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

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.

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

*Correspondence E-mail: mas.abbasi81@gmail.com
IN TR O D U C TIO N

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:

\[
\min \theta \\
\text{s.t. } \sum_{i=1}^{n} \lambda_i x_{ij} \leq \theta x_p, \quad i = 1, 2, ..., m \\
\sum_{j=1}^{s} \lambda_j y_{rj} \geq y_r, \quad r = 1, 2, ..., s \\
\lambda_j \geq 0, \quad j = 1, 2, ..., n
\]

Where DMU_p=(x_p,y_p) (i=1,2,...,m) and y_r(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:

\[
\max \sum_{i=1}^{m} u_i y_p \\
\text{s.t. } \sum_{i=1}^{m} u_i x_{ij} - \sum_{j=1}^{n} v_j y_j \leq 0, \quad j = 1, 2, ..., n
\]
For solving 1 and (2) Simultaneously, we define the following convex energy function to design our proposed neural network:

\[ E(\lambda, u, v, \theta) = \frac{1}{2} \sum_{i=1}^{s} v_i x_{i} \quad \text{subject to} \quad 0 \leq (\lambda, u, v) \in \mathbb{R}^{m+n+s}, \quad X \text{ and } Y \text{ are input matrix and output matrix of all DMUs. Let } \ w=(\lambda, u, v, \theta)^{t}, \]

So 3 can be rewritten the following form:

\[ E(\lambda, u, v, \theta) = \frac{1}{2} \sum_{i=1}^{s} v_i x_{i} - \frac{1}{2} \max \{ A(\lambda, u, v) \} - \frac{1}{2} \max \{ B(\lambda, u, v) \} \]

Where

\[ A = \begin{pmatrix} X - x_i & 0 & 0 \\ -Y & 0 & 0 & 0 \\ 0 & Y' & -X' \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ -y_i \\ 0 \end{pmatrix} \]

\[ \Omega \text{ and } P_{\Omega} \text{ are defined as follow:} \]

\[ \Omega = \{ w \in \mathbb{R}^{m} \mid w_i \geq 0, i = 1, 2, \ldots, n + m + s \} \]

The proposed neural network has the following basic properties.

\[ \text{Lem m a 3.1. The equilibrium of the neural network in 5 is equal to solution of 1 and 2. Moreover, for any initial point } w_0 = w(t_0), \text{ there exist a unique continuous solution } w(t) \text{ for 5 over } [t_0, \infty). \]

\[ \text{Proof. Let } w^- \text{ be equilibrium of 5 so } V E(w^-) = 0, \text{ since } E \text{ is a convex function then } w^- \text{ be minimizer of } E(w). \text{ Moreover minimizer of } E(w) \text{ is solution of 1 and 2.} \]

\[ \text{Let } F(w) = \left[ C^{t_1} (C P_{\Omega} (w) - c) - D^{t_1} (D P_{\Omega} (w) - d) \right] \]

for any \( \bar{w}, \hat{w} \in \mathbb{R}^{m+n+s} \) we have:

\[ \|F(\bar{w}) - F(\hat{w})\| \leq \alpha \|P_{\Omega}(\bar{w}) - P_{\Omega}(\hat{w})\| \]

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

\[ \|F(\bar{w}) - F(\hat{w})\| \leq \alpha \|P_{\Omega}(\bar{w}) - P_{\Omega}(\hat{w})\| \]

THEORETICAL ANALYSIS

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

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

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

\[ \|F(x) - F(y)\| \leq L \|x - y\| \]

The equilibrium points of system described by Eq. 4 can be applied to identify optimal solution of 1 and 2.
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 \( w(t) \subseteq Ω \) 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:

\[
\frac{dV(w(t))}{dt} = \frac{dV}{dw} \frac{dw}{dt} = -\nabla E(w(t)) \cdot \nabla E(w(t)) \tag{7}
\]

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:

\[
\frac{dV(w(t))}{dt} = -\nabla E(w(t)) \cdot \nabla E(w(t)) \leq 0, \forall t \in [t_0, T) \tag{7}
\]

Thus:

\[
\| w(t) - w^* \|^2_2 \leq \beta \| w(t_0) - w^* \|^2_2, \forall t \in [t_0, T)
\]

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 \( w^k \in Ω \) and

\[
\lim_{k \to \infty} \| w^k \| = +\infty, \text{ by property of level sets (Ortega & Rheinboldt, 1970)}
\]

we see that all the levels sets of \( V \) are bounded though all level sets of \( E(w) \) are unbounded, thus \( Ω^- = \{ w \in Ω \mid V(w) \leq V(w_0) \} \) is bounded. Because \( V(w) \) is continuously differentiable on the compact set \( Ω^- \) and \( \{ w(t) \mid t \geq t_0 \} \subseteq Ω^- \), it follows from the LaSalle’s invariance principle (Miller & Michel, 1982) that trajectories \( w(t) \) converge to \( K \), the largest invariant subset of \( K = \{ w \mid \text{dV/dt=0} \} \). Note that if \( \text{dV/dt=0} \) then \((w-w^*)^TVE(w)=0\), 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 \( \text{dV/dt=0} \) then \( V(w)=0 \) and \((w-w^*)^TVE(w)=0\). So

\[
\frac{dV(w(t))}{dt} = 0 \iff \frac{dw}{dt} = 0
\]

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

\[
\| w(t_k) - \hat{u} \| < \epsilon, \quad k \geq q
\]

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

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

So

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

By (7) we have:

\[
\frac{dV(w(t))}{dt} \leq 0
\]

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

<table>
<thead>
<tr>
<th>DMUs</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$y_3$</th>
<th>$y_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td>2</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>3.0</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td>3</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>3.0</td>
<td>2.0</td>
</tr>
<tr>
<td>4</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>3.0</td>
</tr>
<tr>
<td>5</td>
<td>1.0</td>
<td>1.0</td>
<td>2.0</td>
<td>2.5</td>
<td>3.5</td>
<td>2.0</td>
</tr>
<tr>
<td>6</td>
<td>1.0</td>
<td>3.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>4.0</td>
</tr>
<tr>
<td>7</td>
<td>2.0</td>
<td>1.0</td>
<td>2.0</td>
<td>2.5</td>
<td>2.25</td>
<td>3.0</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DMUs</th>
<th>$\lambda_1$</th>
<th>$\lambda_2$</th>
<th>$\lambda_3$</th>
<th>$\lambda_4$</th>
<th>$\lambda_5$</th>
<th>$\lambda_6$</th>
<th>$\lambda_7$</th>
<th>$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.2</td>
<td>0</td>
<td>0</td>
<td>0.6</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
<td>0.25</td>
<td>0.5</td>
<td>0.75</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DMUs</th>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$u_1$</th>
<th>$u_2$</th>
<th>$u_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1393</td>
<td>0.3607</td>
<td>0.2214</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0.3753</td>
<td>0.1247</td>
<td>0</td>
<td>0.2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0.1909</td>
<td>0.3091</td>
<td>0.25</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0.25</td>
<td>0.25</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0.3398</td>
<td>0.6602</td>
<td>0.2344</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0.2877</td>
<td>0.2374</td>
<td>0</td>
<td>0</td>
<td>0.0033</td>
</tr>
<tr>
<td>7</td>
<td>0.0369</td>
<td>0.9263</td>
<td>0</td>
<td>0.2285</td>
<td>0.0144</td>
</tr>
</tbody>
</table>

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


