A NONFEASIBLE QUADRATIC APPROXIMATION RECURRENT NEURAL NETWORK FOR EQUALITY CONSTRAINED OPTIMIZATION PROBLEMS

Maria P. Barbarosou* and Nicholas G. Maratos**

**Hellenic Air-Force Academy, Department of Aeronautical Sciences,*
Division of Electronics and Communication Engineering
Dekelia Air Force Base, GR 1010, Greece,
E-mail: mbarbar@teipir.gr.

***National Technical University of Athens, School of Electrical and Computer Engineering,*
9 Iroon Polytechniou St., GR 15773, Athens, Greece,
E-mail: maratos@ece.ntua.gr.

Abstract

Convex optimization techniques are widely used in the design and analysis of communication systems and signal processing algorithms. In this paper a novel recurrent neural network is presented for solving nonlinear strongly convex equality constrained optimization problems. The proposed neural network is based on recursive quadratic programming for nonlinear optimization, in conjunction with homotopy method for solving nonlinear algebraic equations. It constructs generally a nonfeasible trajectory which satisfies the constraints as $t \rightarrow$*. The boundedness of solutions and the global convergence to the optimal point of the problem are proven. The correctness and the performance of the proposed neural network are evaluated by simulation results on illustrative numerical examples.*

1. Introduction

The use of convex optimization is ubiquitous in communications and signal processing. Many problems in these fields can be converted into convex optimization problems, which greatly facilitate their analysis and numerical solutions [1]-[2].

Consider the following equality constrained optimization problem:

(P) $min\ \{ f(x),\ A^T x - b =$ (1)

where $f: R^n \rightarrow$, *A* an nxmatrix with $m <$ and *b* an *m* vector. We make the following assumption, standard for quadratic approximation programming:

Assumption: (a) The function is strongly convex and twice continuously differentiable in . (b) The matrix *A* has full rank.

Since Tank and Hopfield's pioneering work [3]-[5] on linear programming neural network and analogue circuits, the recurrent neural network approach for solving nonlinear programming has received a great of attention in the last two decades, see [6]-[13] and the references therein. Different approaches towards designing such networks have been developed. Some neural networks employed penalty functions [3]–[7], or the logarithmic barrier function [8], while others [9]-[10] make direct use of the Lagrangian function. In [11] a neural network for solving linear projection equations is described. More recently, neural networks based on gradient projection method for nonlinear programming are designed [12]-[13].

The proposed neural network does not make use of a penalty function or of a projection equation. It solves the problem directly, based on a combination of the recursive quadratic programming [14] and the continuous Newton-Raphson method [14] for solving the constraint equations.

The reminder of the paper is organized as follows. The new neural network description is presented in Section II. In Section III we prove the global convergence to the optimal point of (P). Illustrative examples are given in Section IV. Finally Section V concludes the paper.

2. Derivation of the proposed neural network

Let $L(x,\lambda) = f(x) + \lambda^T(A^T x -$ be the Lagrangian function for problem (P), where $\lambda \in l$ is the vector of Lagrangian multipliers.

Since is strongly convex, It is well known [14] that if the optimal point of (P) exists, then it is unique, and also there exist vector such that: $\nabla L(x^*,\lambda^*) =$ and $A^T x^* - b =$, where $\nabla L(x,\lambda) = \nabla f(x) +$.

In the first instance, we consider the following system of implicit ordinary differential equations:

$$\nabla L(x(t),\lambda(t)) = e^{-\rho t}\nabla L(x_o, \lambda \quad (2.1)$$

$$A^T x(t) - b = e^{-\rho t}\,(A^T x_o - b \quad (2.2)$$

where $(x(t),\lambda($ the solution of system (2) with initial point $(x_o,\lambda_o) = (x(0),\lambda($ and ρ positive constant. Obviously, the norms of $\nabla L(x,$ and the equality constraints are decreasing along the solution of system (2). Differentiation of (2) with respect to t gives:

$$\frac{\partial^2 L(x,\lambda)}{\partial^2 x}\dot{x} + \frac{\partial^2 L(x,\lambda)}{\partial\lambda\partial x}\dot{\lambda} = -\rho e^{-\rho t}\nabla L(x_o,\lambda_o)$$

$$A^T\dot{x} = -\rho e^{-\rho t}\,(A^T x_o - b)$$

where stand for x and λ respectively.

Since $\partial^2 L(x,\lambda)/_{\partial^2 x} = \partial^2 f(x)/_{\partial}$ the above system in matrix form is written as:

$$\begin{bmatrix} \frac{\partial^2 f(x)}{\partial^2 x} & A \\ A^T & 0 \end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{\lambda}\end{bmatrix} = -\rho\begin{bmatrix}\nabla L(x,\lambda)\\ A^T x - b\end{bmatrix} \quad (3)$$

The system (3) is linear with respect of the vector $[\dot{x}^T, \dot{\lambda}$. We solve the system via QR decomposition of the matrix *A* [15]. Namely, *A* is decomposed as:

$$A = Q\begin{bmatrix}R\\0\end{bmatrix} = [Q_1\; Q_2]\begin{bmatrix}R\\0\end{bmatrix} = Q_1 R$$

where *Q* is an n unitary matrix, *R* is an mx upper triangular matrix. The matrices Q_1 and Q_2 consist of the first *m* and the last *n-m* columns of Q, respectively. Under the *Assumption*, the matrix $\begin{bmatrix} \frac{\partial^2 f(x)}{\partial^2 x} \\ A^T \end{bmatrix}$ is invertible. So the system (3) can be solved for $[\dot{x}^T, \dot{\lambda}$ yielding:

$$\dot{x} =$$

$$-\rho[M(x)\nabla f(x) + N(x)(A^T x - b] \quad (4.1)$$

$$\dot{\lambda} =$$

$$-\rho[\lambda + N(x)^T\nabla f(x) + O(x)(A^T x - \quad (4.2)$$

where:

$$M(x) = Q_2{}^T\left(Q_2{}^T\frac{\partial^2 f(x)}{\partial^2 x}Q_2\right)^{-1},$$

$$N(x) = \left(I_n - M(x)\frac{\partial^2 f(x)}{\partial^2 x}\right)Q_1(R^T,$$

$$O(x) = -R^{-1}Q_1{}^T\frac{\partial^2 f(x)}{\partial^2 x}N(,$$

In the following proposition a Lyapunov function for dynamical system (4) is given.

Proposition: Let the *Assumption* hold, then the function $V : R^{n+m}$ – be defined as:

$$V(x,\lambda) = \frac{1}{2}(\|\nabla L(x,\lambda)\|^2 + \|A^T x - b\|^2)$$

is decreasing along the solution of (4) and approaches zero as time tends to infinity, where $\|$ is the Euclidean norm.

Proof: Finding the directional derivative of $V(x,$ in the direction of the solution of (4) we obtain

$$\frac{dV(x,\lambda)}{dt} = \nabla_x V(x,\lambda)\dot{x} + \nabla_\lambda V(x,\lambda)\dot{\lambda} =$$

$$\begin{bmatrix}\nabla L(x,\lambda)\\ A^T x - b\end{bmatrix}^T\begin{bmatrix} \frac{\partial^2 f(x)}{\partial^2 x} & A \\ A^T & 0 \end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{\lambda}\end{bmatrix}$$

where and denote the gradients with respect to and , respectively.

Since the systems (3) and (4) are equivalent, from (3) we have

$$\frac{dV(x,\lambda)}{dt} =$$

$$-2\rho(\|\nabla L(x,\lambda)\|^2 + \|A^T x - b\|^2) < 0$$

which means that

$$\frac{dV(x,\lambda)}{dt} = -\rho V(x,\lambda) \quad (5)$$

From (5) it follows that the function $V(x,$ is decreasing exponentially along the solution of (4). This proves the assertions of the proposition.

The dynamics of the proposed neural network are defined in explicit form, by the system of differential equations (4.1). This is an autonomous dynamical system for x, since the multipliers λ on its right hand side has been eliminated. A block diagram realization of our neural network is given in Fig.1.

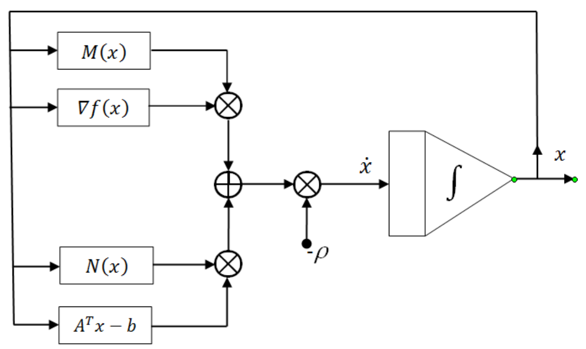

Figure 1. Block diagram realization of the proposed neural network

3. Global convergence

The solution of a dynamical system is said to be globally convergent to a point if for any initial point $x_o \in I$ $\lim_{t\to\infty}\{x(t)\} =$ This result can be derived [16] by the boundedness of the solution , and the existence of a Lyapunov function with zero gradient at .

Theorem: Let the *Assumption* hold, and let be the unique minimize of problem (P). Then the solution of (4.1) starting from any initial point, is bounded, extends to infinite time and converges to , i.e. $\lim_{t\to\infty}\{x(t)\} =$

Proof: The following relationships are used throughout this proof

$$Q_1 Q_1{}^T + Q_2 Q_2{}^T = I_n, Q_1 R = A,\ Q_1{}^T Q_2 = 0$$
$$Q_1{}^T Q_1 = I_m,\ A^T x^* - b =.$$

We shall first show that the solution x of (4.1) is bounded. It holds that

$$x - x^* = (Q_1 Q_1{}^T + Q_2 Q_2{}^T)(x - \quad (6)$$

Premultiplication (4.1) by $(x-x^*)^T Q_1($ after simple algebra, we get

$$\frac{d\|Q_1{}^T(x-x^*)\|^2}{dt^2} = -2\rho\|Q_1{}^T(x-x^*)\|^2 \quad (7)$$

This simply states that $\|Q_1{}^T(x-x$ hence $\|A^T x -$ are bounded along the solution of (4.1). From this result and Proposition we have that $\|\nabla L(x,$ is also bounded along the solution of (4). Since $\|Q_2{}^T \nabla f(x)\| = \|Q_2{}^T \nabla L(x,$ it follows that $\|Q_2{}^T \nabla f($ is also bounded. Premultiplication (4.1) by $(x-x^*)^T Q_2($ after simple algebra, we get

$$\frac{d\|Q_2{}^T(x-x^*)\|^2}{dt^2} =$$
$$-2\rho(x-x^*)^T Q_2 Q_2{}^T M(x)\nabla f(x)$$
$$+2\rho(x-x^*)^T Q_2 Q_2{}^T M(x)\frac{\partial^2 f(x)}{\partial^2 x}(x-x^*)$$
$$-2\rho(x-x^*)^T Q_2 Q_2{}^T M(x)\frac{\partial^2 f(x)}{\partial^2 x} Q_2{}^T Q_2(x-x^*)$$

At this point we use the strongly convexity of the objective function, so it holds that [14]

$$\exists m > 0 : \frac{\partial^2 f(y)}{\partial^2 y} > m\|y\|^2, \forall y \in R^n$$

From the above property and the boundedness of $\|Q_2{}^T \nabla f($ and $\|Q_1{}^T(x-x$, it follows that for some finite a, β, γ it holds that

$$\frac{d\|Q_2{}^T(x-x^*)\|^2}{dt^2} \le$$
$$-a\|Q_2{}^T(x-x^*)\|^2 + \beta\|Q_2{}^T(x-x^*)\| + \gamma$$

This result means that when $\|Q_2{}^T(x-x^*$ touches a finite upper bound, it will start to decrease along the solution of (4.1), therefore $\|Q_2{}^T(x-x$ is bounded. Thus from (6) the solution of (4.1) is bounded, hence it extends to infinite time [16]. Since x is bounded, It can be proved easily from (4.2) that λ is also bounded.

Let the set *D* be defined as:

$$D = \left\{(x,\lambda) \in R^{n+m} : {dV(x,\lambda)}/{dt} = 0\right\}$$

where $V(x$ is the function of Proposition 1. Then from (5) we have

$$D = \{(x,\lambda) \in R^{n+m} :$$
$$\|\nabla L(x,\lambda)\| = 0 \ and\ \|A^T x - b\| =\}$$

hence because of the *Assumption* $D = \{x^*,$. Since $(x(t),\lambda($ is bounded and satisfies Proposition, from LaSalle's Theorem [16] it follows that $(x(t),\lambda(t)) \to D = (x^*,$ as $t \to$. This competes the proof.

4. Numerical Examples

The performance of our neural network is evaluated by using MATLAB for several test problems. In this section two illustrative examples are given. Example 1 has quadratic objective function and satisfies both parts of *Assumption*. To demonstrate the effectiveness of our neural network in more general optimization problems, we choose Example 2, whose objective function is a Gaussian as shown in Fig. 2, that is pseudoconvex. So, Example 2 satisfies only the part (b) of *Assumption*.

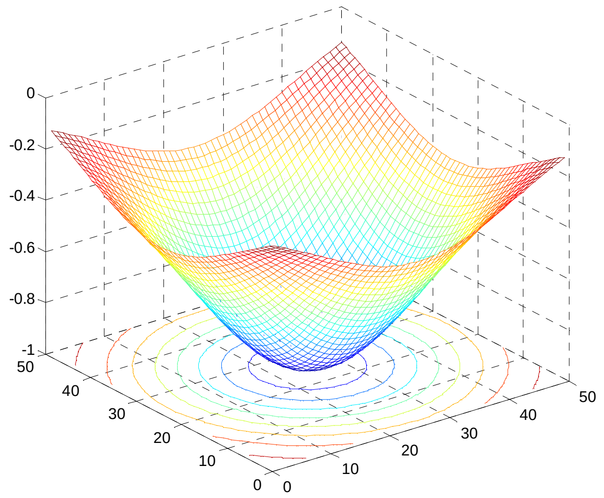

Figure 2. The 2D Caussian function of Example 2

Example 1: Consider the following strongly convex problem [6], with n = and m = :

$$\min_{x\in R^6}\{\|x\|^2 : A^T x - b = 0\}$$

where $b^T = [2 \quad -1 \quad$ - and

$$A^T = \begin{bmatrix} 2 & -1 & 4 & 0 & 3 & 3 \\ 5 & 1 & -3 & 1 & 2 & 0 \\ 1 & -2 & 1 & -5 & -1 & 4 \end{bmatrix}$$

This problem has a unique global minimizer at x^* [0.08824674 0.010828343 0.27326648 0.50466163 0.38281032 -0.30970696], written to eighth decimal place. The trajectories x obtained by the proposed neural network with $\rho =$, starting from five random non-feasible initial points in (-1 1), are shown in Fig.2a. Fig.2b shows the convergence of the cost function for each case. At the end of the simulation, all trajectories reach with final error $e = \|x(t) - x$ of order 10^{-6}.

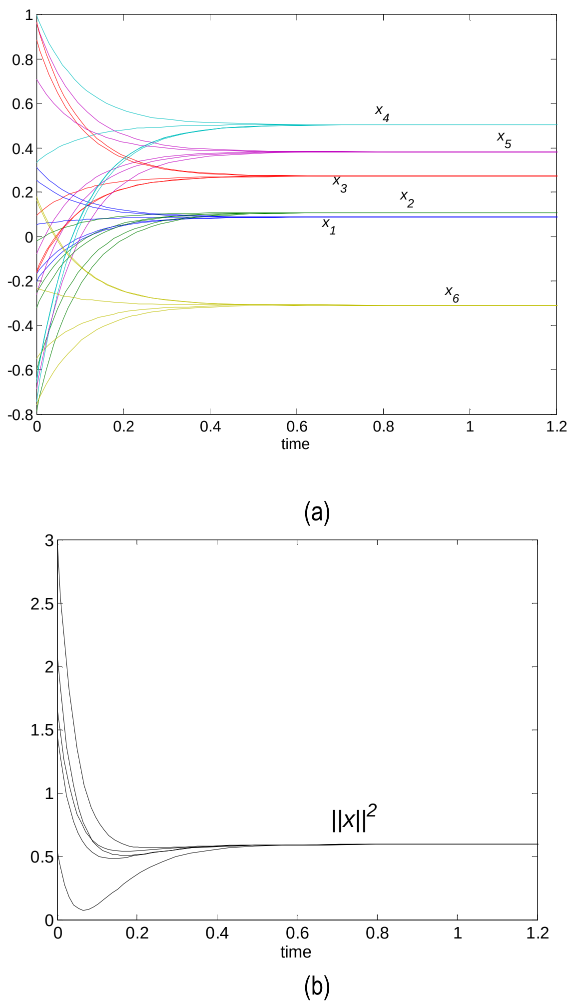

Figure 3. Example 1: (a) Trajectories x of the proposed neural network for 5 random initial points and (b) the corresponding cost functions, vs time

Example 2: Consider the following pseudoconvex optimization problem [9], with n = and m = :

$$\min_{x\in R^2}\{e^{-(x_1^2+x_2^2)} : A^T x - b = 0\}$$

where $A = \begin{bmatrix}0.7\\0.5\end{bmatrix}$ and $b = 0.8$.

This problem has a unique global minimizer at x^* [0.62745172 0.500937796], written to eighth decimal place. Fig. 3a shows the trajectories of the proposed neural network with $\rho =$, starting from fifteen non-feasible initial points, random generated from the uniform distribution over (0,1). Fig.3b shows the convergence of the cost function for each case. At the end of the simulation, all trajectories reach with final error $e = \|x(t) - x$ of order 10^{-6}.

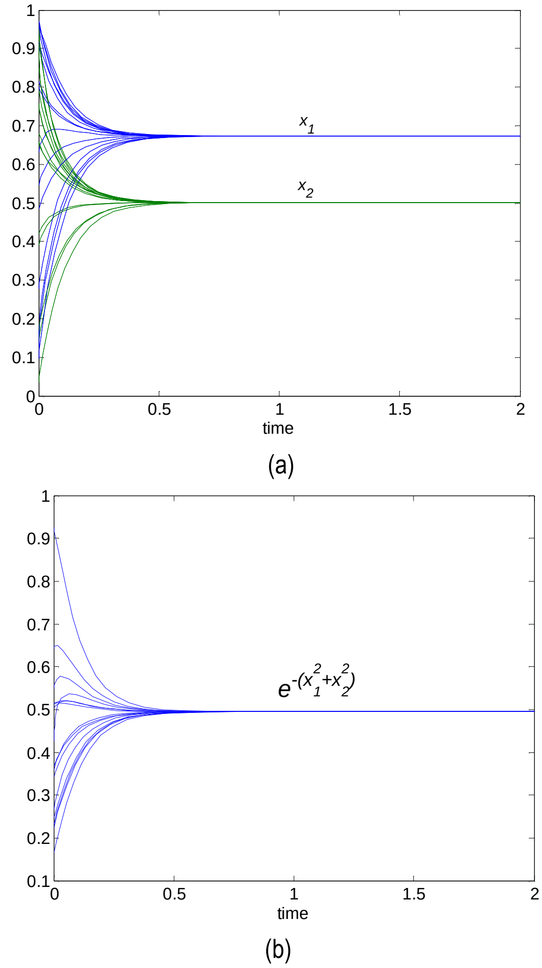

Figure 4. Example 2: (a) Trajectories $\boldsymbol{x}$ of the proposed neural network for 15 random initial points and (b) the corresponding cost functions, vs time.

5. Conclusions

In this paper a recurrent neural network for strongly convex constrained optimization problem is presented, based on quadratic approximation method for nonlinear programming. If initial point is non-feasible, the proposed neural network defines a non-feasible trajectory which satisfies the constraints as $t \to$. Global convergence is proven. Simulation on illustrative numerical examples substantiates the effectiveness and the correctness of the proposed neural network.

References

[1] Z. Luo and W. Yu, "An introduction to convex optimization for communications and signal processing", *IEEE on Selected Areas in communications*, vol.24, pp. 1426-1438.

[2] Z. Q. Luo, 'Applications of convex optimization in signal processing and digital communication", *Math Program.*, 2003, pp. 177-207.

[3] D.W. Tank and J.J. Hopfield, "Simple neural optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit", *IEEE Trans. Circuits and Systems*, vol. 33, 1986, pp. 533-541.

[4] L.O. Chua and G.-N. Lin, "Nonlinear programming without computation", *IEEE Trans. Circuits and Systems*, vol. 31, 1984, pp. 182-188.

[5] M. Kennedy and L.O. Chua, "Unifying the Tank and Hopfield linear programming circuit and the canonical nonlinear programming circuit of Chua and Lin", *IEEE Trans.Circuits and Syst.*, vol. 34, 1986, pp. 210-214.

[6] W. E. Lillo S. Hui and S. H. Zak, "Neural Networks for constrained optimization problems", *Int. J. of Circuit Th. And Appl.*, vol. 21, 2000, pp. 385-399.

[7] A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing*, John Wiley & Sons, 1993.

[8] P.T. Krasopoulos and N.G. Maratos, "A neural network for convex optimization", in *Proc. ISCAS 2006*, pp.747-750.

[9] S. Zhang and A.G. Constandinides, "Lagrange programming neural networks", *IEEE Trans.Circuits System II*, vol. 39,1992, pp.441-452.

[10] S. Zhang, X. Zhu and L.-H. Zou, "Second-order neural nets for constrained optimization", *IEEE Trans. Neural Networks*, vol. 3, 1992, pp. 1021-1024.

[11] Y. Xia H. Leuhg and J. Wang, "A projection neural network and its application to constrained optimization problems", *IEEE Trans. Circuits and Systems,* vol. 13, 2002, pp. 337-350.

[12] M. Barbarosou and N.G. Maratos, "A non -feasible gradient projection recurrent neural network for equality constrained optimization", in *Proc. IJCNN 2004*, pp.2251-2256.

[13] M. Barbarosou and N.G. Maratos, "A non -feasible gradient projection recurrent neural network for equality constrained optimization problems", *IEEE Trans. Neural Networks*, vol 19, 2008, pp. 1665-1677.

[14] D. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1995.

[15] G. Colub and C. Van Loan, *Matrix Computations*, Johns Hopkins University, 1996.

[16] N. Rouche, P. Hebets, and M. Laloy, *Stability Theory by Liapunov's Direct Method*, Springer-Verlag, New York, 1977.