Optimization
Iranian Journal of Optimization
Volume 11, Issue 1, 2019, 1-7
Research Paper
Online version is available on: www.ijo.iaurasht.ac.ir

Islamic Azad University
Rasht Branch
ISSN: 2588-5723
E-ISSN:2008-5427

A Recurrent Neural Network Model for Solving CCR Model in Data Envelopment Analysis

Masomeh Abbasi* and Abbas Ghomashi

Department of Mathematics, Kermanshah Branch, Islamic Azad University, Kermanshah, Iran

Received: 20 November 2017
Accepted: 24 April 2018

Keywords:
Recurrent neural network
Gradient method
Data envelopment analysis
CCR
Stability
Global convergence

Abstract

In this paper, we present a recurrent neural network model for solving CCR Model in Data Envelopment Analysis (DEA). The proposed neural network model is derived from an unconstrained minimization problem. In the theoretical aspect, it is shown that the proposed neural network is stable in the sense of Lyapunov and globally convergent to the optimal solution of CCR model. The proposed model has a single-layer structure. A numerical example shows that the proposed model is effective to solve CCR model in DEA.

***Correspondence E-mail**: mas.abbasi81@gmail.com

INTRODUCTION

DEA is a nonparametric approach in operations research to estimate the performance evaluation and relative efficiency of a set of homogeneous DMUs such as business units, government agencies, police departments, hospitals, educational institutions and etc. Charnes et al in their seminal DEA model (CCR model) in 1978 proposed that the efficiency of a DMU can be obtained as the maximum of a ratio of weighted outputs to weighted inputs, subject to the condition that the same ratio for all DMUs must be less than or equal to one (Charnes et al., 1979). Linear programming is needed to recognize efficient and inefficient DMUs in CCR models. The dimension and denseness of the structure of linear programming increases as the numbers of DMUs and numbers of input and output increases, in this case, the numerical methods become less affected for solving corresponding linear programming. One promising approach to solve CCR models is to employ the artificial neural networks based on circuit implementation. Mathematically, the optimization problem to be solved is mapped into a dynamical system so that whose state output can give the optimal solution and the optimal solution is then obtained by tracking the state trajectory of the designed dynamical system based on the numerical ordinary differential equation technique. A neural network with a good computational performance should satisfy threefold. First, the global convergence of the neural networks with an arbitrarily given initial state should be guaranteed. Second, the network design preferably contains no variable parameter. Third, the equilibrium points of the network should correspond to the exact or approximate solution (Xia & Wang, 1998). Solving optimization problems using recurrent neural networks has fascinated much attention since seminal work of Tank and Hopfield (Tank & Hopfield, 1986). Many neural network for constrained optimization problems has been developed during the past two decades, e.g. see (Hu, 2009; Hu & Zhang, 2009; Kennedy & Chua, 1988; Xia & Wang, 2016; Liu & Wang, 2013; Maa & Shanblatt, 1992; Nazemi, 2014; Nazemi & Nazemi, 2014; Rodriguez-Vazquez et al., 1990; Tank & Hopfield, 1986; Wu et al.,1996; Xia, 1996; Xia, 2009; Xia & Wang, 2000a; Xia & Leung, 2014; Xia & Wang, 1998; Xia & Wang, 2000b; Xia & Wang, 2000c; Xia & Wang, 2004; Xia & Wang, 2005; Xia et al.,2004; Xia et al., 2008; Xia et al.,2002; Xia et al.,2012; Xue & Bian, 2007; Yan, 2014; Yang & Cao, 2008; Yang et al.,2014; Zhang & Zhang, 2010) and references therein. To formulate an optimization problem in terms of a neural network, there exist three types of methods. One approach commonly used in developing an optimization neural network is to first convert the constrained optimization problem into an associated unconstrained optimization problem, and then design a neural network that solves the unconstrained problem with gradient methods. Another approach is to construct a set of differential equations such that their equilibrium points correspond to the desired solutions and then find an appropriate Lyapunov function such that all trajectories of the systems converges to some equilibrium points. The third is Combining the above two types of the methods. In this paper, we proposed a neural network model based on the third type. The remainder of this paper is organized as follows. In Section II, a CCR model and it's Dual is described, and a new neural network model is presented. In Section III, the global stability and convergence of the proposed neural network is analyzed. In Section IV, the performance of the proposed neural network is illustrated. Finally, the conclusion is drawn in Section V.

NEURAL NETWORK MODEL

The CCR model (input-based), can be expressed as:

$$\begin{aligned} &\min \theta \\ &s.t. \sum_{j=1}^{n} \lambda_j \mathbf{x}_{ij} \le \theta \mathbf{x}_p, i = 1,2,\dots,m \\ &\quad \sum_{j=1}^{n} \lambda_j \mathbf{y}_{rj} \ge \mathbf{y}_p, r = 1,2,\dots,s \\ &\quad \lambda_j \ge 0, j = 1,2,\dots,n \end{aligned} \tag{1}$$

Where $DMU_p=(x_p,y_p)$ is evaluating DMU,x_{ij} (i=1,2,…,m) and y_{rj}(r=1,2,…,s) are the ith input and the rth output of DMU_j (j=1,2,…,n) respectively. The Dual of 1 is:

$$\begin{aligned} &Max \sum_{j=1}^{n} \mathbf{u}_r \mathbf{y}_{rp} \\ &s.t. \sum_{r=1}^{s} \mathbf{u}_r \mathbf{y}_{rj} - \sum_{i=1}^{m} \mathbf{v}_i \mathbf{x}_{ij} \le 0, \; j=1,2,\dots,n \end{aligned} \tag{2}$$

$$\sum_{i=1}^{m} v_i x_{ip} = 1$$
$$v_i \geq 0,\ i = 1, 2, \ldots, m,\ u_r \geq 0,\ r = 1, 2, \ldots, s$$

For solving 1 and (2) Simultaneously, we define the following convex energy function to design our proposed neural network:

$$E(\lambda,\mathbf{u},\mathbf{v},\theta)=\frac{1}{2}\left\|\sum_{j=1}^{n}\mathbf{u}_r\mathbf{y}_{rp}-\theta\right\|_2^2+\frac{1}{2}\left\|\max\left\{\mathbf{A}\begin{pmatrix}\lambda\\ \theta\end{pmatrix}-\begin{pmatrix}0\\ -\mathbf{y}_p\end{pmatrix},0\right\}\right\|_2^2$$
$$+\frac{1}{2}\left\|\max\left\{\mathbf{B}\begin{pmatrix}\mathbf{u}\\ \mathbf{v}\end{pmatrix},0\right\}\right\|_2^2+\frac{1}{2}\left\|\sum_{i=1}^{m}\mathbf{v}_i\mathbf{x}_{ip}-1\right\|_2^2 \qquad (3)$$

Where

$$A=\begin{pmatrix}\mathbf{X} & -\mathbf{x}_p\\ -\mathbf{Y} & \mathbf{0}\end{pmatrix},\ B=\begin{pmatrix}\mathbf{Y}^t & -\mathbf{X}^t\end{pmatrix},$$

θ is free, $(0,0,0) \leq (\lambda,u,v) \in \mathbb{R}^{m+n+s}$, **X** and **Y** are input matrix and output matrix of all DMUs. Let $w=(\lambda,u,v,\theta)^t$,

$$C=\begin{pmatrix}X & -x_p & 0 & 0\\ -Y & 0 & 0 & 0\\ 0 & 0 & Y^t & -X^t\end{pmatrix},\quad c=\begin{pmatrix}0\\ -y_p\\ 0\\ 0\end{pmatrix}$$

$$D=\begin{pmatrix}0 & -1 & y_p^t & 0\\ 0 & 0 & 0 & -x_p^t\end{pmatrix},\quad d=\begin{pmatrix}0\\ 1\end{pmatrix}$$

So 3 can be rewritten the following form:

$$E(w)=\frac{1}{2}\left\|P_{\Omega}(\mathbf{Cw}-\mathbf{c})\right\|_2^2+\frac{1}{2}\left\|\mathbf{Dw}-\mathbf{d}\right\|_2^2 \qquad (4)$$

Where Ω and P_Ω are defined as follow:

$$\Omega=\left\{\mathbf{w}\in \mathbb{R}^{n+m+s}\,|\,\mathbf{w}_i\geq 0, i=1,2,\ldots,n+m+s\right\}$$
$$P_{\Omega}(\mathbf{u})=[P_{\Omega}(\mathbf{u}_1),P_{\Omega}(\mathbf{u}_2),\ldots,P_{\Omega}(\mathbf{u}_{n+m+s})]$$
$$P_{\Omega}(\mathbf{u}_i)=\max(\mathbf{u}_i,0),\quad i=1,2,\ldots,n+m+s$$

Using gradient method to obtain minimizer of 4, we have the following dynamic system:

$$\frac{d\mathbf{w}(t)}{dt}=-\alpha\nabla E(P_{\Omega}(\mathbf{w}))$$
$$=-\alpha\left[C^t\left(P_{\Omega}(\mathbf{C}P_{\Omega}(\mathbf{w})-\mathbf{c})\right)+D^t(\mathbf{D}P_{\Omega}(\mathbf{w})-\mathbf{d})\right] \qquad (5)$$

Where α is a scaler parameter, ∇E is the gradient of E.

The equilibrium points of system described by Eq. 4 can be applied to identify optimal solution of 1 and 2.

THEORETICAL ANALYSIS

In this section, we consider global convergence of 5 under assumption $W^* = \{w \in \Omega \mid w \text{ is minimizer of } 3\}$.

Definition 3.1. The neural network in 5 is said to be stable in the sense of Lyapunov, globally convergent and globally asymptotically stable, if the corresponding dynamic system is so (Xia et al.,2002).

Definition 3.2. A function $F:\mathbb{R}^m \to \mathbb{R}^m$ is said to be Lipschitz continuous with constant L>0 (Kinderlehrer & Stampacchia, 1980) If for each pair of points $x,y \in \mathbb{R}^m$

$$\|F(\mathbf{x})-F(\mathbf{y})\| \leq L\,\|\mathbf{x}-\mathbf{y}\|$$

The proposed neural network has the following basic properties.

Lemma 3.1. The equilibria of the neural network in 5 is equal to solution of 1 and 2. Moreover, for any initial point $w_0 = w(t_0)$, there exist a unique continuous solution w(t) for 5 over $[t_0, \infty)$.

Proof. Let $\bar{w}$ be equilibrium of 5 so $\nabla E(\bar{w})=0$, since E is a convex function then $\bar{w}$ be minimizer of E(w). Moreover minimizer of E(w) is solotion of 1 and 2.

Let $F(w)=\left[C^t\left(P_{\Omega}(\mathbf{C}P_{\Omega}(\mathbf{w})-\mathbf{c})\right)+D^t(\mathbf{D}P_{\Omega}(\mathbf{w})-\mathbf{d})\right]$

for any $\tilde{\mathbf{w}},\hat{\mathbf{w}}\in \mathbb{R}^{n+m+s}$

we have:

$$\|F(\tilde{w})-F(\hat{w})\|=$$
$$\left\|\left[C^t\left(P_{\Omega}(\mathbf{C}P_{\Omega}(\tilde{\mathbf{w}})-\mathbf{c})\right)+D^t(\mathbf{D}P_{\Omega}(\tilde{\mathbf{w}})-\mathbf{d})\right]-\left[C^t\left(P_{\Omega}(\mathbf{C}P_{\Omega}(\hat{\mathbf{w}})-\mathbf{c})\right)+D^t(\mathbf{D}P_{\Omega}(\hat{\mathbf{w}})-\mathbf{d})\right]\right\|$$
$$\leq\left\|\left[C^t\left(P_{\Omega}(\mathbf{C}P_{\Omega}(\tilde{\mathbf{w}})-\mathbf{c})\right)-C^t\left(P_{\Omega}(\mathbf{C}P_{\Omega}(\hat{\mathbf{w}})-\mathbf{c})\right)\right]+\left[D^t(\mathbf{D}P_{\Omega}(\tilde{\mathbf{w}})-\mathbf{d})-D^t(\mathbf{D}P_{\Omega}(\hat{\mathbf{w}})-\mathbf{d})\right]\right\|$$
$$\leq\left\|C^t\left(P_{\Omega}(\mathbf{C}P_{\Omega}(\tilde{\mathbf{w}})-\mathbf{c})\right)-C^t\left(P_{\Omega}(\mathbf{C}P_{\Omega}(\hat{\mathbf{w}})-\mathbf{c})\right)\right\|+\left\|D^t(\mathbf{D}P_{\Omega}(\tilde{\mathbf{w}})-\mathbf{d})-D^t(\mathbf{D}P_{\Omega}(\hat{\mathbf{w}})-\mathbf{d})\right\|$$

by property of projection operator (Kinderlehrer & Stampacchia, 1980) we have:

$$\|F(\tilde{w})-F(\hat{w})\|=$$
$$\leq\left\|C^tC\right\|\left\|P_{\Omega}(\tilde{\mathbf{w}})-P_{\Omega}(\hat{\mathbf{w}})\right\|+\left\|D^tD\right\|\left\|P_{\Omega}(\tilde{\mathbf{w}})-P_{\Omega}(\hat{\mathbf{w}})\right\|$$
$$=\left(\left\|C^tC\right\|+\left\|D^tD\right\|\right)\left\|P_{\Omega}(\tilde{\mathbf{w}})-P_{\Omega}(\hat{\mathbf{w}})\right\|$$
$$\leq\left(\left\|C^tC\right\|+\left\|D^tD\right\|\right)\left\|\tilde{\mathbf{w}}-\hat{\mathbf{w}}\right\|$$

By the existence theory of ordinary differential equations (Miller & Michel, 1982) we see that for any an initial point taken in Ω, there exists a unique and continuous solution $w(t) \subseteq \Omega$ for the systems in 5 over $[t_{0,T})$ since the function E(w) satisfies local Lipschitz conditions. Consider the following function:

$$V(w(t)) = \frac{1}{2}\|w(t) - w^*\|_2^2 \tag{6}$$

So, the time derivative of V along the trajectory of (6) is as follows:

$$\frac{dV(w(t))}{dt} = \frac{dV}{dw}\frac{dw}{dt} = -\alpha(w(t) - w^*)^T \nabla E(w(t))$$

moreover E(w) is continuously differentiable and convex on Ω and $\nabla E(w^*)$, so by property of convex function (Ortega & Rheinboldt, 1970) we have:

$$\frac{dV(w(t))}{dt} = -\lambda(w(t) - w^*)^t \nabla E(w(t)) \le 0, \forall t \in [t_0, T) \tag{7}$$

Thus:

$$\|w(t) - w^*\|_2^2 \le \beta \|w(t_0) - w^*\|_2^2, \quad \forall t \in [t_0, T)$$

Where β is a positive constant. So w(t) is bounded on $[t_{0,T})$, thus T= $+\infty$.

Theorem 3.1. The state trajectory of 5 is globally convergent to W*within a finite time. Moreover, the convergence rate of the neural network in 5 increases as α increases.

Proof: Eq. 7 yields that the system in 5 is Lyapunov stable at each equilibrium point.

On the other hand, since $\lim_{k\to\infty} V(w^k)=+\infty$ whenever the sequence $w^k \in \Omega$ and $\lim_{k\to\infty} \|w^k\| =+\infty$, by property of level sets (Ortega & Rheinboldt, 1970) we see that all the level sets of V are bounded though all level sets of E(w) are unbounded, thus $\hat{\Omega} =\{w\in\Omega \mid V(w)\le V(w_0)\}$ is bounded. Because V(w) is continuously differentiable on the compact set $\hat{\Omega}$ and $\{w(t)|t\ge t0\}\subseteq\hat{\Omega}$, it follows from the LaSalle's invariance principle (Miller & Michel, 1982) that trajectories w(t) converge to $\hat{K}$, the largest invariant subset of K= { w| dV/dt=0}. Note that if dV/dt=0 then $(w-w^*)^t\nabla E(w)=0$, by property of convex function (Ortega & Rheinboldt, 1970) we have $E(w)=E(w^*)$, so w is an equilibrium point of the system in (5). Conversely, if dV/dt=0 then $\nabla E(w)=0$ and $(w-w^*)^t\nabla E(w)=0$. So

$$\frac{dV(w(t))}{dt} = 0 \Leftrightarrow \frac{dw(t)}{dt} = 0$$

Hence we have $K=\{w\in \Omega \mid dV/dt=0\}$. Finally, let $\lim_{k\to\infty} w(tk)= \hat{u}$, then $\hat{u}\in W^*$ Therefore, for $\varepsilon>0$ there exists q>0 such that

$$\|w(t_k) - \hat{u}\| < \varepsilon, \quad k \ge q$$

Eq. (7) holds for each $\hat{u}\in W^*$, then $\|w(t)-\hat{u}\|$ is decreasing as $t\to\infty$. Thus

$$\lim_{t\to\infty} \|w(t) - \hat{u}\|_2 = 0$$

So

$$\lim_{t\to\infty} w(t) = \hat{u}$$

By (7) we have:

$$\frac{dV(w(t))}{dt} \le 0$$

Then we can result that as α increases, the convergence rate of the neural network in 5 increases. This proof is completed.

NUMERICAL EXAMPLE

In this section, we simulate the effectiveness of the proposed method through one illustrative examples. The ordinary differential equation solver engaged in ode23 in matlab 2017.

Example 1. The inputs and outputs of seven DMUs which each DMU consumes two inputs (x_1, x_2) to produce four outputs (y_1, y_2, y_3, y_4) is presented in Table 1.

The results of running our proposed model are summarized in Table 2 and Table 3. The results comparison can report that our proposed neural network model is effective to solve CCR model. Fig. 1 shows that transient behavior of the neural network of 5 in terms of θ. As can be seen from Fig.1, the proposed neural network model is globally convergent to the optimal solution.

Table 1: Data set in Example 1

DMUs	x_1	x_2	y_1	y_2	y_3	y_4
1	2.0	2.0	2.0	2.0	2.0	2.0
2	2.0	2.0	2.0	3.0	2.0	2.0
3	2.0	2.0	2.0	2.0	3.0	2.0
4	2.0	2.0	2.0	2.0	2.0	3.0
5	1.0	1.0	2.0	2.5	3.5	2.0
6	1.0	3.0	2.0	2.0	2.0	4.0
7	2.0	1.0	2.0	2.5	2.25	3.0

Table 2: Results of our model to solve CCR model 1 in Example1

DMUs	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	θ
1	0	0	0	0	1	0	0	0.5
2	0	0	0	0	1.2	0	0	0.6
3	0	0	0	0	1	0	0	0.5
4	0	0	0	0	0.25	0.25	0.5	0.75
5	0	0	0	0	1	0	0	1
6	0	0	0	0	0	1	0	1
7	0	0	0	0	0	0	1	1

Table 3: Results of our model to solve CCR model 2 in Example 1

DMUs	v_1	v_2	u_1	u_2	u_3
1	0.1393	0.3607	0.2214	0	0
2	0.3753	0.1247	0	0.2	0
3	0.1909	0.3091	0.25	0	0
4	0.25	0.25	0	0	0
5	0.3398	0.6602	0.2344	0	0
6	0.2877	0.2374	0	0	0.0033
7	0.0369	0.9263	0	0.2285	0.0144

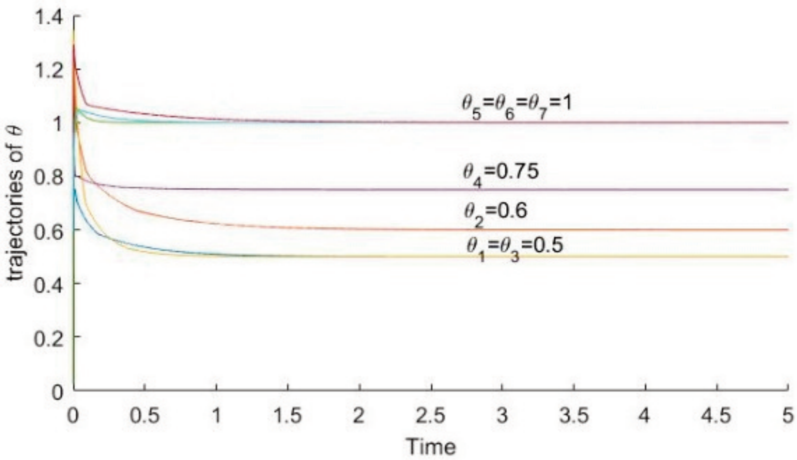

Fig. 1. Transient behavior of (5) in terms of trajectories for evaluating in example 1

CONCLUSIONS

In this paper, a recurrent neural network introduced to solve CCR model in DEA. The proposed model is a one-layer neural network. In the proposed model is used projection operator for non-negative variables thus respect to the similar models, the number of neurons reduces so proposed model has lower complexity. It is shown here that the proposed neural network is stable in the sense of Lyapunov and globally convergent to the optimal solutions. Finally, the example is provided to show the effectiveness of the proposed neural network.

REFERENCES

Charnes, A., Cooper, W. W., & Rhodes, E. (1979). Measuring the efficiency of decision-making units. *European journal of operational research, 3*(4), 339.

Hu, X. (2009). Applications of the general projection neural network in solving extended linear-quadratic programming problems with linear constraints. N*eurocomputing, 72* (4), 1131-1137.

Hu, X., Zhang, B. (2009). A New Recurrent Neural Network for Solving Convex Quadratic Programming Problems with an Application to the-Winners-Take-All Problem. Neural Networks, I*EEE Transactions on, 20*(4), 654-664.

Kennedy, M. P., Chua, L. O. (1988). Neural networks for nonlinear programming. *Circuits and Systems, IEEE Transactions on, 35*(5), 554-562.

Kinderlehrer, D., & Stampacchia, G. (1980). An introduction to variational inequalities and their applications (Vol. 31). Siam.

Liu, Q., Wang, J. (2013). A one-layer projection neural network for nonsmooth optimization subject to linear equalities and bound constraints. *Neural Networks and Learning Systems, IEEE Transactions on, 24*(5), 812-824.

Maa, C. Y., Shanblatt, M. A. (1992). Linear and quadratic programming neural network analysis. *Neural Networks, IEEE Transactions on, 3*(4), 580-594.

Miller, R. K., & Michel, A. N. (1982). Ordinary differential equations. Academic Press.

Nazemi, A. (2014). A neural network model for solving convex quadratic programming problems with some applications. *Engineering Applications of Artificial Intelligence, 32*, 54-62.

Nazemi, A., Nazemi, M. (2014). A Gradient-Based Neural Network Method for Solving Strictly Convex Quadratic Programming Problems. Cognitive Computation. 1-12.

Ortega, J. M., & Rheinboldt, W. C. (1970). Iterative solution of nonlinear equations in several variables (Vol. 30). Siam.

Rodriguez-Vazquez, A., Dominguez-Castro, R., Rueda, A., Huertas, J. L., Sanchez-Sinencio, E. (1990). Nonlinear switched capacitor neural networks for optimization problems.*Circuits and Systems, IEEE Transactions on, 37*(3), 384-398.

Tank, D., Hopfield, J. J. (1986). Simple 'neural' optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit. *Circuits and Systems*, *IEEE Transactions on, 33*(5), 533-541.

Wu, X. Y., Xia, Y. S., Li, J., Chen, W. K. (1996). A high-performance neural network for solving linear and quadratic programming problems. *Neural Networks, IEEE Transactions on, 7*(3), 643-651.

Xia, Y. (1996). A new neural network for solving linear and quadratic programming problems. Neural Networks, *IEEE Transactions on, 7*(6), 1544-1548.

Xia, Y. (2009). A Compact Cooperative Recurrent Neural Network for Computing General Constrained Norm Estimators. Signal Processing, IEEE Transactions on, 57(9),3693-3697.

Xia, Y. S., Wang, J. (2000a). On the stability of globally projected dynamical systems. *Journal of Optimization Theory and Applications, 106*(1), 129-150.

Xia, Y., & Wang, J. (2016). A bi-projection neural network for solving constrained quadratic optimization problems. *IEEE transactions on neural networks and learning systems, 27*(2), 214-224.

Xia, Y., Feng, G., Wang, J. (2004). A recurrent neural network with exponential convergence for solving convex quadratic program and related linear piecewise equations. *Neural Networks, 17*(7), 1003-1015.

Xia, Y., Feng, G., Wang, J. (2008). A novel recurrent neural network for solving nonlinear optimization problems with inequality constraints. Neural Networks, IEEE Transactions on,

19(8), 1340-1353.
Xia, Y., Leung, H. (2014). A Fast Learning Algorithm for Blind Data Fusion Using a Novel-Norm Estimation. *Sensors Journal, IEEE, 14*(3), 666-672.
Xia, Y., Leung, H., Wang, J. (2002). A projection neural network and its application to constrained optimization problems. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on, 49*(4), 447-458.
Xia, Y., Sun, C., Zheng, W. X. (2012). Discrete-time neural network for fast solving large linear estimation problems and its application to image restoration. *Neural Networks and Learning Systems, IEEE Transactions on, 23*(5), 812-820.
Xia, Y., Wang, J. (1998). A general methodology for designing globally convergent optimization neural networks. *Neural Networks, IEEE Transactions on, 9*(6), 1331-1343.
Xia, Y., Wang, J. (2000b). A recurrent neural network for solving linear projection equations. *Neural Networks, 13*(3), 337-350.
Xia, Y., Wang, J. (2000c). Global exponential stability of recurrent neural networks for solving optimization and related problems. *Neural Networks, IEEE Transactions on,11*(4), 1017-1022.
Xia, Y., Wang, J. (2004). A general projection neural network for solving monotone variational inequalities and related optimization problems. *Neural Networks, IEEE Transactions on, 15*(2), 318-328.
Xia, Y., Wang, J. (2005). A recurrent neural network for solving nonlinear convex programs subject to linear constraints. *Neural Networks, IEEE Transactions on, 16*(2),379-386.
Xue X, Bian W. (2007). A project neural network for solving degenerate convex quadratic program. Neurocomputing.70:244959. 46.
Yan, Y. (2014). A New Nonlinear Neural Network for Solving QP Problems. In Advances in Neural Networks ISNN 2014 (pp. 347-357). Springer International Publishing.
Yang Y, Cao J. (2008). A feedback neural network for solving convex constraint optimization problems. Appl Math Comput. 201: 34050. 47.
Yang, Y., Cao, J., Xu, X., Hu, M., & Gao, Y. (2014). A new neural network for solving quadratic programming problems with equality and inequality constraints. *Mathematics and Computers in Simulation, 101*, 103-112.
Zhang, J., & Zhang, L. (2010). An augmented Lagrangian method for a class of inverse quadratic programming problems. *Applied Mathematics and Optimization, 61*(1), 57.