Computers & Industrial Engineering 56 (2009) 249–254

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A combined neural network and DEA for measuring efficiency of large scale datasets

Ali Emrouznejad [a,*], Estelle Shale [b]

[a] Operations & Information Management, Aston Business School, Aston University, Birmingham B4 7ET, UK
[b] Operational Research and Management Sciences, Warwick Business School, University of Warwick, Coventry CV4 7AL, UK

ARTICLE INFO

Article history:
Received 26 January 2007
Received in revised form 12 December 2007
Accepted 23 May 2008
Available online 11 July 2008

Keywords:
Neural networks
Data Envelopment Analysis
Large datasets
Back-propagation DEA

ABSTRACT

Data Envelopment Analysis (DEA) is one of the most widely used methods in the measurement of the efficiency and productivity of Decision Making Units (DMUs). DEA for a large dataset with many inputs/outputs would require huge computer resources in terms of memory and CPU time. This paper proposes a neural network back-propagation Data Envelopment Analysis to address this problem for the very large scale datasets now emerging in practice. Neural network requirements for computer memory and CPU time are far less than that needed by conventional DEA methods and can therefore be a useful tool in measuring the efficiency of large datasets. Finally, the back-propagation DEA algorithm is applied to five large datasets and compared with the results obtained by conventional DEA.

1. Introduction

Data Envelopment Analysis is a linear programming technique for assessing the efficiency and productivity of Decision Making Units (DMUs). Over the last decade DEA has gained considerable attention as a managerial tool for measuring performance. It has been used widely for assessing efficiency, in the public and private sectors, of organizations such as banks, airlines, hospitals, universities and manufacturers (Charnes, Cooper, & Rhodes, 1978). As a result, new applications with more variables and more complicated models are being introduced (Emrouznejad, Tavares, & Parker, in press).

DEA for a large dataset with many input/output variables and/or DMUs would require huge computer resources in terms of memory and CPU time. Such applications are now emerging in practice in both public and private sectors driven in part by the availability of very large quantitative datasets in data warehouses. This paper explores an alternative algorithm using a neural network to estimate the efficiency of DMUs in large datasets. This method offers considerable computational savings.

The paper unfolds as follows. The DEA technique and method of calculation in DEA are explained in Section 2. Section 3 describes a neural network algorithm for DEA (NNDEA) followed by a back-propagation DEA algorithm in Section 4. This paper uses five large datasets of DMUs to compare the result of NNDEA with conventional DEA calculations in Section 5. This is followed by conclusions in Section 6.

2. About DEA

DEA is a method for measuring the efficiency of DMUs using linear programming techniques to "envelop" observed input–output vectors as tightly as possible. One main advantage of DEA is that it allows several inputs and several outputs to be considered at the same time. In this case, efficiency is measured in terms of inputs or outputs along a ray from the origin.

Assume a set of observed DMUs, $\{DMU_j;\ j = 1,\ldots,n\}$, associated with m inputs, $\{x_{ij};\ i = 1,\ldots,m\}$, and s outputs, $\{y_{rj};\ r = 1,\ldots,s\}$. In the method originally proposed by Charnes et al. (1978), (often referred to as the DEA-CCR model) the efficiency of the DMU_{j_0} is defined as follows.

Model 1. Output oriented – CRS model

$$
\begin{aligned}
&\text{Max } h\\
&\text{s.t.}\\
&\qquad \sum \lambda_j x_{ij} + S_i^+ = x_{ij_0} \quad \forall i\\
&\qquad \sum \lambda_j y_{rj} - S_r^- = h y_{rj_0} \quad \forall r\\
&\qquad S_i^+, S_r^- \geqslant 0 \quad \forall i, \forall r\\
&\qquad \lambda_j \geqslant 0 \quad \forall j.
\end{aligned}
$$

Where

x_{ij} the amount of the ith input at DMU_j,
y_{rj} the amount of the rth output from DMU_j, and
j_0 the DMU to be assessed.

If h^* is the optimum value of h, then DMU_{j_0} is said to be efficient if $h^* = 1$ and the optimal values of S_i^- and S_r^+ are zero for all i and r. In Model (1), S_i and S_r represent slack variables. Thus a slack in an

* Corresponding author. Tel.: +44 121 204 3092.
E-mail addresses: A.Emrouznejad@aston.ac.uk (A. Emrouznejad), Estelle.Shale@wbs.ac.uk (E. Shale).

0360-8352/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2008.05.012

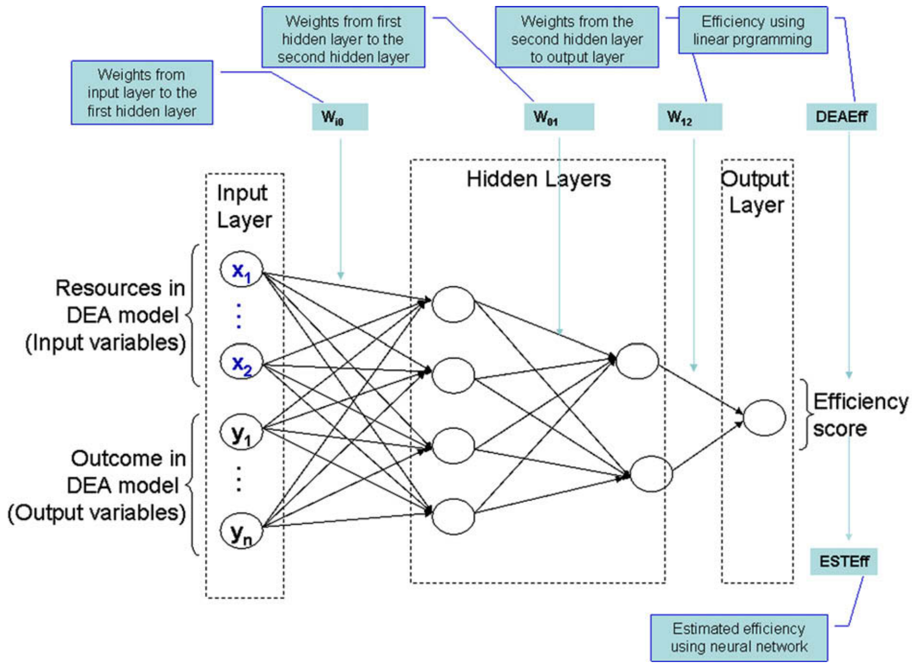

Fig. 1. Back-propagation DEA.

input i, i.e. $S_i^* > 0$, represents additional inefficiency in the use of input i. A slack in an output r, i.e. $S_r^* > 0$, represents an additional inefficiency in the production of output r.

The DEA Model (1) is known as an output – oriented model because it expands the output of DMU$_{j0}$ within the production space. It must be solved *n times*, once for each DMU being evaluated, to generate n optimal values of (h^*, λ^*).

For DMU$_{j0}$

- If radial expansion is possible Model (1) will yield $h_{j_0}^* > 1$.
- If radial expansion is not possible Model (1) will yield $h_{j_0}^* = 1$.

Examples of applications of DEA can be seen in Cooper, Seiford, and Zhu (2004), Emrouznejad and Podinovski (2004), Emrouznejad (2003), Emrouznejad and Thanassoulis (2005), Mulwa et al. (2008), Emrouznejad and Amin (in press), Kirigia, Emrouznejad, Vaz, Bastiene, and Padayachy (2008), also see Emrouznejad et al. (in press) for a full bibliography of DEA. Due to the complexity of DEA calculation several specialist software products have been developed (e.g. Emrouznejad, 2005; Emrouznejad & Thanassoulis, 2006).

In addition to the theoretical development of DEA, practitioners in a number of fields have quickly recognized that DEA is a useful methodology for measuring productivity and efficiency. Recently some large organizations have started to use DEA for evaluation of millions of DMUs. An example of this is the use of DEA at pupil level by the Department for Education and Skills (DfES) in England. An application of this kind needs very many calculations. Even with a very fast computer it may take a long time to get the results as a separate linear programme must be solved for each DMU.

The aim of the NNDEA developed in this paper is to select a random set of DMUs for training a neural network and then use the generated model for estimating the efficiency scores without any need to solve linear programming problems for every single DMU. Since NNDEA requirements for computer memory and CPU time are far less than those which are needed by conventional methods of DEA it can be a useful tool in measuring efficiency in large datasets.

3. Neural network DEA (NNDEA)

The most popular neural network algorithm is the back-propagation algorithm, proposed in the 1980s (Rumelhart, Hinton, & Williams, 1986). A back-propagation algorithm per-

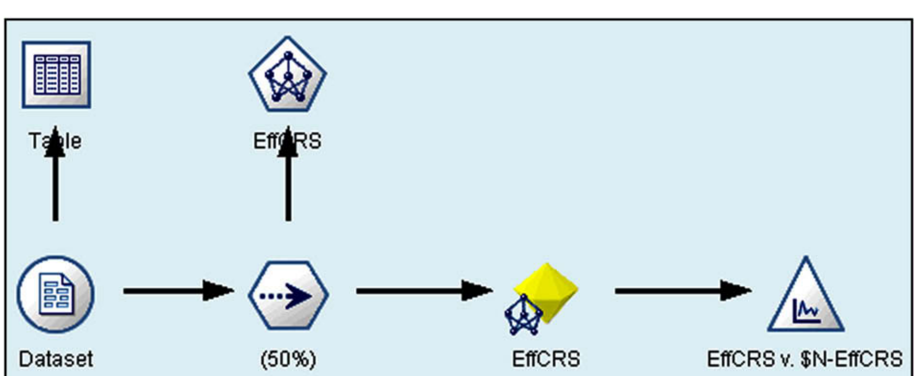

Fig. 2. A Clementine NNDEA stream for training.

forms learning on a multilayer feed-forward neural network. An example of such a network that we may use for measuring the efficiency of Decision Making Units is shown in Fig. 1.

The NN inputs correspond to the attributes that can be used to measure the DEA efficiency (i.e. resources and outcome variables in DEA) and the NN output corresponds to the value that should be predicted (i.e. DEA efficiency score). The inputs are fed simultaneously into a layer of units making up the *input layer*. The weighted outputs of theses units are, in turn, fed simultaneously to a second layer of "neuron-like" units, known as a *hidden layer*. The hidden layer's weighted outputs can be input to *another hidden layer*, and so on. The number of hidden layers is arbitrary, although in practice, usually only one (or maximum three) is used. The weighted outputs of the last hidden layer are input to the unit making up the *output layer*, which produces the network's prediction for given set of DMUs.

The multilayer NNDEA shown in Fig. 1 has two hidden layers and one output layer and therefore we refer to it as a three layer neural network. The network is feed-forward in that none of the weights cycles back to an input unit or to an output unit of a previous layer. It is fully connected in that every unit provides input to each unit in the next forward layer.

This paper shows that a multilayer feed-forward DEA network, given sufficient hidden layers, can closely approximate DEA efficiency score for DMUs in a large dataset.

4. Back-propagation DEA

Back-propagation DEA learns by iteratively processing a training sample, comparing the network's prediction of efficiency scores for each sample of DMUs with actual known efficiency scores. For each training sample, the weights are modified so as to minimize the mean squared error between the network's prediction and actual efficiency score as obtained in a conventional DEA model. These modifications are made in the "backwards" direction, that is, from the output layer, through each hidden layer down to the first hidden layer (hence the name back-propagation). The algorithm is summarized as follows:

Back-propagation DEA algorithm

1) Initialize all weights // usually to small random numbers //
2) While terminating condition is not satisfied {
3) For each training sample of DMUs in samples
{
4) For each hidden layer neuron j
{
// note that for resource variables $x_1 \dots x_n$ and
// outcome variables $y_1 \dots y_n$ the $O_k = I_k$, θ_k is bias //
5) $I_j = \sum_i w_{ij} O_j + \theta_j$
6) $O_j = 1/(1+e^{-I_j})^{-1}$;
}
7) $Err_j = DEAeff_j\,(1 - DEAeff_j)\,(ESTeff_j - DEAeff_j)$
// $DEAeff_j$ is the efficiency as obtain from DEA
// $ESTeff_j$ is the efficiency as estimated by neural network
8) For each unit j in the hidden layers
9) $Err_j = O_j(1 - O_j) \sum_k Err_k w_{jk}$;
10) For each weight w_{ij} in network
{
11) $\Delta w_{ij} = (l)\, Err_j \times O_j$;
12) $w_{ij} = w_{ij} + \Delta w_{ij}$;
}
13) For each bias θ_j in network
{
14) $\Delta \theta_j = (l)\, Err_j$
15) $\theta_j = \theta_j + \Delta \theta_j$;
}
16) }
17) }

5. NNDEA in practice

Generally before training can begin in any neural network, the user must decide on the network topology by specifying the

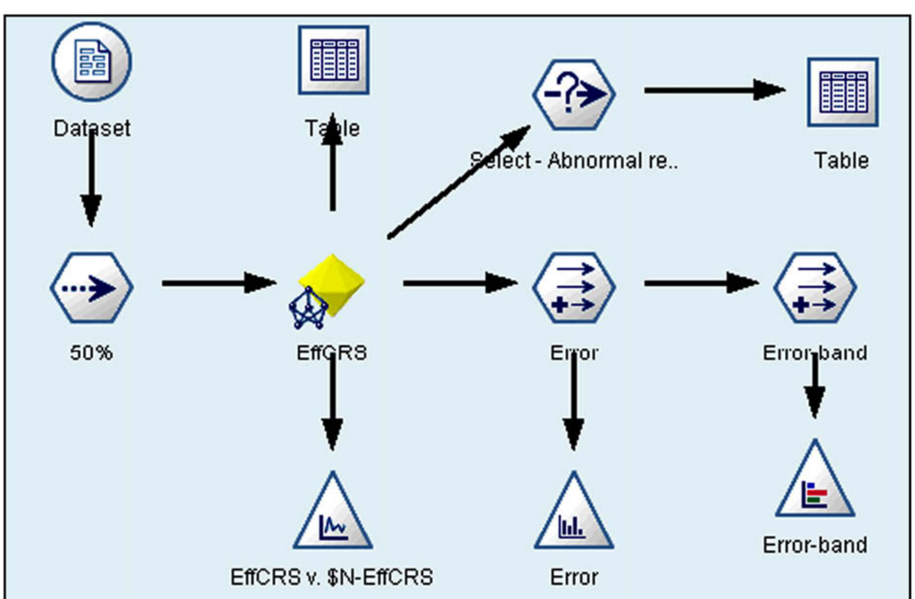

Fig. 3. A Clementine stream for prediction DEA score using NNDEA.

number of variables in the input layer, the number of hidden layers (if more than one), the number of variables in each hidden layers, and the number of variables in the output layer. In the NNDEA we use resources and outcomes in the corresponding DEA model as variables in the input layer and DEA efficiency score as the only variable in the output layer.

There are no clear rules as to the "best" number of hidden layer units. Network design is a trail and error process and may affect the accuracy of the resulting trained NNDEA. The initial values of the weights may also affect the resulting accuracy. Once a network has been trained and its accuracy is not considered acceptable, it is recommended to repeat the training process with a different network topology or a different set of initial weights. This paper uses a five datasets with six fields to demonstrate the use of NNDEA to calculate the efficiency of Decision Making Units. We are particularly interested in the efficiency scores for large datasets. The analysis is conducted in three stages, (1) training NNDEA with a sample of DMUs, (2) testing NNDEA with another sample of DMUs, and (3) estimating the DEA efficiency score using the generated NNDEA model.

For the purpose of demonstration five samples of data have been used, each containing approximately 10,000 DMUs. Each sample has been randomly divided into two half sets of DMUs for training and testing NNDEA. The DEA-CCR efficiency scores

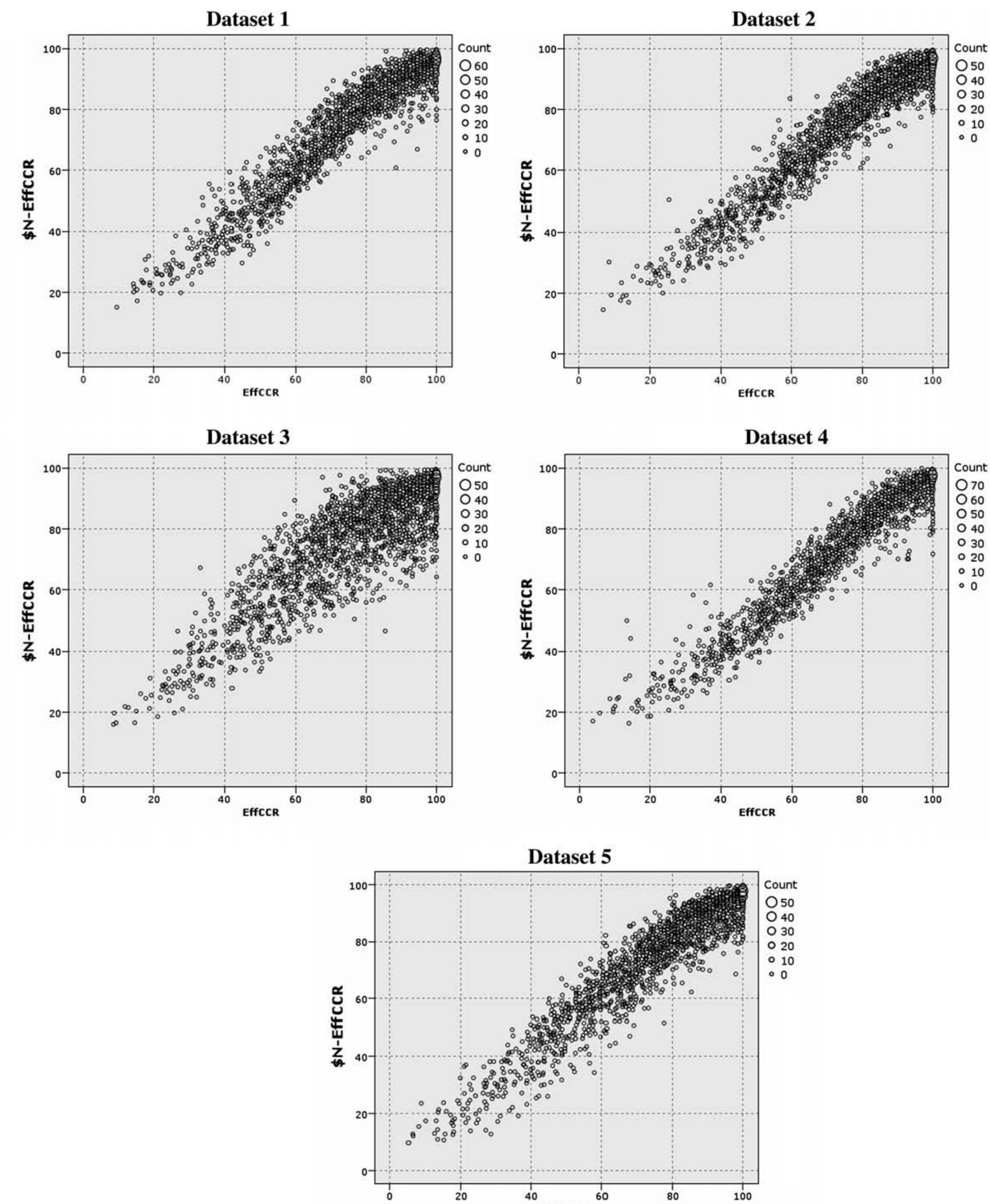

Fig. 4. NNDEA prediction as compared with actual DEA efficiency score for the five datasets.

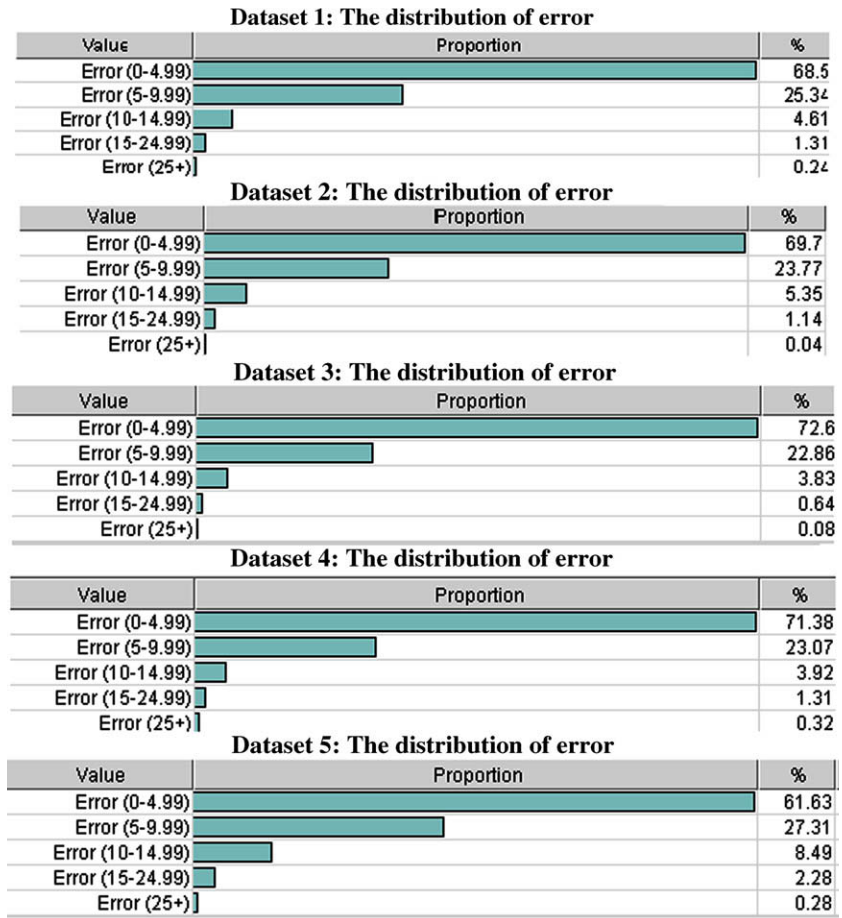

Fig. 5. The distribution of error in the five datasets.

have been obtained using PIM-DEAsoft (Emrouznejad & Thanassoulis, 2005). The NNDEA itself has been implemented in SPSS-Clementine software (http://www.spss.com/clementine/). Fig. 2 shows the stream for training.

After training NNDEA, the generated model is tested by estimating the efficiency score of the test dataset, the Clementine stream for this is presented in Fig. 3.

The $N-EffCRS field generated by the model indicates the efficiency score as calculated by NNDEA as compared to the actual efficiency score of EffCRS as obtained from the DEA-CCR model. The plots in Fig. 4 show that the NNDEA predictions for efficiency score appear to be a good estimate for the majority of cases.

As can be seen from Fig. 5 the efficiency scores of between 88.9 and 95.4% of the DMUs are estimated to be correct or within 10% and for at most 2.5% of DMUs does the estimated NNDEA score vary by 15% or more from the actual DEA-CCR efficiency score. The authors have also used NNDEA on several larger datasets. The results indicate that the error is far less when the dataset is very large and therefore the method performs best for the class of problems for which it is intended.

A second measure of similarity between the scores obtained is in terms of the rank ordering of the DMUs by the two methods of estimation. This can be assessed by calculating Spearman's Rho measure of bivariate correlation of the ranks. The results are given in Table 1. All the correlations are significant at the 0.01 level (2-tail test). Hence the rank ordering of the DMUs is very similar.

Finally, one can use the Wilcoxon Signed Ranks test (see Table 2) to investigate the probability that the two sets of efficiency scores come from the same underlying distribution (the null hypothesis).

Table 1
Spearman's correlation and p-value for bivariate comparison of NNDEA and conventional DEA CCR scores

	Spearman's Rho	p-Value
Dataset 1	0.939	0.000
Dataset 2	0.941	0.000
Dataset 3	0.792	0.000
Dataset 4	0.945	0.000
Dataset 5	0.929	0.000

Four of the five tests shown in Table 3 indicate a high level of probability that the NNDEA and conventional DEA-CCR scores come from the same underlying estimator of efficiency. One dataset fails the test i.e. the null hypothesis is rejected. Given the number of observations in the test samples this is a strong result as Conover (1999) has noted that if one employs a large enough sample size, almost any goodness of fit test will result in the rejection of the null hypothesis.

These results and observations also cast some light on the issue of extrapolation. In the experimental design the NNDEA method has been tested on data other than that used to train the network but both sets come from the same underlying data collection exercise. This is how the method is proposed to be used in practice with very large sets with perhaps millions of DMUs. A large random sample would be selected to train the network before using NNDEA for assessing the efficiency of all the observations. Sampling theory suggests that a random sample of this size will be a good descriptor

Table 2
Wilcoxon's Signed Ranks for the comparison of NNDEA and conventional DEA CCR scores

	N	Mean rank	Sum of ranks
$N-EffCCRset1 – EffCCRset1			
Negative ranks	1318[a]	1220.79	1609005.00
Positive ranks	1196[b]	1297.95	1552350.00
Ties	0[c]		
Total	2514		
$N-EffCCRset2 – EffCCRset2			
Negative ranks	1239[d]	1233.44	1528234.50
Positive ranks	1226[e]	1232.55	1511110.50
Ties	0[f]		
Total	2465		
$N-EffCCRset3 – EffCCRset3			
Negative ranks	1272[g]	1216.64	1547564.50
Positive ranks	1235[h]	1292.48	1596213.50
Ties	0[i]		
Total	2507		
$N-EffCCRset4 – EffCCRset4			
Negative ranks	1367[j]	1260.99	1723773.50
Positive ranks	1156[k]	1263.19	1460252.50
Ties	0[l]		
Total	2523		
$N-EffCCRset5 – EffCCRset5			
Negative ranks	1265[m]	1245.09	1575045.00
Positive ranks	1232[n]	1253.01	1543708.00
Ties	0[o]		
Total	2497		

[a] $N-EffCCRset1 < EffCCRset1.
[b] $N-EffCCRset1 > EffCCRset1.
[c] $N-EffCCRset1 = EffCCRset1.
[d] $N-EffCCRset2 < EffCCRset2.
[e] $N-EffCCRset2 > EffCCRset2.
[f] $N-EffCCRset2 = EffCCRset2.
[g] $N-EffCCRset3 < EffCCRset3.
[h] $N-EffCCRset3 > EffCCRset3.
[i] $N-EffCCRset3 = EffCCRset3.
[j] $N-EffCCRset4 < EffCCRset4.
[k] $N-EffCCRset4 > EffCCRset4.
[l] $N-EffCCRset4 = EffCCRset4.
[m] $N-EffCCRset5 < EffCCRset5.
[n] $N-EffCCRset5 > EffCCRset5.
[o] $N-EffCCRset5 = EffCCRset5.

Table 3
Wilcoxon's Signed Ranks test statistics for the comparison of NNDEA and conventional DEA CCR scores

	$N-EffCCRset1 – EffCCRset1	$N-EffCCRset2 – EffCCRset2	$N-EffCCRset3 – EffCCRset3	$N-EffCCRset4 – EffCCRset4	$N-EffCCRset5 – EffCCRset5
Z	−.778[a]	−.242[a]	−.671[b]	−3.601[a]	−.435[a]
Asymp. Sig. (2-tailed)	.436	.809	.502	.000	.664

[a] Based on positive ranks.
[b] Based on negative ranks.

of the underlying production technology pertaining throughout the system. Thus the network has an excellent source of information to learn about the performance of the system as a whole in transforming inputs to outputs. Troutt, Rai, and Zhang (1995) suggests that there should be at least 10 training observations for each input to the network in order to avoid the problem of under training. With such large datasets this is unlikely to be an issue.

6. Conclusions

DEA is a non-parametric method that is widely used for measuring the efficiency and productivity of Decision Making Units. DEA for a large dataset with many input/output variables and/or many DMUs would require huge computer resources in terms of memory and CPU time. This paper combined a neural network with DEA to introduce an alternative algorithm and approach to estimating the efficiency of DMUs in large datasets.

The NNDEA back-propagation algorithm has been used for measuring the efficiency of sets of DMUs and the results indicate that the NNDEA prediction for efficiency score appears to be a good estimate for the majority of DMUs. An analysis of error shows that the larger the dataset the smaller error.

In terms of the topology used in NNDEA, further investigation could be undertaken to investigate the number of hidden layers and the number of units in the hidden layers to minimize error in estimating the DEA efficiency score. This should enable the already strong similarity between the NNDEA and conventional DEA scores to be enhanced. However, Curram, Athanassopoulos, and Shale (submitted for publication) found for smaller datasets (up to 100 DMUs) that a neural network approach to estimating efficiency has a certain degree of robustness to the number of hidden neurons provided there are sufficient units to model the nonlinearity of the efficient frontier. Nevertheless, as few as four units in a single hidden layer performed well for the 16 datasets examined in that paper.

References

Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. *European Journal of Operational Research, 2*, 429–444.
Conover, W. J. (1999). *Practical nonparametric statistics*. New York: Wiley.
Cooper, W. W., Seiford, L. M., & Zhu, J. (2004). *Handbook on data envelopment analysis*. Kluwer Academic.
Curram, S. P., Athanassopoulos, A. D., & Shale, E. A. (submitted for publication). A frontier-based neural network for assessing the efficiency of activity units. *European Journal of Operational Research*.
Emrouznejad, A. (2003). An alternative DEA measure: A case of OCED countries. *Applied Economic Letters, 10*, 779–782.
Emrouznejad, A. (2005). Measurement efficiency and productivity in SAS/OR. *Journal of Computers and Operation Research, 32*(7), 1665–1683.
Emrouznejad, A., & Amin, G. R. (in press). DEA models for ratio data: Convexity consideration. *Applied Mathematical Modeling*.
Emrouznejad, A., & Thanassoulis, E. (2006). Performance Improvement Management Software, (PIM-DEAsoft). Available from www.DEAsoftware.co.uk.
Emrouznejad, A., Tavares, G., & Parker, B. (in press). A bibliography of Data Envelopment Analysis (1978–2006). *Socio-Economic Planning Sciences*.
Emrouznejad, A., & Podinovski, V. (Eds.). (2004). *Data envelopment analysis and performance management*. 0 902683 73 X (pp. 421). UK: Aston University Print.
Emrouznejad, A., & Thanassoulis, E. (2005). A mathematical model for dynamic efficiency using data envelopment analysis. *Journal of Applied Mathematics and Computation, 160*(2), 14.
Kirigia, J. M., Emrouznejad, A., Vaz, R. G., Bastiene, H., & Padayachy, J. (2008). A comparative assessment of performance and productivity of health centers in Seychelles. *International Journal of Productivity & Performance Management, 57*(1).
Mulwa, R., Emrouznejad, A., & Muhammad, L. (2008). Economic Efficiency of smallholder maize producers in Western Kenya: a DEA meta-frontier analysis. *Int. J. Operational Research, 3*(6).
Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*, 533–536.
Troutt, M. D., Rai, A., & Zhang, A. (1995). The Potential Use of DEA for Credit Applicant Acceptance Systems. *Computers and Operations Research, 4*, 405–408.