & \text{if} \quad S=0. \end{cases}$$

Obviously, the condition number $\kappa(S)$ reaches the global solution at S=0. To avoid this trivial situation, we consider that Λ does not contain the null matrix. The condition number function κ is proved to be strongly pseudoconvex on the cone f symmetric positive definite $n \times n$ matrices [34] and [35].

Example 5. Consider the following condition number optimization problem of a nonzero matrix

$$\begin{aligned} &\min \quad \kappa(A) \\ &\textit{subject to } A \in \Omega, \qquad (40) \end{aligned}$$

where A is a symmetric matrix defined by

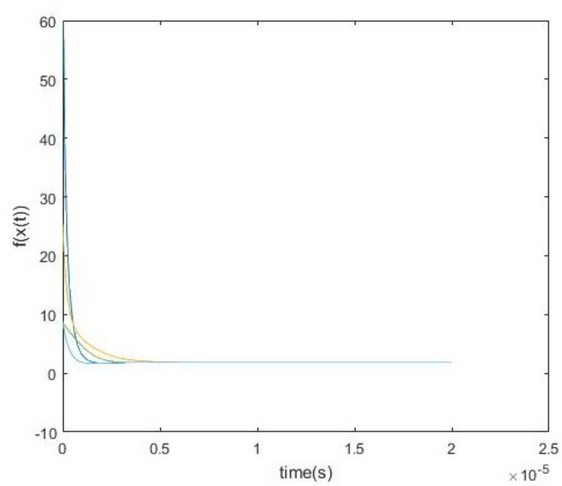

Figure 10: f(x(t)) along the solution of the neural network (3) with 4 random initial points of the Table A-1 to solve Example 4.

$$A=\begin{pmatrix} a^T x + a_0 & 0 \\ 0 & c^T x + c_0 \end{pmatrix},$$

Where

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$x = (x_1, x_2, x_3, x_4)^T \in \mathbb{R}^4, a = (3,1,2,1),$
$c = (2,1,1,0), a_0 = 4, c_0 = 2$

and Ω is defined by

$\Omega = \{x \in \mathbb{R}^4 : x_1 - x_2 = 4 \, and \, x_3 - x_4 = 3 \, and \, x \geq 0\}.$
Clearly, Ω is a compact convex set. If we denote $\lambda_{max}(A)$ and $\lambda_{min}(A)$ as the maximum and minimum eigenvalues of matrix A respectively, then the condition number of matrix A, $\kappa(A)$, is defined by $\kappa(A) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$. The pseudoconvexity of the objective function $\kappa(A)$ of x on Ω is proved in [34] and [35] and we can write $\kappa(A)$ as

$$\kappa(A) = \begin{cases} \dfrac{a^T x + a_0}{c^T x + c_0} & \text{if } a^T x + a_0 \geq c^T x + c_0, \\ \dfrac{c^T x + c_0}{a^T x + a_0} & \text{if } c^T x + c_0 \geq a^T x + a_0. \end{cases}$$

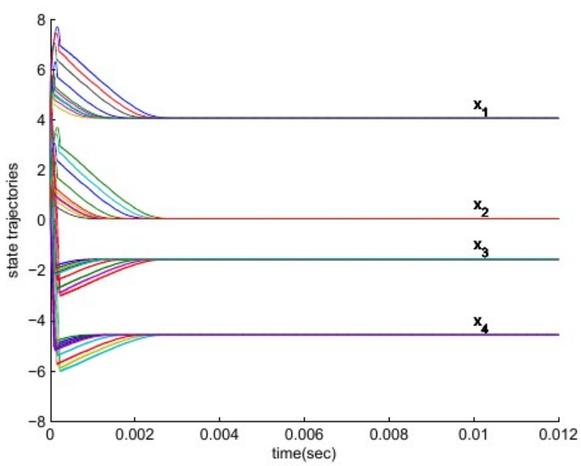

Figure 11: Transient behavior of the neural network (3) with 10 random initial points in Example 5.

We solve this problem by using the neural network (3). Fig. 11 shows the transient states of the neural network (3) with 10 random initial points, Fig. 12, shows the convergence of the condition number to optimal solution $\kappa(A) = 1$ with 10 random initial points. The authors in [22] proposed a model for solving the optimization problems similar to the optimization problem (1). In their model, there exists a penalty parameter θ and it needs Lipchitz constant of objective function f. Comparing to the proposed model in [22], our model does not need neither any penalty parameter θ in its structure nor any Lipchitz constant. It is clear that choosing inappropriate penalty parameter may lead to wrong solution or degeneracy.

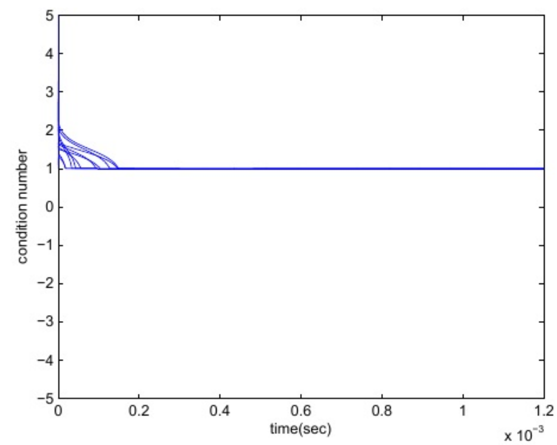

Figure 12: Transient and convergence behaviors of the optimal condition number evaluated by the neural network (3) with 10 random initial points in Example 5.

6. CONCLUSIONS

In this paper, a new one layer recurrent neural network for solving nonsmooth pseudoconvex optimization problems subject to linear constraints was proposed. By using Lyapunov theory and differential inclusion analysis, the convergence of the neural network to the optimal solution of the nonsmooth pseudoconvex optimization problems is guaranteed. We have shown the effectiveness and performance of the proposed neural network by some illustrative examples. Moreover, the proposed neural network is shown to be useful for condition number optimization problem. We will extend the proposed model for solving nonsmooth pseudoconvex optimization problems with general constraints in the future.

REFRENCES:

[1] D. Bertsekas, "Constrained optimization and Lagrange multiplier methods", Academic Press, USA (1982).

[2] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, "Nonlinear programming: theory and algorithms", Wiley, USA (1993).

[3] D. Tank, and J. Hopfield, "Simple neural optimization networks: An a/d converter, signal

decision circuit, and a linear programming circuit", IEEE Trans. Circuits Syst, 33, (1986), 533-554.

[4] M. Kennedy, and L. Chua, "Neural networks for nonlinear programming", IEEE Transactions on Circuits and Systems", 35, (1988), 554-562.

[5] S. Zhang, and A. G. Constantinides, "Lagrange programming neural networks", IEEE Trans. Circuits Syst., 39, no. 7, (1992), 441-452.

[6] J. Wang, Q. Hu, and D. Jiang, "A lagrangian network for kinematic control of redundant robot manipulators", IEEE Trans. Neural Netw., 10, no. 5, (1999), 1123-1132.

[7] Y. Xia, and J. Wang, "A recurrent neural network for solving nonlinear convex programs subject to linear constraints", IEEE Trans. Neural Netw., 16, no. 2, (2005), 379-386.

[8] Y. Xia, and J. Wang, "On the stability of globally projected dynamical systems", Journal of Optimization Theory and Applications, 106, (2000), 129-150.

[9] Y. Xia, H. Leung, and J. Wang, "A projection neural network and its application to constrained optimization problems", IEEE Transactions Circuits and Systems-I, 49, (2002), 447-458.

[10] Q. Liu, and J. Wang, "A one-layer recurrent neural network for constrained nonsmooth

Optimization", IEEE Trans. Systems, Man, and Cybernetics, Part B., 41, (2011), 1323-1333.

[11] Z. Guo, Q. Liu, and J. Wang, "A one-layer recurrent neural network for pseudoconvex optimization subject to linear equality constraints", IEEE Trans. Neural Networks., 22, (2011), 1892-1900.

[12] Q. Liu, Z. Guo, and J. Wang, "A one-layer recurrent neural network for constrained pseudoconvex optimization and its application for dynamic portfolio optimization", Neural Networks, 26, (2012), 99-109.

[13] C. Olsson, and F. Kahl, "Generalized convexity in multiple view geometry", J. Math Imaging Vis., 38, no. 1, (2010), 35-51.

[14] F. Forgo, and I. Joó, "Fixed point and equilibrium theorems in pseudoconvex and related spaces", J. Global Optim., 14, no. 1, (1999), 27-54.

[15] W. Dinkelbach, "On nonlinear fractional programming", Management Science, 13, no. 7, (1967), 492-498.

[16] L. Carosi, and L. Martein, "Some classes of pseudoconvex fractional functions via

the charnes-cooper transformation", in Generalized Convexity and Related Topics, Springer-Verlag, Germany(2006), 177-188.

[17] J. R. Barber, "Solid mechanics and its applications", Springer-Verlag, Germany (2004).

[18] G. Li, S. Song, C. Wu, and Z. Du, "A neural network model for non-smooth optimization over a compact convex subset", In: J. Wang, Z. Yi, J.M. ÿ Zurada, B.L. Lu, H.Yin, (eds.) ISNN 2006. LNCS, Springer, Heidelberg, (2006), 3971, 344-349.

[19] Q. Liu, and J. Wang, "A recurrent neural network for non-smooth convex programming subject to linear equality and bound constraints", Neural Information Processing (Lecture Notes in

Computer Science, 4233, (2006), 1004-1013.

[20] Q. Liu, and J. Wang, "A one-layer recurrent neural network for non-smooth convex optimization subject to linear equality constraints", Lecture Notes Comput Sci, 5507, (2009), 1003-1010.

[21] A. Hosseini, S. M. Hosseini, and M. Soleimani-damaneh, "A differential inclusion-based approach for solving nonsmooth convex optimization problems", Optimization, 62, (2013), 1203-1226.

[22] A. Hosseini, J. Wang, and S. M. Hosseini, "A recurrent neural network for solving a class of generalized convex optimization problems", Neural Networks, 44, (2013), 78-86.

[23] A. Hosseini, "A non-penalty recurrent neural network for solving a class of constrained optimization problems", Neural Networks, 73, (2016), 10-25.

[24] M. J. Ebadi, A. Hosseini, and M. M. Hosseini, "A projection type steepest descent neural network for solving a class of nonsmooth optimization problems", Neurocomputing, 235, (2017), 164-181.

[25] J. P. Aubin, and A. Cellina, "Differential inclusions", Springer-Verlag, Germany, (1984).

[26] F. Clarke, "Optimization and nonsmooth analysis", Wiley, USA, (1983).

[27] J. Penot, and P. Quang, "Generalized convexity of functions and generalized monotonicity of set-valued maps", J. Optim. Theory Appl., 92, (1997), 343-356.

[28] M. Forti, P. Nistri, and M. Quincampoix, "Generalized neural network for nonsmooth nonlinear programming problems", IEEE Trans. Circuits Syst. I, Reg. Papers, 51, no. 9, (2004), 1741-1754.

[29] W. Bian, and X. Xue, "Neural network for solving constrained convex optimization problems

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

with global attractivity", IEEE Trans. Circuits Syst. I, Reg. Papers, 60, no. 3, (2013), 710-723.

[30] Q. Liu, and J. Wang, "A one-layer recurrent neural network with a discontinuous hardlimiting activation function for quadratic programming", IEEE Transactions on Neural Networks, 19, (2008), 558-570.

[31] S. Qin, D. Fan, P. Su, and Q. Liu, "A simplified recurrent neural network for pseudoconvex optimization subject to linear equality constraints", Commun Nonlinear Sci Numer Simulat, 19, (2014), 789-798.

[32] X. Xue, and W. Bian, "Subgradient-based neural networks for nonsmooth convex optimization problems", IEEE Transactions on Circuits and Systems —I: Regular Papers, 55, no. 8, (2008), 2378-2391.

[33] A. Forsgren, P.E. Gill, and M.H. Wright "Interior methods for nonlinear optimization", SIAM Rev, 44, (2002), 525-597.

[34] X. Chen, R. Womersley, and J. Ye, "Minimizing the condition number of a gram matrix", SIAM Journal on Optimization, 21, no. 1, (2011), 127-148.

[35] P. Maréchal, and J. Ye, "Optimizing condition numbers", SIAM Journal on Optimization, 20, no. 2 (2010), 935-947.

[36] Q. Li, Y. Liu, and L. Zhu, "Neural network for nonsmooth pseudoconvex optimization with general constraints", Neurocomputing, 131, (2014), 336-347.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Appendix A:

Table A-1: Comparison of solutions of Example 4. with our proposed model, an interior point method [33] and a penalty based model [12].

Method	Initial point	Running time(second)	Penalty value	f	Feasibility
Herein	$(-2.52,\ 4.03,\ -1.06,\ 4.81)^T$	0.400532	-	1.8153	feasible
Interior point	ditto	4.642466	-	-3.5792e+20	infeasible
Penalty method	ditto	0.144208	0.5	-0.8166	infeasible
Penalty method	ditto	1458.963847	52	1.8153	feasible
Herein	$(7.26, 18.03, -3.73, 0.997)^T$	0.400096	-	1.8153	feasible
Interior point	ditto	2.170202	-	3.3845e+11	infeasible
Penalty method	ditto	0.692933	0.5	-0.8166	infeasible
Penalty method	ditto	1206.057870	52	1.8153	feasible
Herein	$(-1.13, 10.13, 3.92, -1.07)^T$	0.482393	-	1.8153	feasible
Interior point	ditto	2.188825	-	-2.0930e+12	infeasible
Penalty method	ditto	3.835418	0.5	-0.8166	infeasible
Penalty method	ditto	1557.441025	52	1.8153	feasible
Herein	$(-12.95, 5.82, -3.40, 14.81)^T$	0.463285	-	1.8153	feasible
Interior point	ditto	2.104498	-	-8.3806e+18	infeasible
Penalty method	ditto	0.424831	0.5	-0.8166	infeasible
Penalty method	ditto	10047.428608	52	1.8153	feasible