Available Online at http://jnrm.srbiau.ac.ir
Vol.1, No.3, Autumn 2015

Journal of New Researches in Mathematics

Science and Research Branch (IAU)

A Recurrent Neural Network to Identify Efficient Decision Making Units in Data Envelopment Analysis

A. Ghomashi[a*], G. R. Jahanshahloo[a**], F. Hosseinzadeh Lotfi[a***]

[(a)] *Department of Mathematics, Science and Research Branch, Islamic Azad University, Tehran, Iran*

Received Summer 2015, Accepted Autumn 2015

Abstract

In this paper we present a recurrent neural network model to recognize efficient Decision Making Units(DMUs) in Data Envelopment Analysis(DEA). The proposed neural network model is derived from an unconstrained minimization problem. In theoretical aspect, it is shown that the proposed neural network is stable in the sense of lyapunov and globally convergent. The proposed model has a single-layer structure. Simulation shows that the proposed model is effective to identify efficient DMUs in DEA.

Keywords: Recurrent neural network, Gradient method, Data envelopment analysis, Efficient DMU, Stability, Global convergence.

*. a_ gh_ l@yahoo.com
**. jahanshahloomath@gamil.com
***. Corresponding author: farhad@hosseinzadeh.ir

1. 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[3]. Banker et al in 1984 developed a variable returns to scale version of the CCR model that was called BCC model [1]. DEA successfully divides DMUs into two categories: efficient DMUs and inefficient DMUs. DEA does this by assigning a relative efficiency score to each DMU such that the DMUs in efficient category have identical relative efficiencies equal to one and the rest have the relative efficiencies between zero and one.

Linear programming is needed to recognize efficient and inefficient DMUs in CCR and BCC models. The dimension and denseness of the structure of linear programming increases as the numbers of DMUs increases, in this case, the numerical methods become less affected for solving corresponding linear programming. One promising approach to solve CCR and BCC models with hight dimension and denseness of structure is to employ the artificial neural networks based on circuit implementation [7]. The neural networks are computing systems composed of a number of highly interconnected simple information processing units, and thus can usually solve optimization problems in execution times at the orders of magnitude much faster than most popular optimization algorithms for general-purpose digital[9]. One approach commonly used in developing an optimization neural network is to first convert the constrained optimization problem into an associated unconstrained optimization problem, then establish an energy function and a dynamic system which is a representation of an neural network model. The dynamic system is normally in the form of first order ordinary differential equations.

In this paper, based on properties of Additive form in BCC, we establish a convex energy function which its minimum points can applied to identify efficient DMUs in BCC and using gradient method we construct a first order ordinary differential equation which its equilibrium points correspond to the minimum points of proposed energy function, in this case, the obtained dynamic system can be realized by a recurrent neural network model which one-layer structure. The proposed neural network model is proved to has globally convergent. Simulation results present effectiveness of the proposed model to identify efficient DMUs in BCC model. This paper is divided into six sections. In next section preliminary information is introduced to facilitate later discussions. In section III, we proposed a neural network model to identify efficient DMUs in BCC models. we analyze stability condition and global convergence in section IV. In section V, some simulation examples are discussed. Section VI gives the conclusions of this paper.

2. Preliminaries

2.1. DEA Preliminaries

Suppose there are n DMUs with m inputs and s outputs. The input and output vectors of $DMU_j (j=1,\dots,n)$ are $X_j=(x_{1j},\dots,x_{mj})^t$, $Y_j=(y_{1j},\dots,y_{sj})^t$,respectively, where

$X_j \geq 0, X_j \neq 0, Y_j \geq 0, Y_j \neq 0$.

The PPS is represented as

$$T=\{(X,Y)\in R^{m+s} \mid Y \text{ can be produced from } X\}$$

Charnes et al [3] have concluded the following PPS for CCR model:

$$T_{CCR}=\{(X,Y)\in R_+^{m+s} \mid X \geq \sum_{j=1}^{n}\lambda_j X_j,\ Y \leq \sum_{j=1}^{n}\lambda_j Y_j,\ \lambda_j \geq 0 \quad j=1,\dots,n\}$$

The CCR model is used for an efficiency measure of DMUs under the condition of constant returns to scale (CRS)[3]. The envelopment model of CCR in input-oriented to evaluate the efficiency of a specific $DMU_p (p\in\{1,\dots,n\})$ under T_{CCR} is as follows[5]:

$$\theta^*_{CCR}=Min\ \{\theta \mid \sum_{j=1}^{n}\lambda_j X_j \leq \theta X_p,\ \sum_{j=1}^{n}\lambda_j Y_j \geq Y_p, \lambda_j \geq 0 \quad j=1,\dots,n\}$$

Banker et al[1] omitted the ray unboundedness postulate from the CCR postulates and inferred the following PPS for BCC model:

$$T_{BCC}=\{(X,Y)\in R_+^{m+s} \mid X \geq \sum_{j=1}^{n}\lambda_j X_j,\ Y \leq \sum_{j=1}^{n}\lambda_j Y_j, \sum_{j=1}^{n}\lambda_j=1, \lambda_j \geq 0 \quad j=1,\dots,n\}$$

The BCC model is used for an efficiency measure under the condition of variant returns to scale(VRS)[1]. The envelopment model of BCC in input-oriented to evaluate the efficiency of a specific $DMU_p (p\in\{1,\dots,n\})$ under T_{BCC} is as follows[5]:

$$\theta^*_{BCC} = Min\{\theta \mid \sum_{j=1}^{n}\lambda_j X_j \le \theta X_p, \sum_{j=1}^{n}\lambda_j Y_j \ge Y_p, \sum_{j=1}^{n}\lambda_j = 1, \lambda_j \ge 0 \; j = 1,\ldots,n\} \quad (1)$$

Envelopment model of BCC model in output-oriented is expressed as[5]

$$\varphi^*_{BCC} = Max\{\varphi \mid \sum_{j=1}^{n}\lambda_j X_j \le X_p, \sum_{j=1}^{n}\lambda_j Y_j \ge \varphi Y_p, \sum_{j=1}^{n}\lambda_j = 1, \lambda_j \ge 0 \; j = 1,\ldots,n\} \quad (2)$$

Combination both orientations in a single model is called the Additive model which is as follows:

$$z^*_{ADD} = Max\{\sum_{i=1}^{m} S_i^- + \sum_{r=1}^{s} S_r^+ \mid \sum_{j=1}^{n}\lambda_j X_j + S^- = X_p, \sum_{j=1}^{n}\lambda_j Y_j - S^+ = Y_p, \sum_{j=1}^{n}\lambda_j = 1, \lambda_j \ge 0 \; j = 1,\ldots,n\} \quad (3)$$

Where $S^- \in R^m$ *and* $S^+ \in R^s$. The dual problem to the above can be expressed as follows:

$$w^*_{ADD} = Min\{V^t X_p - U^t Y_p + u_0 \mid U^t X_j - V^t Y_j - u_0 \le 0, j = 1,\ldots,n, V_i \ge 1, i = 1,\ldots,m, U_r \ge 1, r = 1,\ldots,s, u_0 \; free\} \quad (4)$$

Definition 2.1.

DMU_p is ADD-efficient if and only if $w^*_{ADD} = 0$ [5].

Definition 2.2.

if DMU_p is ADD-efficient then DMU_p is called BCC-efficient otherwise DMU_p is called BCC-inefficient[5].

2.2. Functions

Definition 2.3.

A function $F: R^m \to R^m$ is said to be Lipschitz continuous with constant $L > 0$ [6] *if for each pair of points $x, y \in R^m$*

$$\|F(x) - F(y)\| \le L\|x - y\|$$

Definition 2.4.

Let $X = \{x \in R^m \mid l_i \le x \le u_i, \forall i \in N - \{1,\ldots,m\}\}$, $P_X: R^m \to X$ is called a projection operator and defined the following form[6]*:*

$$P_X(x) = arg \;\; min\, \|x - y\|, y \in X$$

Indeed $P_X(x)$ is a projection of vector x to set X. Since X is a box set, the projection operator P_X can be stated by

$$P_X(x) = [P_X(x_1),\ldots,P_X(x_m)]'$$

Where $P_X(x_i)$ is a piecewise function. For $i \in \{1,\ldots,m\} - N, P_X(x_i) = x_i$ and for $i \in N$, $P_X(x_i)$ is defined the following form[6]:

$$P_X(x_i) = \begin{cases} l_i, & x_i < l_i \\ x_i, & l_i \le x_i \le u_i \\ u_i, & x_i > u_i \end{cases}$$

Lemma 2.1.

Let $X \in R^n$ be a closed convex set. Then

$$\|P_X(x) - P_X(y)\| \le \|x - y\| \qquad \forall x, y \in R^n$$

where $P_X(x)$ is a projection operator on X.

Proof: See[6].

Theorem 2.1. *Let S be a nonempty open convex set in R^n and let $f : S \to R$ be a differentiable function on S. Then f is convex if and only if for each $x_1, x_2 \in S$*

$$(\nabla f(x_2) - \nabla f(x_1))^T (x_2 - x_1) \ge 0$$

Proof. See[2].

2.3. Differential Equation Preliminaries

Definition 2.5.

Take the following dynamical system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t)), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \in \boldsymbol{R}^n \qquad (5)$$

Where $f : R^n \to R^n$. x^* is called an equilibrium point of (5) if [10]

$$\boldsymbol{f}(\boldsymbol{x}^*) = 0.$$

Theorem 2.2.

Assume that f in (5) is a continuous mapping, then for arbitrary $t_0 \ge 0$ and $x_0 \in R^n$ there exists a local solution $x(t)$ to (5) where $t \in [t_0, \tau]$ for some $\tau > t_0$. Furthermore if f is locally Lipschitzian continuous at x_0 then the solution is unique, and if f is Lipschitzian continuous in R^n then τ can be extended to $+\infty$ [10].

Theorem 2.3. *Let x^* be an equilibrium point of (5) and $X \subset R^n$ be an open neighborhood of x^*, if $V : R^n \to R$ is a continuously differentiable function over X and V satisfies in the following conditions:*

$$\boldsymbol{V}(\boldsymbol{x}^*) = 0, \quad \frac{\boldsymbol{dV}(\boldsymbol{x}^*)}{\boldsymbol{dt}} = 0$$

$$\frac{dV(x(t))}{dt} \le 0, \ V(x) > 0, \qquad \forall x \in X - \{x^*\}$$

$$\|x\| \to \infty \Rightarrow \|V(x)\| \to \infty$$

Then x^* is a lyapunov stable equilibrium and the solution always exist globally[10].

Lemma 2.2.

Let u and v be real-valued non negative continuous functions with domain $\{t \mid t \ge t_0\}$, let $a(t) = a_0(\|t - t_0\|)$ where a_0 is monotone increasing function. if for $t \ge t_0$

$$\boldsymbol{u}(t) \le \boldsymbol{a}(t) + \int_{t_0}^{t} \boldsymbol{u}(s)\boldsymbol{v}(s)\boldsymbol{ds}$$

hence

$$\boldsymbol{u}(t) \le \boldsymbol{a}(t)\boldsymbol{e}^{\int_{t_0}^{t} v(s)ds}$$

Proof: See[8].

3. Neural Network Model

Based upon definition 2.1 and 2.2 DMU_p is called BCC-efficient if the following system of linear inequality has a solution:

$$\begin{aligned}&U'Y_p - V'X_p - u_0 = 0\\&U'Y_j - V'X_j - u_0 \le 0, \qquad j = 1,\dots,n, \quad (6)\\&V_i \ge 1, \qquad i = 1,\dots,m\\&U_r \ge 1, \qquad r = 1,\dots,s\end{aligned}$$

Let

$$A = \begin{pmatrix} Y_1,\cdots, & Y_n \\ -X_1,\cdots, & -X_n \\ -1,\cdots, & -1 \end{pmatrix},\ B = \begin{pmatrix} Y_p \\ -X_p \\ -1 \end{pmatrix},\ w = \begin{pmatrix} U \\ V \\ u_0 \end{pmatrix}$$

So (6) can be rewritten as the following form:

$$\begin{aligned}&A^t w \le 0\\&B^t w = 0 \qquad (7)\\&w_i \ge 1,\ i = 1,\dots,m+s\\&w_{m+s+1}\ \text{free}\end{aligned}$$

Now we define the following convex energy function to design our proposed neural network model to solve (7):

$$E(w) = \frac{1}{2}\|B^t w\|_2^2 + \frac{1}{2}\|\mathrm{Max}\{A^t w, 0\}\|_2^2, \quad w \in \Omega \quad (8)$$

Where

$\Omega = \{x \in R^{m+s+1} \mid x_i \ge 1, i = 1,\dots,m+s\}$

$E(w) \ge 0$ for all $w \in R^{m+s+1}$,so if w^* be minimizer of (8)and $E(w^*) = 0$ then (6) has a solution, hence based on definition 2.1 and 2.2 , DMU_p is BCC-efficient, otherwise DMU_p is inefficient in BCC.

We use gradient method to obtain minimizer of (8) then we have the following dynamic system:

$$\begin{aligned}\frac{dw(t)}{dt} &= -\lambda \nabla E(P_\Omega(w)) \qquad (9)\\&= -\lambda[B(B^t P_\Omega(w)) + A P_{\bar\Omega}(A^t P_\Omega(w))]\end{aligned}$$

Where $\lambda > 0$ is a scaler parameter, ∇E is the gradient of E, Ω is defined in (8), $\bar\Omega = \{x \in R^n \mid x_i \ge 0, i = 1,\dots,n\}$, P_Ω and $P_{\bar\Omega}$ are projection operator as follows:

$$P_\Omega(u) = [P_\Omega(u_1),\dots,P_\Omega(u_{m+s}), u_{m+s+1}]^t$$
$$P_{\bar\Omega}(v) = [P_\Omega(v_1),\dots,P_\Omega(v_n)^+]^t$$

where $P_{\bar\Omega}(v_i) = max(0, v_i)$ and

$$P_\Omega(u_i) = \begin{cases} 1, & u_i < 1 \\ u_i, & 1 \le u_i \end{cases}$$

Based on definition 2.5 and foregoing statement, the equilibrium points of system described by Eq.(9) can be applied to identify efficient DMU in BCC.

We see from Fig.1 that the system described by Eq.(9) can be realized by a recurrent neural network with one-layer structure. The operator P_Ω *and* $P_{\bar\Omega}$ may be implemented by using a piecewise activation function[4].

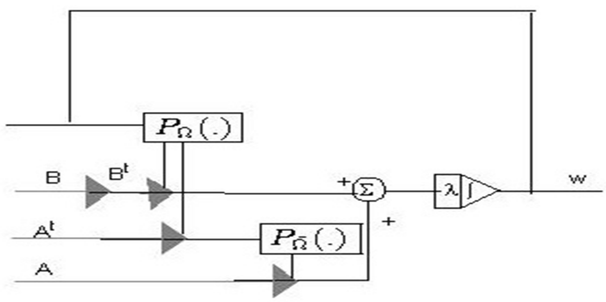


Fig. 1. Block diagram of the recurrent neural network in (9).

As shown in Fig.1, the proposed neural network can be implemented by using a simple hardware only without analog multipliers for variables or the penalty parameter. The proposed neural network consists of $m+s+1$ integrators, m+s+n piecewise activation function and m+s+1 summers.

4. Global Stability

In this section, we consider global convergence of (9) under assumption $W^* = \{w \in \Omega \mid w \text{ is minimizer of } (8)\}$. We first give one definition for latter discussions.

Definition 4.1.

The neural network in (9) is said to be stable in the sense of Lyapunov, globally convergent and globally asymptotically stable, if the corresponding dynamic system is so [9].

The proposed neural network has the following basic property.

Lemma 4.1.

The equilibria of the neural network in (9) is equal to W^*. Moreover, for any initial point $w_0 = w(t_0)$, there exist a unique continuous solution $w(t)$ for (9) over $[t_0, \infty)$.

Proof. Let $\overline{w}$ be equilibrium of (9) so $\nabla E(\overline{w}) = 0$, since E is a convex function then $\overline{w}$ be minimizer of $E(w)$. Moreover $E(w)$ always has minimum so $W^* \neq \varnothing$.

Let $\mathrm{F}(\boldsymbol{w}) = [\boldsymbol{B}(\boldsymbol{B}^t \mathrm{P}_\Omega(\boldsymbol{w})) + \boldsymbol{A}\mathrm{P}_{\overline{\Omega}}(\boldsymbol{A}^t \mathrm{P}_\Omega(\boldsymbol{w}))]$,

for any $u, v \in R^{m+s+1}$ we have:

$$\begin{aligned}
&\|F(u) - F(v)\| \\
&= \left\| \begin{matrix} \left[B^t\left(BP_\Omega(u)\right) + AP_{\overline{\Omega}}\left(A^tP_\Omega(u)\right)\right] \\ -\left[B\left(B^tP_\Omega(v)\right) + AP_{\overline{\Omega}}\left(A^tP_\Omega(v)\right)\right] \end{matrix} \right\| \\
&\le \|BB^tP_\Omega(u) - BB^tP_\Omega(v)\| \\
&+ \left\|AP_{\overline{\Omega}}\left(A^tP_\Omega(u)\right) - AP_{\overline{\Omega}}\left(A^tP_\Omega(v)\right)\right\|
\end{aligned}$$

by lemma 2.1 we have

$$\begin{aligned}
&\|F(u) - F(v)\| \\
&\qquad \le \|BB^t\|\|P_\Omega(u) - P_\Omega(v)\| \\
&\qquad + \|AA^tP_\Omega(u) - AA^tP_\Omega(v)\| \\
&\qquad \le \|BB^t\|\|P_\Omega(u) - P_\Omega(v)\| \\
&\qquad + \|AA^t\|\|P_\Omega(u) - P_\Omega(v)\| \\
&\qquad \le (\|BB^t\| \\
&\qquad + \|AA^t\|)\|u - v\|
\end{aligned}$$

hence $F(w)$ is lipschitz continuous in R^{m+s+1}, so based on theorem 2.2 for any initial point $w_0 = w(t_0)$ there exist a unique continuous solution $w(t)$ for (9) over $[t_0, T)$ which $[t_0, T)$ be its maximal interval of existence. Now we present that $T = \infty$. for this, let $w^* \in W^*$, then we have:

$$\begin{aligned}
&\|F(w)\| = \|F(w) - F(w^*)\| \\
&= \left\| \begin{matrix} \left[\|BB^t\|P_\Omega(w) - AP_{\overline{\Omega}}\left(A^tP_\Omega(w)\right)\right] \\ -\left[BB^tP_\Omega(w^*) - AP_{\overline{\Omega}}\left(A^tP_\Omega(w^*)\right)\right] \end{matrix} \right\| \\
&\le \|BB^tP_\Omega(w) - BB^tP_\Omega(w^*)\| \\
&+ \left\|AP_{\overline{\Omega}}\left(A^tP_\Omega(w)\right) - AP_{\overline{\Omega}}\left(A^tP_\Omega(w^*)\right)\right\| \\
&\le (\|BB^t\| + \|AA^t\|)\|w\| \\
&+ (\|BB^t\| + \|AA^t\|)\|w^*\|
\end{aligned}$$

using foregoing inequality and (9) we have

$$\|\boldsymbol{w}(\mathrm{t})\| \leq \|\boldsymbol{w}(\mathrm{t}_0)\| + \int_{t_0}^{t} \lambda \,\|\mathrm{F}(\boldsymbol{w}(\mathrm{s}))\| \mathrm{ds}$$

$$\leq (\|\boldsymbol{w}(\mathrm{t}_0)\| + \overline{\mathrm{q}}(\mathrm{t}-\mathrm{t}_0)) + \mathrm{q}\int_{t_0}^{t} \|\boldsymbol{w}(\boldsymbol{s})\| \mathrm{ds}$$

where $\overline{\mathrm{q}} = \lambda(\|\boldsymbol{BB'}\| + \|\boldsymbol{AA'}\|)\|\boldsymbol{w}^*\|$ and $\mathrm{q} = \lambda(\|\boldsymbol{BB'}\| + \|\boldsymbol{AA'}\|)$. Regarding lemma 2.2 we have

$$\|\boldsymbol{w}(\mathrm{t})\| \leq (\|\boldsymbol{w}(\mathrm{t}_0)\| + \overline{\mathrm{q}}(\mathrm{t}-\mathrm{t}_0))\mathrm{e}^{\mathrm{q}(\mathrm{t}-\mathrm{t}_0)}, \quad \mathrm{t} \in [\mathrm{t}_0, \mathrm{T}]$$

so $w(t)$ is bounded on $[t_0, T)$ then $T = \infty$, this proof is completed.

We now affirm our basic result as follows.

Theorem 4.1.

The state trajectory of (9) is globally convergent to W^* within a finite time when the parameter λ is large enough. Moreover, the convergence rate of the neural network in (9) increases as λ increases.

Proof. Let $\boldsymbol{w}^* \in \boldsymbol{W}^*$ and $w(t)$ be the trajectory of the state equation defined in(9) with any given initial point $w(t_0) = w_0$. Consider the following lyapunov function:

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

So, the time derivative of V along the trajectory of (9) is as follows

$$\frac{\mathrm{d}\boldsymbol{V}(\boldsymbol{w}(\mathrm{t}))}{\mathrm{dt}} = \frac{\mathrm{d}\boldsymbol{V}}{\mathrm{d}\boldsymbol{w}}\frac{\mathrm{d}\boldsymbol{w}}{\mathrm{dt}} = -\lambda(\boldsymbol{w}(\mathrm{t}) - \boldsymbol{w}^*)^{\mathrm{T}}\nabla\mathrm{E}(\boldsymbol{w}(\mathrm{t})) \quad (10)$$

moreover $E(w)$ is continuously differentiable and convex on Ω and $\nabla\mathrm{E}(\boldsymbol{w}^*) = 0$, so by theorem 2.1 we have

$$\frac{dV(w(t))}{dt} = -\lambda(w(t) - w^*)^t\,\nabla E(w(t)) \leq 0$$
$$\forall t \in [t_0, +\infty] \quad (11)$$

(1) and (9) yield

$$\frac{\mathrm{dV}(\boldsymbol{w}(\mathrm{t}))}{\mathrm{dt}} = 0 \Leftrightarrow \frac{\mathrm{d}\boldsymbol{w}(\mathrm{t})}{\mathrm{dt}} = 0 \quad (12)$$

using (10), (12), we have

$$\frac{dV(w(t))}{dt} = -\lambda(w(t) - w^*)^t\,\nabla E(w(t)) \leq 0$$
$$w(t) \notin w^* \quad (12)$$

Moreover if $\|w\| \to \infty$ then $\|V(w)\| \to \infty$ so by applying the theorem 2.3, we get result that the proposed neural network is globally convergent to the solution set of (7).

Now, we show that the convergence time is finite. For this, suppose $w(t_0) = w_0 \in \Omega$ is not equilibrium point. we define

$$g(w(t)) = (w(t) - w^*)^t\,\nabla E(w(t)),$$
$$\forall t \geq t_0 \quad (14)$$

from (11), we know that $g(w(t)) \geq 0$ for all $t \geq t_0$, since $w(t_0)$ is not equilibrium point then $g(w(t)) > 0$. g(w(t)) is continuous, so there exist $\alpha > 0$ and $\kappa > 0$ such that

$$\mathrm{g}(\boldsymbol{w}(\mathrm{t})) > 0 \qquad \forall \mathrm{t} \geq \mathrm{t}_0 \quad (15)$$

using (10) and (14) we have:

$$\frac{\mathrm{dV}(\boldsymbol{w}(\mathrm{t}))}{\mathrm{dt}} = -\lambda\,\mathrm{g}(\boldsymbol{w}(\mathrm{t})) \quad (16)$$

hence

$$\int_{t_0}^{t}\frac{dV(\boldsymbol{w}(z))}{dz}dz=-\lambda\int_{t_0}^{t}g(\boldsymbol{w}(z))dz,\ \forall t\in[t_0,t_0+\kappa)$$

thus by (15) we have:

$$V(w(t))=V(w(t_0))$$
$$g(w(z))dz\le$$
$$-\lambda\int_{t_0}^{t_0+k}V(w(t_0))-\lambda\int_{t_0}^{t_0+k}\alpha dz$$
$$=V(w(t_0))-\lambda\alpha k$$

so

$$V(\boldsymbol{w}(t))\le V(\boldsymbol{w}(t_0))-\lambda\alpha\kappa$$

by taking $\lambda=\dfrac{V(\boldsymbol{w}(t_0))}{\alpha\kappa}$ we have

$$V(\boldsymbol{w}(t))=0,\qquad \forall\ t\ge t_0+k$$

thus

$$\boldsymbol{w}(t)=\boldsymbol{w}^*,\qquad \forall\ t\ge t_0+\kappa$$

that is, the state trajectory of proposed neural network is globally convergent to W^* within a finite time. By (12) we have

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

then we can result that as λ increases, the convergence rate of the neural network in (9) increases. This proof is completed.

5. Illustrative Example

In this section, we demonstrate the effectiveness and performance of the propose neural network model with two illustrative examples. The ordinary differential equation solver engaged in ode23 in matlab 2012.

Example 1. *Fourteen DMUs were evaluated in terms of three inputs* (x_1,x_2,x_3) *and three output* (y_1,y_2,y_3) *that defined in table 1.*

The result of proposed model and additive model(4) to identify BCC-efficient DMUs is provided in table 2. As can be seen from table 2, the result of proposed model similar to additive model, so it is effective approach to identify BCC-efficient DMUs.

Table 1: Data set in Example 1.

DMUs	x_1	x_2	x_3	y_1	y_2	y_3
1	225935.0	405.0	1000.0	178285.0	4967.0	3388.0
2	102200.0	179.0	575.0	75526.0	3808.0	2083.0
3	94900.0	135.0	388.0	63868.0	5435.0	1246.0
4	51100.0	95.0	263.0	31835.0	1222.0	559.0
5	43800.0	43.0	186.0	19360.0	1112.0	373.0
6	31755.0	36.0	116.0	23372.0	2095.0	275.0
7	31390.0	28.0	99.0	17798.0	462.0	277.0
8	29930.0	31.0	159.0	14067.0	794.0	184.0
9	23360.0	39.0	153.0	18127.0	1143.0	269.0
10	20440.0	31.0	94.0	14860.0	949.0	157.0
11	10585.0	15.0	67.0	4498.0	148.0	29.0
12	8030.0	15.0	62.0	8311.0	335.0	70.0
13	11680.0	26.0	110.0	9449.0	435.0	161.0
14	11315.0	20.0	92.0	4797.0	71.0	53.0

Table 2: Results of comparison of our model and BCC model in Example 1.

DMUs	$E(w)$	Result of proposed model	w^*_{ADD}	Result of Additive model
1	0.000	Efficient	0.000	Efficient
2	0.000	Efficient	0.000	Efficient
3	0.000	Efficient	0.000	Efficient
4	0.2851	Inefficient	0.5095	Inefficient
5	0.1545	Inefficient	0.2890	Inefficient
6	0.000	Efficient	0.000	Efficient
7	0.000	Efficient	0.000	Efficient
8	0.1535	Inefficient	0.2796	Inefficient
9	0.0465	Inefficient	0.0808	Inefficient
10	0.0489	Inefficient	0.0860	Inefficient
11	0.0421	Inefficient	0.0842	Inefficient
12	0.000	Efficient	0.000	Efficient
13	0.000	Efficient	0.000	Efficient
14	0.0882	Inefficient	0.1659	Inefficient

Example 2. *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 3.*

Table 3: Data set in Example 2

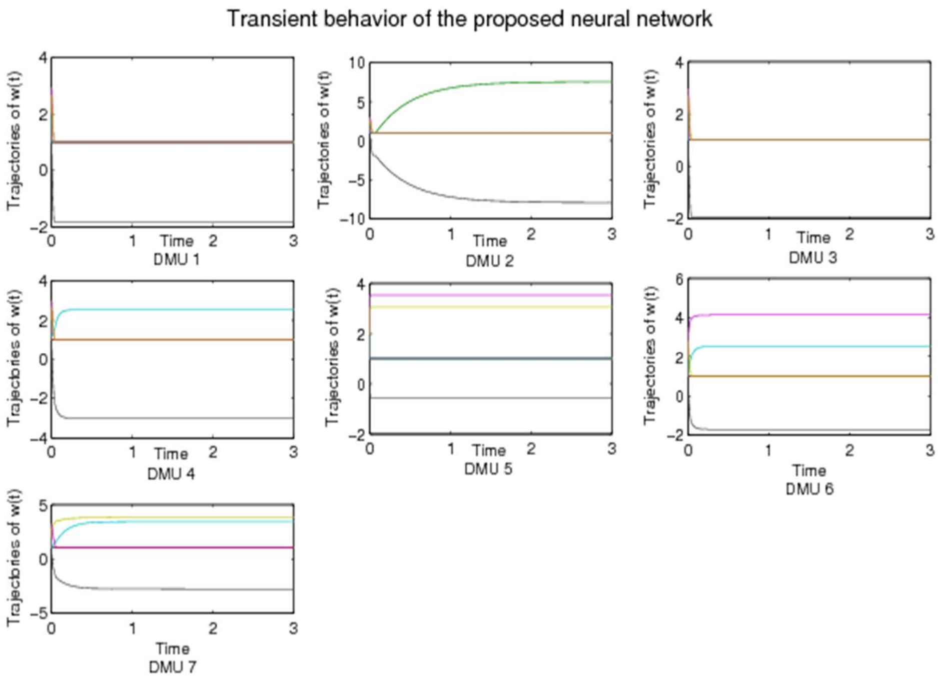
DMUs	x_1	x_2	y_1	y_2	y_3	y_4
1	2.0	2.0	2.0	2.0	2.0	2.0
2	2.0	2.0	2.0	3.0	2.0	2.0
3	2.0	2.0	2.0	2.0	3.0	2.0
4	2.0	2.0	2.0	2.0	2.0	3.0
5	1.0	1.0	2.0	2.5	3.5	2.0
6	1.0	3.0	2.0	2.0	2.0	4.0
7	2.0	1.0	2.0	2.5	2.25	3.0

The results of running additive model and our proposed model are summarized in table 4. The results comparison can report that our proposed neural network model is effective to identify efficient and inefficient DMUs in T_{BCC}. Fig. 2 shows that transient behavior of the neural network of (9) in terms of $w(t)$. As can be seen from Fig. 2, the proposed neural network model is globally convergent to the optimal solution.

Table 4: Results of comparison of our model and additive model in Example 2

DMUs	$E(w)$	Result of proposed model	w^*_{ADD}	Result of additive model
1	0.7162	Inefficient	1.4286	Inefficient
2	0.000	Efficient	0.000	Efficient
3	0.5714	Inefficient	1.1429	Inefficient
4	0.4524	Inefficient	0.7976	Inefficient
5	0.000	Efficient	0.000	Efficient
6	0.0000	Efficient	1.000	Efficient
7	0.0000	Efficient	1.000	Efficient

Fig. 2. Transient behavior of the neural network of (9) in terms of $w(t)$ in example 2



6. Conclusion

In this paper, a recurrent neural network introduced to identify efficient DMUs in T_{BBC}. The proposed model is a one-layer neural network. It is shown here that the proposed neural network is stable in the sense of Lyapunov and globally convergent to the optimal solutions. Finally, examples are provided to show the effectiveness of the proposed neural network.

References

[1] Banker, R. D., Charnes, A., & Cooper, W. W. (1984). Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Management science*, *30*(9), 1078-1092.

[2] Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (2013). *Nonlinear programming: theory and algorithms*. John Wiley & Sons.

[3] Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. *European journal of operational research*, *2*(6), 429-444.

[4] Cochocki, A., & Unbehauen, R. (1993). *Neural networks for optimization and signal processing*. John Wiley & Sons, Inc..

[5] Cooper, W. W., Seiford, L. M., & Tone, K. (2006). *Introduction to data envelopment analysis and its uses: with DEA-solver software and references*. Springer Science & Business Media.

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

[7] Xia, Y., & Wang, J. (1998). A general methodology for designing globally convergent optimization neural networks. *Neural Networks, IEEE Transactions on*, *9*(6), 1331-1343.

[8] Xia, Y., & Wang, J. (2000). A recurrent neural network for solving linear projection equations. *Neural Networks*, *13*(3), 337-350.

[9] 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.

[10] Zabczyk, J. (2009). *Mathematical control theory: an introduction*. Springer Science & Business Media.