Back-propagation DEA

Ali Emrouznejad
Operations & Information Management
Aston Business School,
Aston University,
Birmingham
B4 7ET,
UK
a.emrouznejad@aston.ac.uk

Abstract

Data Envelopment Analysis (DEA) is one of the most widely used methods in the measurement 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 introduces a neural network back-propagation Data Envelopment Analysis. Neural network requirements of computer memory and CPU time are far less than what is needed by conventional methods DEA and can be a useful tool in measuring efficiency of large datasets. Finally, the back-propagation DEA algorithm is applied to a large dataset to identify the source of inefficiency of DMUs and compare it with the result obtained by conventional DEA.

Keywords: Neural Network, Data Envelopment Analysis, Large Dataset, Back-propagation 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 of organizations and it has been used widely for assessing the efficiency of public and private sectors such as banks, airlines, hospitals, universities and manufactures (Charnes, Cooper, and Rhodes; 1978). As a result, in the last three decade new applications with more variables and more complicated models are being introduced (Emrouznejad and Podinovski; 2004).

In the DEA the assumption is that DMUs typically consume multiple resources in producing multiple products. The technical efficiency of a DMU is measured by weighted sum of its outputs divided by weighted sum of its inputs, that is, the ratio of its virtual output to its virtual input. Within the context of input augmentation, a DMU is considered technically efficient, or Pareto–Koopmans efficient, if the performance of other DMUs does not provide evidence that the same amount of all outputs could be produced while using less of at least one input and no more of any other input. Given an output orientation, a DMU is technically efficient if the performance of the other DMUs does not suggest that the same amount of all inputs could be consumed while producing more of at least one output and no less of any other output. Therefore DEA is a non-parametric method which uses linear programming to construct a piece-wise linear segmented efficiency frontier based on best practice.

DEA for a large dataset with many input/output variables would require huge computer resources in terms of memory and CPU time. This paper explores an alternative algorithm using neural network to estimate efficiency of DMUs in large datasets.

The paper unfolds as follows. The DEA technique and method of calculations in DEA are explained in section 2. Section 3 describes a neural Network algorithm for DEA (NNDEA) followed by back-propagation DEA algorithm in section 4. Finally, this paper uses a large dataset of DMUs to show the result of NNDEA as compared with conventional DEA calculation in section 5. This follows by conclusion in section 6.

2. About DEA

DEA is a method for measuring 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, $\{DMU_j; j=1,...,n\}$, associated with m inputs, $\{x_{ij} ;\ i=1,...,m\}$, and s outputs, $\{y_{rj} ; r=1,...s\}$. In the method originally proposed by Charnes, Cooper, Rhodes (1978) the efficiency of the DMU_{j0} is defined as follows.

Model 1. Output oriented - CRS model

$$\text{Max } h$$

s.t.

$$\sum \lambda_j x_{ij} + S_i^+ = x_{ij_0} \qquad \forall i$$

$$\sum \lambda_j y_{rj} - S_r^- = h y_{rj_0} \qquad \forall r$$

$$S_i^+ ,\ S_r^- \geq 0 \qquad \forall i, \forall r$$

$$\lambda_j \geq 0 \qquad \forall j.$$

Where;

x_{ij} = the amount of the i^{th} input at DMU_j,
y_{rj} = the amount of the r^{th} output from DMU_j and
j0 = the DMU to be assessed.

If h^* is the optimum value of h, then DMU_{j0} is said to be Pareto efficient iff $h^*=1$ and the optimal values of S_i^- & S_r^+ are zero for all i & r. In Model (1), S_i and S_r represent slack variables. Thus a slack in an input i, i.e. $S_i^*>0$, represents an additional inefficiency 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 output - oriented model because it expands output of DMU_{j0} within the production space. It should 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 $h^*_{j0} > 1$,
- If radial expansion is not possible Model (1) will yield $h^*_{j0} = 1$.

For other DEA models see DEA website (www.DEAzone.com). Examples of applications of DEA can be seen in Field and Emrouznejad (2003) and Kirigia *et al*. (2002 and 2004). Due to complication of DEA calculation several software have been developed (e.g. Emrouznejad (2004) and Emrouznejad and Thanassoulis (2005)).

Besides developing DEA in theory, 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 using DEA at pupil level by Department for Education and Skills (DfES) in England. However example of this kind needs a lot of calculations. Even with a very fast computer it may take days to get the results as one set of linear programming should be solved for each DMU.

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

3. Neural Network DEA (NNDEA)

The most popular neural network algorithm is the back-propagation algorithm, proposed in the 1980s (Rumelhart *et al.*; 1986). A back-propagation algorithm performs learning on a multilayer feed-forward neural network. An example of such network that we may use for measuring efficiency of decision making units is shown in Figure 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 correspond 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 units making up the *output layer*, which produces the network's prediction for given set of DMUs.

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

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

4. Back-propagation DEA

Back-propagation DEA learns by iteratively processing a set of training sample, comparing the network's prediction of efficiency score for each sample of DMUs with actual known efficiency score. 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 obtain in the 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:

<u>Back-propagation DEA algorithm</u>

```
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 unit j
              {
              // note that for resource variables x1…xn and outcome
              variables y1…yn the Oj=Ij //
5)            Ij= Σ_i wij Oi + θj
6)            Oi = 1 / (1 + e^(-Ij)) ;
              }
7)         Errj=DEAeffj (1- DEAeffj) ( ESTeffj - DEAeffj)
           // DEAeffj is the efficiency as obtain from DEA
           // ESTeffj is the efficiency as estimated by neural network
8)         For each unit j in the hidden layers
9)            Errj=Oj(1- Oj) Σ_i Errk wjk ;
10)        For each weight wij in network
              {
11)           Δwij =(l) Errj Oj;
12)           wij=wij +Δwij ;
              }
13)        For each bias θj in network
              {
14)           Δ θj =(l) Errj
15)           θj = θj + Δ θj;
              }
16)    }
17) }
```

Step 5: $$I_j = \sum_i w_{ij} O_i + \theta_j$$

Step 6: $$O_i = \frac{1}{1+e^{-I_j}};$$

Step 9: $$Err_j = O_j(1 - O_j) \sum_i Err_k w_{jk};$$

5. NNDEA in practice

Generally before training can begin in any neural network, the user must decide on the network topology by specifying the number of variables in 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 layers units. Network design is a trail-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 example uses a data set with nine fields to demonstrate the use of NNDEA to calculate the efficiency of Decision Making Units. We are particularly interested in efficiency score 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 purpose of this demonstration only one sample of data has been used. The sample then has been divided into two half set of DMUs for training and testing NNDEA. The DEA-CCR efficiency score has been obtained using DEAsoft for calculation of DEA scores. The $N-EffCRS field generated by the model indicates the efficiency score as calculated by NNDEA as compare with the actual efficiency score of EffCRS as obtained from DEA-CCR model. The plot in Figure 2 shows the NNDEA prediction for efficiency score appears to be a good estimate for the majority of cases.

As it can be seen in Figure 3 the efficiency scores of about 75% of DMUs are estimated to be correct or within 5% error and only for 4% of DMUs the estimated NNDEA score is within 15% or above the actual CCR-DEA efficiency score. The author used NNDEA on several large datasets. The result shows the error is far less when dataset is very large and therefore it can be used instead of calculating DEA score.

6. Conclusion

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

The NNDEA back-propagation algorithm has been used for measuring efficiency of a set of DMUs and the result indicates 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 dataset the smaller error.

In terms of topology used in NNDEA, further investigation is needed to choose number of hidden layers, number of units in the hidden layers. An analysis of source of error could help DEA users to focus on the source of inefficiency as well as to improve the NNDEA estimator for better accuracy.

7. References

Charnes, A., Cooper, W. W., and Rhodes, E. (1978), "Measuring the Efficiency of Decision Making Units," *European Journal of Operational Research*, 2, 429-444

Emrouznejad A. (1995). *Ali Emrouznejad's DEA HomePage*, www.DEAzone.com, Warwick Business School, Coventry CV4 7AL, UK.

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*, Volume 32, Issue 7, PP 1665-1683.

Emrouznejad A. and V. Podinovski (Editors) (2004) *Data Envelopment Analysis and Performance Management*, Warwick University Print, UK, 421 pages, ISBN: 0 902683 73 X

Emrouznejad A and E. Thanassoulis (2005) A mathematical model for dynamic efficiency using data envelopment analysis, *Journal of Applied Mathematics and Computation*, Volume 160, Issue 2, 14

Emrouznejad A and E. Thanassoulis (2006) *Performance Improvement Management Software*, www.DEAsoftware.co.uk.

Field K. and E. Emrouznejad (2003) Measuring the Performance of Neonatal Care Units In Scotland, *Journal of Medical Systems*, 27 (4) p.315-324

Kirigia JM, Emrouznejad A and Sambo LG (2002). Measurement of technical efficiency of public hospitals in Kenya: using Data Envelopment Analysis. *Journal of Medical Systems*, 26(1): 39-45.

Kirigia JM, Emrouznejad A, Sambo LG, Munguti N, Liambila W (2004) Using data envelopment analysis to measure the technical efficiency of public health centers in Kenya. *Journal of Medical Systems,* 28(2): 155-166.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323, 533-536.

Figure 1. Back-propagation DEA

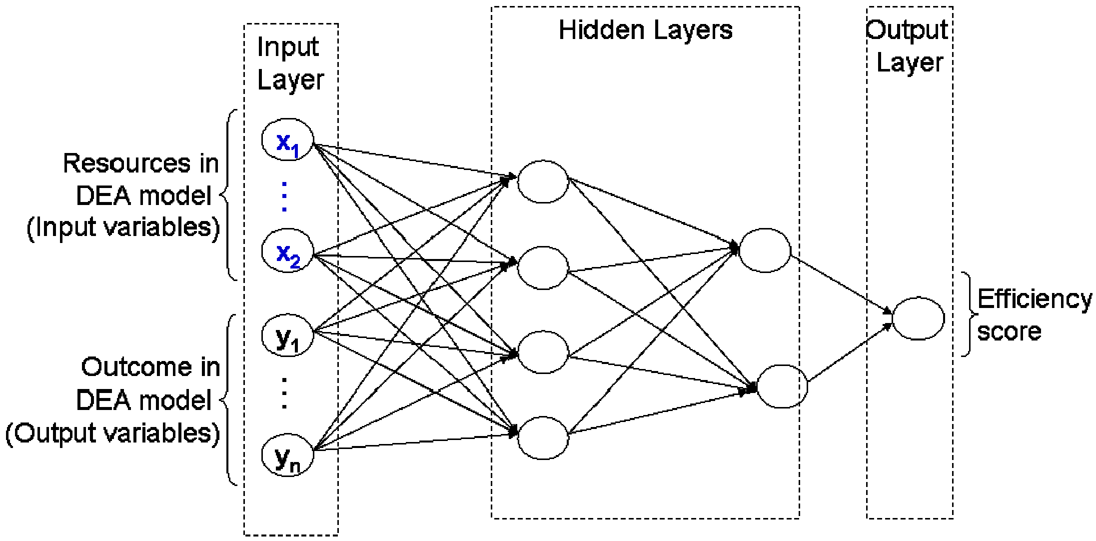

Figure 2: NNDEA prediction as compared with actual DEA-efficiency score

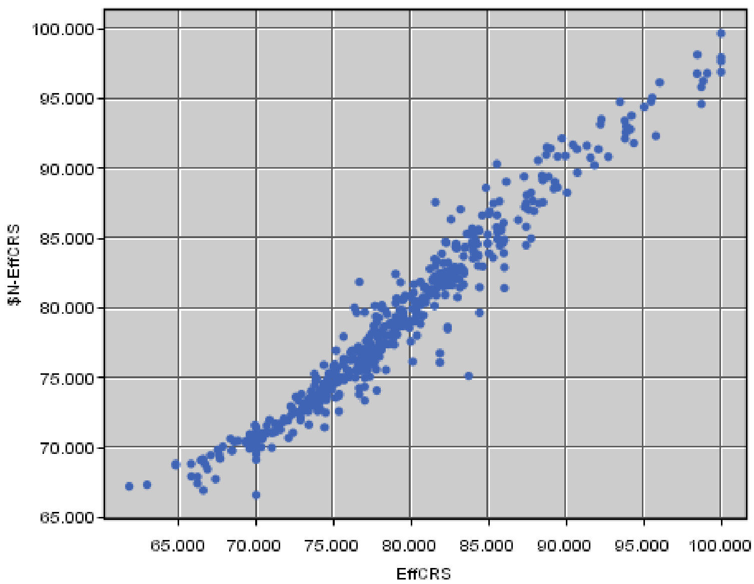

Figure 3: The distribution of error

Value	Proportion	%
Error (0 to 4.99)		73.6
Error (5 to 9.99)		15.8
Error (10 to 14.99)		6.6
Error (15 to 24.99)		3.6
Error (25+)		0.4