A NEURAL NETWORK FOR NONLINEAR OPTIMIZATION WITH GENERAL LINEAR CONSTRAINTS

Maryam Yashtini   Alaeddin Malek
e-mail: *m_yashtini@modares.ac.ir*   *e-mail: mala@modares.ac.ir*

Tarbiat Modares University, Department of Mathematics, P.O. Box 14115-175, Tehran, Iran

ABSTRACT

In this study, we investigate a novel neural network for solving nonlinear convex programming problems with general linear constraints. Furthermore, we extend this neural network to solve a class of variational inequalities problems. These neural networks are stable in the sense of Lyapunov and globally convergent to a unique optimal solution. The present convergence results do not requires Lipschitz continuity condition on the objective function. These models have no adjustable parameter and have a low complexity for implementation and converge to an exact optimal solution.

I. INTRODUCTION

Consider the following nonlinear programming problem

$$
\begin{aligned}
\text{Minimize} \quad & f(x) \\
\text{Subject to} \quad & Ax \geq b, \quad Bx = c, \\
& \ell \leq x \leq \hbar
\end{aligned}
\tag{1}
$$

where $f: R^n \to R$ is a continuously differentiable and convex $A \in R^{m \times n}$, $B \in R^{r \times n}$, $\ell, \hbar \in R^n$, $b \in R^m$ and $c \in R^r$.

It is well-known that nonlinear programming problems arise in a wide variety of scientific and engineering applications including regression analysis, image and signal processing, parameter estimation, filter design, robot control, etc [1]. Many of them have time-varying nature and thus have to be solved in real time [2,3]. Because of the nature of digital computers, convivial numerical optimization techniques may not be effective for such real-time applications. As parallel computational models, neural networks possess many desirable properties such as real-time information processing [4]. In particular, recurrent neural network for optimization have received tremendous interests in recent years. At present, there are several recurrent neural networks for solving nonlinear programming problem. Kennedy and Chua [5] presented a primal-dual neural network. Because the network contains a finite penalty parameter, thus it converges to an approximate solution only. To overcome the problem of the penalty parameter, a few primal-dual neural network with two and one-layer structure were developed [6-8]. It is well-known that neural networks with a low model complexity and fast convergence rate are very desirable [9,10]. In [11] Xia and Feng introduced a modified neural network for quadratic programming. Since in many real-world optimization problems, one has to deal with nonlinear optimization, the object of this study is to propose a primal-dual neural network for solving (1) and its dual problem. More exactly, the proposed neural network has one-layer structure without the need of computing an inverse matrix. Not only the state trajectory of proposed neural network converges globally to an equilibrium point, also compared with the existing convergence results, the present results do not require Lipschitz continuity condition the objective function. Furthermore we extend the proposed neural network to solve a class of monotone variational inequality problems.

II. NEURAL NETWORK MODEL

According to the Karush-Kuhn-Tucker (KKT) conditions for (1)[1], we see that x^* is an optimal solution of (1) if and only if there exist $y^* \in R^m$ and $z^* \in R^r$ such that $(x^*, y^*, z^*)^T$ satisfies the following conditions:

$$
\begin{cases}
x = P_X (x - \nabla f(x) + A^T y + B^T z) \\
\quad y = (y - Ax + b)^+ \\
\quad\quad Bx = c
\end{cases}
\tag{2}
$$

where $X = \{x \in R^n \mid \ell \leq x \leq \hbar\}$, $(y)^+ = [(y_1)^+, \ldots, (y_m)^+]$, and $(y_i)^+ = \max\{0, y_i\}$. Also $P_X(x) = [P_X(x_1), \ldots, P_X(x_n)]^T$ is defined for $i = 1, \ldots, n$

$$P_X(x_i) = \begin{cases} \ell_i & x_i < \ell_i \\ x_i & \ell_i \leq x_i \leq \hbar_i \\ \hbar_i & x_i > \hbar_i. \end{cases}$$

For simplicity we denote

$$W(u) = \begin{pmatrix} \nabla f(x) - A^T y - B^T z \\ Ax - b \\ Bx - c \end{pmatrix}.$$

Then (2) can be rewritten as in a compact form $P(u - W(u)) = u$, where $u = [x, y, z]^T \in R^{n+m+r}$ and $P(u) = [P_X(x), (y)^+, z]^T$.

In this paper, we propose a recurrent neural network for solving (1), with its dynamical equation being given by
State equation:

$$\frac{du}{dt} = -u + P(u - W(u)) = H(u) \tag{3}$$

Output equation:

$$x(t) = Du(t)$$

where $u \in R^{n+m+r}$ is a state vector, $x \in R^n$ is an output vector, $D = [I, O], I \in R^{n \times n}$ is an unit matrix, and $O \in R^{n \times (m+r)}$ is a zero matrix.

III. STABILITY AND CONVERGENT RESULTS

Lemma1: Let $\Omega = \{u = (x, y, z)^T \in R^{n+m+r} \mid x \in X, y \geq 0\}$. For any initial point $u_0 \in X \times R_+^m \times R^r$ there exists a unique solution $u(t) = (x(t), y(t), z(t))^T$ for (3).

Proof: P is locally Lipschitz continuous then according to the local existence and uniqueness theorem of ODEs [12], there exists a unique continuous solution of (3) for (t_0, T). We will show that $u(t)$ is bounded and the local existence for solution of (3) can be extended to global existence.

Theorem 1: Assume that $f(x)$ is strictly convex and twice differentiable. Then the proposed neural network of (3) with the initial point $u^0 \in \Omega$ is stable in the Lyaponov sense and globally convergent to the stationary point $u^* = (x^*, y^*, z^*)^T$, where x^* is the optimal solution of (1).

Proof: We define the following Energy function:

$$V(u) = -W(u)^T H(u) - \tfrac{1}{2}\| H(u) \|^2 + \tfrac{1}{2}\| u - u^* \|^2$$

Let $S_1 \subseteq R^{n+m+r}$ be a neighborhood of u^*. We show that $V(u)$ is a suitable Lyapunov function for dynamic system (3). By the results give in [13], we know that

$$-W(u)^T H(u) \geq \| H(u) \|^2 \tag{4}$$

$$(H(u) + u - u^*)^T (-H(u) - G(u)) \geq 0. \tag{5}$$

It is obvious that $V(u^*) = 0$ and for all $u \in S_1 \setminus \{u^*\}$, $V(u) \geq 0$.

In the following, we show that $\frac{dV(u)}{dt} \leq 0$. Since

$$\frac{dV}{dt} = \nabla E(u)^T \frac{du}{dt},$$

then from theorem 3.2 of [14],

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

where $\nabla W(u)$ denotes the Jacobian matrix of W. Then

$$\frac{dV(u)}{dt} = (W(u) - (\nabla W(u) - I)H(u) + u - u^*)^T H(u),$$

$$= (W(u) + u - u^*)^T H(u) + \| H(u) \|^2 - H(u)^T \nabla W(u)H(u).$$

From (5) we can write

$$(W(u) + u - u^*)^T H(u) \leq -(u - u^*)^T W(u) - \| H(u) \|^2.$$

Thus

$$\frac{dV(u)}{dt} \leq -(u - u^*)^T W(u) - H(u)^T \nabla W(u)H(u) \tag{6}$$

Since $\nabla W(u)$ is positive semidefinite then $(u - u^*)^T W(u) \geq 0$ and $H(u)^T \nabla W(u)H(u) \geq 0$ then

$$\frac{dV(u)}{dt} \leq -(u - u^*)^T W(u) - H(u)^T \nabla W(u)H(u) \leq 0. \tag{7}$$

Then the function $V(u)$ is an Energy function of (3).
From (7), $V(u)$ is monotonically nonincreasing for all $t \geq t_0$.
It is easy to see that $\phi = \{u \in R^{n+m+r} \mid V(u) \leq V(u^0)\}$ is bounded since

$$V(u_0) \geq V(u) \geq \frac{1}{2}\| H(u) \|^2 + \frac{1}{2}\| u - u^* \|^2 \geq \frac{1}{2}\| u - u^* \|^2 \geq 0,$$

therefore $T = \infty$.
Thus from positively invariance principle [12], trajectories $u(t)$ of (3) converge to ϑ as $t \to +\infty$, where

ϑ is the largest invariant set in $\Pi = \left\{ u \in \Phi \mid \dfrac{dV(u)}{dt} = 0 \right\}$.

Now we show that $\dfrac{du}{dt} = 0 \iff \dfrac{dV}{dt} = 0$. Clearly, if

$\dfrac{du}{dt} = 0$ then $\dfrac{dV}{dt} = \nabla V(u)^T \dfrac{du}{dt} = 0$.

To prove converse, let $\hat{u} = (\hat{x}, \hat{y}, \hat{z})^T \in \Pi$, then $\dfrac{dV(\hat{u})}{dt} = 0$. It is enough to show that $\dfrac{d\hat{x}}{dt} = 0$, $\dfrac{d\hat{y}}{dt} = 0$ and $\dfrac{d\hat{z}}{dt} = 0$. From (6), it follows that

$$(u - u^*)^T W(u) + H(u)^T \nabla W(u) H(u) = 0 \qquad (8)$$

Since $\nabla W(\hat{u})$ is positive semidefinite and $(u - u^*)^T W(\hat{u}) \geq 0$.

Furthermore, (8) implies $H(\hat{u})^T \nabla W(\hat{u}) H(\hat{u}) = 0$, $(W(\hat{u}) - W(u^*))^T (\hat{u} - u^*) = 0$, $(\hat{u} - u^*)^T W(\hat{u})^T = 0$.

Because

$$H(\hat{u})^T \nabla W(\hat{u}) H(\hat{u}) = [P_X (\hat{x} - \nabla f(\hat{x}) + A^T \hat{y} + B^T \hat{z}) - \hat{x}]^T$$
$$\times \nabla^2 f(\hat{x}) [P_X (\hat{x} - \nabla f(\hat{x}) + A^T \hat{y} + B^T \hat{z}) - \hat{x}] = 0.$$

The positive-definiteness of $\nabla^2 f(\hat{x})$ implies that $[P_X (\hat{x} - F(\hat{x}) + A^T \hat{y} + B^T \hat{z}) - \hat{x}] = 0$, i.e. $\dfrac{d\hat{x}}{dt} = 0$.

Also $(W(\hat{u}) - W(u^*))^T (\hat{u} - u^*) = 0$, then

$$(\nabla f(\hat{x}) - \nabla f(x^*))^T (\hat{x} - x^*) =$$
$$(x - x^*)^T \nabla^2 f(x_\mu)(\hat{x} - x^*) = 0$$

where $x_\mu = (1 - \mu)\hat{x} + \mu x^*$, for all $0 \leq \mu \leq 1$. It follows that $\hat{x} = x^*$, thus $B\hat{x} - c = 0$, i.e. $\dfrac{d\hat{z}}{dt} = 0$.

Now, consider that $(\hat{u} - u^*)^T W(\hat{u})^T = 0$. This gives the following form

$$(\hat{x} - x^*)^T (\nabla f(\hat{x}) - A^T \hat{y} - B^T \hat{z}) + (\hat{y} - y^*)^T (A\hat{x} - b)$$
$$+ (\hat{z} - z^*)^T (B\hat{x} - c) = 0.$$

Since $\hat{x} = x^*$, it is equivalently written as bellow

$$(\hat{y} - y^*)^T (A\hat{x} - b) = 0. \qquad (9)$$

then

$$\hat{y}^T (A\hat{x} - b) = (y^*)^T (A\hat{x} - b) = (y^*)^T (Ax^* - b) = 0.$$

Furthermore, $\hat{y}^T (A\hat{x} - b) = 0$, $\hat{y} \geq 0$ and $A\hat{x} - b \geq 0$ if and only if $(\hat{y} - A\hat{x} + b)^+ - \hat{y} = 0$ i.e. $\dfrac{d\hat{y}}{dt} = 0$.
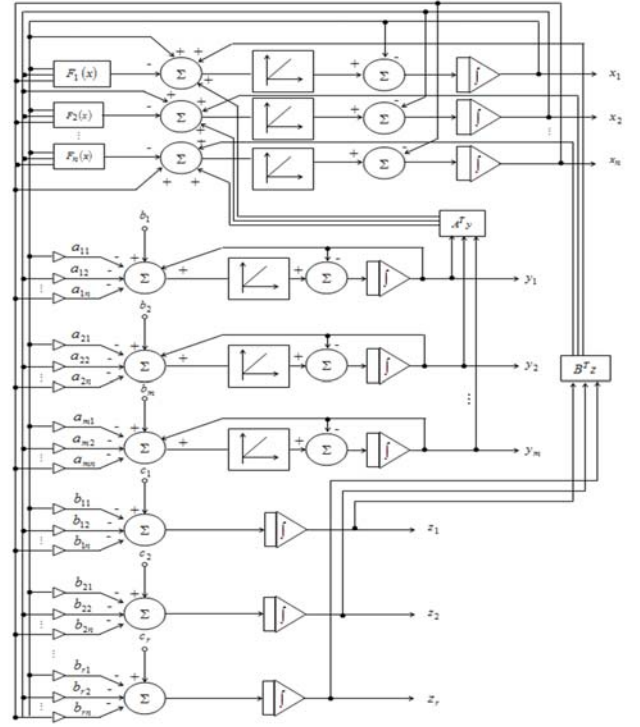

Figur 1. Simplified architecture of the neural network model (12).

Therefore $\dfrac{du}{dt} = 0$ if and only if $\dfrac{dV}{dt} = 0$.

Then the proposed neural network in (3) is globally convergent to the optimal solution of (1).

IV. MONOTONE VARIATIONAL INEQUALITIES PROBLEM

Consider the following variational inequalities problem with general linear constraints:

$$(x - x^*)^T F(x^*) \geq 0 \qquad \text{for all} \quad x \in \Omega_1, \qquad (10)$$

where $F : R^n \to R^n$ is continuously differentiable and $\Omega_1 = \{ x \in R^n \mid Ax \geq b, Bx = c, x \in X \}$. It is well-known that $x^* \in X$ is a solution of (10) if and only if there exists (y^*, z^*) such that $u^* = (x^*, y^*, z^*)^T$ is a solution of the following variational inequalities problem:

$$(u - u^*)^T U(x^*) \geq 0 \qquad \text{for all} \quad x \in \Omega_2 \qquad (11)$$

where

$\Omega_2 = \{ u = (x, y, z)^T \in R^{n+m+r} \mid x \in X, y \geq 0 \}$ and

$$U(u) = \begin{pmatrix} F(x) - A^T y - B^T z \\ Ax - b \\ Bx - c \end{pmatrix}.$$

According to the projection theorem [15], we see that (11) can be formulated as $P(u - U(u)) = u$, where

$$P(u) = [P_X(x), (y)^+, z]^T.$$

Thus, as extension of the proposed neural network in (3) we propose the following dynamical equation for solving problem (10) as well.
State equation:

$$\frac{du}{dt} = -u + P(u - U(u)) \qquad (12)$$

Output equation:

$$x(t) = Du(t)$$

where $u \in R^{n+m+r}$ is a state vector, $x \in R^n$ is an output vector, $D = [I, O], I \in R^{n \times n}$ is an unit matrix, and $O \in R^{n \times (m+r)}$ is a zero matrix. The proposed neural network can be implemented by a circuit with a single-layer structure as shown in Figure. 1
From the analysis of Theorem 1, we get the global convergence result on the neural network in (12).

Theorem 2: Assume that F is differentiable and strictly monotone for all $x \in X$. Then the neural network model (12) with the initial point $u_0 = (x_0, y_0, z_0)^T$ is globally convergent to the stationary point $u^* = (x^*, y^*, z^*)$, where x^* the optimal solution of (10).
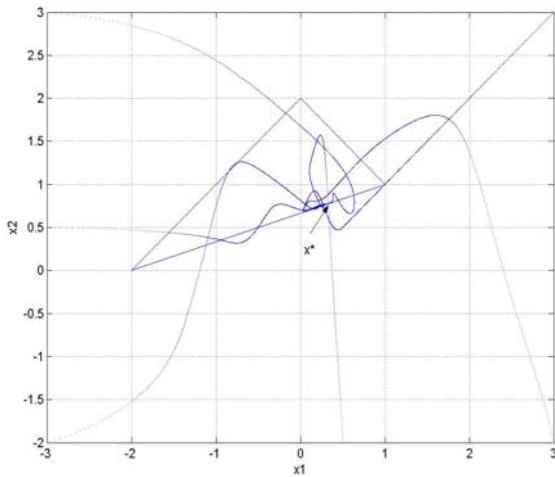

Figure 2. The transient behavior of $x(t)$ using the neural network model (3) for example 1.

V. SIMULATION RESULTS

In this section, we discuss the simulation results through two examples. The simulation is conducted in MATLAB with the 4th order of Runge–Kutta technique. We use step size 0.003 and the stopping criterion is $\| u(t) - u^* \| \leq 10^{-10}$ in all our run.

Example 1: Consider the following nonlinear programming problem

Minmize $\quad f(x) = \frac{1}{4} x_1^4 + 0.5 x_1^2 + \frac{1}{4} x_2^4 + 0.5 x_2^2 - 0.9 x_1 x_2$

Subject to $\quad -x_1 - x_2 \geq -2$

$\qquad\qquad x_1 - x_2 \geq -2$

$\qquad\qquad x_1 - 3x_2 = -2$,

$\qquad\qquad 0 \leq x \leq 1.$

This problem has an optimal solution $x^* = (0.346, 0.782)^T$ (three effective digits). Note that $\nabla^2 f(x)$ is positive definite on R_+^n. Theorem 1 guarantees that neural network model (3) converges to x^* globally. Figure. 2 displays the transient behavior of $x(t)$ with five initial point where $y = (1,1)^T$ and $z = 1$ are fixed. All trajectories converge to optimal solution x^* and $y^* = (0,0)^T$ and $z^* = 0.316$.

Example 2: Consider the nonlinear variational inequality problem (10). The mapping F and the constraint set Ω_1 defined by

$$F(x) = \begin{bmatrix} 3x_1 - \dfrac{1}{x_1} + 3x_2 - 2 \\ 3x_1 + 3x_2 \\ 4x_3 + 4x_4 \\ 4x_3 + 4x_4 - \dfrac{1}{x_4} - 3 \end{bmatrix}$$

and $\Omega_1 = \{x \in R^n \mid x_1 + x_2 = 1, x_3 + x_4 \geq 0, \ell \leq x \leq \hbar\}$, where $\ell = (0.1, 0, 0, 1)^T$ and $\hbar = (10, 10, 10, 10)^T$. This problem has one optimal solution $x^* = (1, 0, 0, 1)^T$. All simulation results show that the proposed neural network (12) convergent $(x^*, y^*, z^*)^T$ where $y^* = 0$ and $z^* = 0$. Figure 3, show transient behavior $u(t)$ and $\| u(t) - u^* \|$ with six and ten random initial point, respectively.
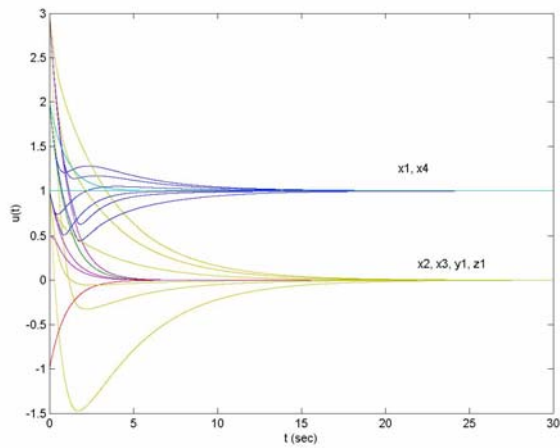
(a)
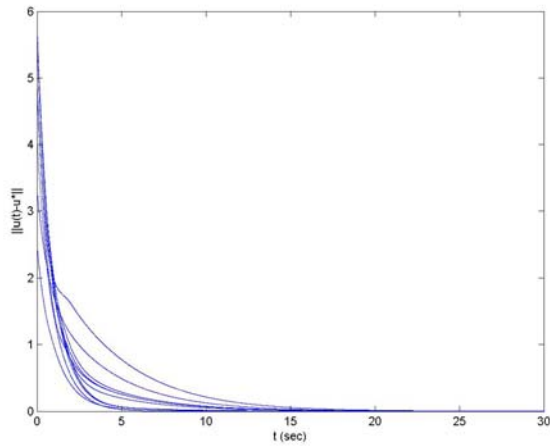
(b)

Figure 3. Simulation results of the neural network model (12) for example 2. (a) Transient behavior of $u(t)$ with six initial points. (b) Transient behavior of the norm $\| u(t) - u^* \|$ with ten initial points.

CONCLUSION REMARKS

We have proposed a recurrent neural network model for solving nonlinear convex programming problems with general linear constraints. It is shown here that the proposed neural network is stable in the sense of Lyapunov and globally convergent to an optimal solution under strictly convex condition of the objective function. This neural network has a simple single-layer sruture and does not have any adjustable parameter then it is very simple to use. The simulation results have demonstrated globally convergence behaviors and characteristics of the proposed neural network for solving several nonlinear programming problems.

REFERENCES

1. M. S. Bazaraa, H. D. Sherali, C. M. Shetty, NonlinearProgramming: Theory and Algorithms, John and Sons, Inc., New York: Wiley, 1990.
2. T.Yoshikawa, Foundations of Robotics: Analysis and control, MIT Press, Cambridge, MA, 1990.
3. N. Kalouptisidis, Signal Processing System, Theory and Design. Wiley, New York, 1997.
4. D. W. Tank, J. J. Hopfield, Simple neural optimization networks: an A/D converter, Signal decision circuit, and a linear programming circuit, IEEE Transaction on circuit and system, Vol. 33, pp. 533-541, 1986
5. M. P. Kennedy, L. O. Chua, Neural Networks for Nonlinear Programming , IEEE Transaction on circuit and system, Vol. 35, No. 5, pp. 554-562, 1988.
6. A. Malek, A.Yari, Primal-Dual Solution for the Linear Programming Problem using Neural Networks, App. Math. Comput. Vol. 169, pp. 198-211, 2005.
7. M. Yashtini, A. Malek, Solving complementarity and variational inequalities problem using neural network, Applied Mathematics and Computation, Vol. 190, pp. 216-230, 2007.
8. M. Yashtini, A. Malek, A discrete-time neural network for solving nonlinear convex problems with hybrid constraints, Applied Mathematics and Computation,Inpress.
9. Y. S. Xia and H. Leung, J. Wang, Projection Neural Network and its Application to Constrained Optimization Problems, IEEE Trans. Circuits Syst. I, Vol. 49 , pp. 447-458, 2002.
10. Y. Xia, J. Wang, Global exponential stability of recurrent neural network for solving optimization and related problem, IEEE Transaction on Neural Network, Vol. 11, No. 4, 2000.
11. Y. Xia, G. Feng, A modified neural network for quadratic programming with real time applications, Neural Information Processing, Vol. 3, No. 3, 2004.
12. R. K. Miller and A. N. Michel, Ordinary Diffrential Equations, Newral Network: Academic, 1982.
13. J. S. Pang, A posteriori error bounds for the linearly-constrained variational inequality problem," Math. Oper. Res. vol. 12 , pp. 474-484, 1987.
14. M. Fukushiima, "Equivalent differentiable optimization problems and decent methods for asymmetric variational inequality problem," Math. Program. Vol. 53, pp. 99-110, 1992.
15. D. P. Bertsekas, J. N. Tsitsikils, Parallel and Sistributed Computation: Neumerical Methods. Englewood Cliffs, NJ: Prentice-Hall, 1989.