Neural network DEA for measuring the efficiency of mutual funds

Payam Hanafizadeh*
School of Management and Accounting,
Allameh Tabataba’i University,
Nezami Ganjavi Street, Tavanneer, Valy Asr Avenue,
P.O. Box 14155-6476, Tehran, Iran
Fax: +98-21-8877-0017
E-mail: hanafizadeh@gmail.com
*Corresponding author

Hamid Reza Khedmatgozar
Department of IT Management,
Iranian Research Institute for Information Science and Technology,
No. 1090, Enghelab Avenue, P.O. Box 13185-1371, Tehran, Iran
Fax: +98-21-6646-2254
E-mail: h.khedmatgozar@gmail.com

Ali Emrouznejad
Aston Business School,
Aston University,
Aston Triangle, Birmingham, B4 7ET, UK
E-mail: a.emrouznejad@aston.ac.uk

Mojtaba Derakhshan
Department of Financial Engineering,
University of Science and Culture,
Asharif Esfahani Blvd., Park Street,
P.O. Box 13145-871, Tehran, Iran
Fax: +98-21-4421-4750
E-mail: e.m.derakhshan@gmail.com

Abstract: Efficiency in the mutual fund (MF), is one of the issues that has attracted many investors in countries with advanced financial market for many years. Due to the need for frequent study of MF’s efficiency in short-term periods, investors need a method that not only has high accuracy, but also high speed. Data envelopment analysis (DEA) is proven to be 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 uses neural network back-propagation DEA in measurement of mutual funds efficiency and shows the requirements, in the
proposed method, 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 a large set of MFs.

Keywords: mutual fund; data envelopment analysis; DEA; back-propagation DEA; neural network; large dataset.


Biographical notes: Payam Hanafizadeh started his career in January 2001 at the School of Management and Accounting at the Allameh Tabataba’i University in Tehran, Iran. He was a Visiting Research Fellow at the University of Canberra, Australia in 2010 and a visiting scholar at the University of Waterloo, Canada in 2004. His research interest is on the broad area of information systems and decision science particularly decision making under uncertainty. He has written over 80 peer-reviewed research manuscripts, including 50+ journal articles, six books, six book chapters, and several conference papers and presentations.

Hamid Reza Khedmatgozar is a PhD candidate of Information Technology (IT) Management at the Iranian Research Institute for Information Science and Technology – IRANDOC, Tehran, Iran. He holds a BSc in Industrial Engineering from Yazd University, Yazd, Iran, and a Master’s degree in Financial Engineering from University of Science and Culture, Tehran, Iran. His research interests revolve around IT adoption and risk management in financial markets and digital identifier systems for information objects. He has published articles in *Electronic Commerce Research, Telematics and Informatics, Iranian Journal of Management Sciences* and *Sharif Industrial Engineering and Management Journal*.

Ali Emrouznejad is a reader in Operational Research and Management at the Aston Business School, Aston University, UK. He holds an MSc in Applied Mathematics and received his PhD in Operational Research and Systems from Warwick Business School, UK. His areas of research interest include performance measurement and management, efficiency and productivity analysis as well as data mining. He serves on the editorial board of several scientific journals; he is the Editor of *Annals of Operations Research*, Senior Editor and one of the founding members of the *Data Envelopment Analysis Journal*. He is the author of the book on *Applied Operational Research with SAS*.

Mojtaba Derakhshan is a graduated Master of Science in the Department of Financial Engineering at the University of Science and Culture in Tehran, Iran. He received his BSc in Industrial Engineering from Tehran Polytechnic University and pursues his research in portfolio selection and organisational diagnosis (OD). He has published some articles in journals such as the *International Journal of Industrial Engineering and Product Management* and accepted in the international IIE conference, among others.
1 Introduction

Data envelopment analysis (DEA) is a linear programming technique for assessing the efficiency of decision making units (DMUs). During the last three decades DEA has gained considerable attention as a managerial tool for measuring comparative performance of both private and public organisations such as banks, airports, hospitals, universities and industries (Charnes et al., 1978). As a result, many new applications with more complicated models are being introduced (Emrouznejad et al., 2008).

Examples of DEAs applications can be seen in Burger and Moormann (2010), Cooper et al. (2011), Djema and Djerdjouri (2012), Field and Emrouznejad (2003), Heidari et al. (2012), Ho and Wu (2008), Iazzolino et al. (2013), Kthiri et al. (2011), Meenakumari et al. (2009), Staat (2006), Sufian (2007, 2010) and Van der Meer et al. (2004). Due to the complexity of DEA calculation, several software have been developed (e.g., Emrouznejad and Thanassoulis, 2005, 2010; Zhanxin et al., 2012).

In addition of the theoretical development of DEA, practitioners in many fields have recognised that DEA is a useful methodology for measuring productivity and efficiency of homogeneous DMUs. Recently, some large organisations have started using DEA for evaluation of thousands of DMUs. Application like these, need careful calculations. Even with a very fast computer it may take a long time to get the results since DEA solves one linear programming for each DMU (Emrouznejad and Shale, 2009).

This paper explores the effective factors in determining mutual funds (MFs) efficiency and uses a combined algorithm of neural network and DEA to estimates the efficiency of a large set of MFs. It is shown that this method offers considerable computational savings in terms of mass and time.

The paper is organised as follows. Section 2 describes MFs. The DEA, method of calculation in DEA and neural network algorithm for DEA (NNDEA) are explained in Section 3. Section 4 describes the assessment procedure and the use of NNDEA for assessment of MFs. This is followed by discussion of the results in Section 5. Conclusions and future direction of research are given in Section 6.

2 Mutual fund

A mutual fund (MF) is a company that collects people and companies’ financial resources and invests in a portfolio of securities. People who buy shares of an MF are the owners or shareholders. Their investments provide the capital for an MF to buy securities such as stocks and bonds. An MF can make money from its securities in two ways: a security can pay dividends or interest to the fund or a security can raise in value. A fund can also lose money and fall in value (Hanafizadeh et al., 2009).

In the USA, investment company assets consist of Mutual, Close End and Exchange Trader Funds, it has been increased from 2,955 billion dollars in 1995 to 14,647 billion dollars in 2012. During the same time, the number of MFs has been increased from 6,263 to 10,593 (resource: Investment Company Institute).

The first question that rises in the mind of each investor is the efficiency assessment “which MF does have strong efficiency and which one weak?” Because of the large number of MFs in the world wide, e.g., about 10,000 in the USA only, investors need a suitable tool in order to be able to select the best MF for their investment. Further, they need to review the efficiency of MFs on a defined time period like on a daily, weekly,
monthly or annual basis. DEA is proven to be a useful tool for helping investors to select a more efficient MF.

Examples of applications of DEA in MFs can be seen in Basso and Funari (2003), Bricc et al. (2007), Chang (2004), Chen et al. (2011), Gregoriou et al. (2005), Haslem and Scheraga (2003), Joro and Na (2006), Lozano and Gutiérrez (2008), McMullen and Strong (1998), Murthi et al. (1997), Premachandra et al. (2012), Wilkens and Zhu (2001), Zhao and Wang (2007) and Zhao et al. (2011). By reviewing these studies, we can see that they generally follow two main goals:
1 improving the DEA model which is used evaluating the efficiency of MFs
2 developing the used variables (input and output) in DEA models for having a complete evaluation of MFs efficiency.

Often the investors want to know the results of the analysis very quickly, so speed is a very critical parameter, but DEA needs to run one linear programming for each MF, hence the results cannot be obtained at the speed the investors need. An alternative to DEA is a method uses combined DEA and neural network (NNDEA) as developed by Emrouznejad and Shale (2009). In this paper, we propose an NNDEA assessment procedure for comparison of MFs in order to reduce the time to get the results for investors.

3 DEA and NNDEA

DEA is a method for measuring the efficiency of DMUs using linear or non-linear programming techniques by input-output vectors of each DMU. One of the main advantages of DEA is that it allows several inputs and outputs to be considered at the same time. In this case, efficiency is measured based on the value which the input/output of every DMU takes.

Consider a set of observed DMUs, \{DMU_j; j = 1, ..., n\}, associated with \(m\) inputs as, \{\(x_{ij}; i = 1, ..., m\)\}, and \(s\) outputs as, \{\(y_{rj}; r = 1, ..., s\)\}. A standard constant return to scale DEA for measuring efficiency of \(DMU_p\) is formulated in model 1.

3.1 Model 1: input oriented – CRS model

\[
\begin{align*}
\text{Min} Y_p &= \theta_p - e \left[ \sum_{i=1}^{m} S_i^+ + \sum_{r=1}^{s} S_r^- \right] \\
\text{St: } &\sum_{j=1}^{n} \lambda_j y_{rj} - S_r^- = y_{rp} \quad r = 1,2,\ldots,s \\
&\sum_{j=1}^{n} \lambda_j x_{ij} + S_i^+ = \theta_p x_{ip} \quad i = 1,2,\ldots,m \\
&\lambda_j \geq 0; \quad j = 1,2,\ldots,n \\
&S_i^-, S_r^+ \geq 0; \quad \forall i, \forall r \\
&\theta_p \text{ free}
\end{align*}
\]
Neural network DEA for measuring the efficiency of mutual funds

where \( m \) is number of inputs, \( n \) is number of DMUs, \( s \) is number of outputs, \( x_{ij} \) is the \( i^{th} \) input of \( DMU_j \), \( y_{jr} \) is the \( r^{th} \) output of \( DMU_j \), \( p \) is the DMU under evaluation and \( \lambda_j \) is the weight of \( DMU_j \) during estimation.

Assume \( \theta^*_p \) is the optimum value of the objective function, if \( \theta^*_p = 1 \) and the optimal evaluation of \( S_i^- \) and \( S_r^+ \) are zero for all \( i \) and \( r \), then \( DMU_p \) is efficient. In model (1), \( S_i \) and \( S_r \) represent slack variables. Thus, a slack in an input \( i \), i.e., \( S_i^- \), represents additional inefficiency in the use of input \( i \). A slack in an output \( r \), i.e., \( S_r^+ \), represents an additional inefficiency in the production of output \( r \).

The DEA model (1) is known as an input – oriented model because it minimises the input of \( DMU_p \) within the production space. It must be solved \( n \) times, once for each DMU being evaluated, and generate \( n \) times the optimal values of \( \theta^*_p \), \( \lambda \), and \( S \).

As alternative to DEA, Emrouznejad and Shale (2009) proposed a combined NNDEA. An example of NNDEA network in measuring the efficiency of the DMUs is shown in Figure 1, this method does learning process on multi-layer neural network.

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.

Figure 1  Back-propagation DEA

The multi-layer NNDEA shown in Figure 1 has one hidden layers and one output layer and therefore we refer to it as a two-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. In such a network, output of every units of a layer, makes the input of the next layer.

The aim in this paper is to propose a multilayer feed-forward DEA network, given sufficient hidden layers which can be used for determination of the optimal evaluation of the efficiency of MFs with a good accuracy.

Back-propagation DEA, see Appendix, learns by iteratively processing a set of training sample, comparing the network’s prediction about efficiency amount for each sample of DMUs with estimated efficiency using standard DEA. For each training sample, the weights are corrected so as to minimise the mean squared error between the network’s prediction and the efficiency value as obtained in a conventional DEA model. These corrections are made in the ‘backwards’ direction. In Appendix, the back-propagation DEA algorithm, corresponding to Figure 1, for two layers neural network and one hidden layer that are used in this paper are shown in details.

4 An NNDEA model for measuring efficiency of MFs

The aim of using NNDEA in this paper is to select a random set of DMUs for training a neural network and to use the generated model for estimating the efficiency scores without any need to solve linear programming problems for every single MF (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 of a large-scaled dataset such as MFs.

To set up the assessment procedure we have to decide about suitable variables, to elaborate the data and to choose an appropriate model.

4.1 Variable selection

For selecting input and output variables, having considered all previous studies in this area, it can be said that there is no agreement on the selection of variables for evaluation of MFs efficiency. However, we decided to choose the most popular variables. According to traditional financial theories and mean-variance portfolio theory (Markowitz, 1952), investors make their investment choices by considering simultaneously the returns approximated by the mathematical mean return and the risks, including both systematic and non-systematic risks, measured by the return dispersion (represented by variance or standard deviation). Recent studies enlarge the evaluation dimension to the skewness (3rd moment), and even other higher moments in order to take into account the non-normality of return distributions (see, for example, Gregoriou and Gueyie, 2003; Leland, 1999; Sortino and Price, 1994; Stutzer, 2000). Negative skewness means there is a substantial probability of a big negative return and positive skewness means that there is a greater-than-normal probability of a big positive return.

While some investors might be more concerned with central tendencies (mean, variance), others may concern more about extreme values such as skewness. Let us consider the positive preference of individuals for skewness first invoked by Arditti (1967). This implies that individuals will be willing to accept a lower expected value from their investments in portfolio A, than in portfolio B, if both portfolios have the same variance, and if portfolio A has greater positive skewness and all higher moments are the
same. In other words, individuals attach a higher importance to the skewness than to the mean of returns.

In measuring the efficiency of MFs, Nguyen-Thi-Thanh (2004) chooses standard deviation and excess kurtosis as inputs and mean return and skewness as outputs and Briec et al. (2007) choose variance as input and mean return and skewness as outputs. Also, Joro and Na (2006) in their mean-variance-skewness (MVS) framework use variance as input and mean return and skewness as outputs. Accordingly, we include variance as input and mean return and skewness as outputs (For more detail see, Joro and Na, 2006).

4.2 Data

A large random sample would be selected to train the network before using NNDEA for assessing all the intended DMUs. Sampling theory suggests that a random sample of the dataset with this size should be a good descriptor of the underlying production technology pertaining throughout the system. Thus, the network is an excellent source of information to understand the performance of the system as a whole in transforming inputs to outputs. Troutt et al. (1996) suggest that there should be at least ten observations in the selected sample in order to avoid the problem while undergo training. Obviously, these numbers should be more in the large-scaled dataset. We delivered our approach as DMU, using random sample consist of 1,150 MFs among 10,049 MFs in the USA. In order to let our sample be a good descriptor of the whole population, we select the samples based on star grouping in morning star database. Hence, we tried to proportionate the number of any star sample with the total number of MFs with the same number of stars. For each MF, monthly returns are taken out from 2006 to the end of 2009 from Morningstar Principia [according to Morningstar Funds (2009)]. Daily returns could be selected for more accuracy or seasonal returns for more speed. Monthly returns of each MF are used for calculation of mean, variance and skewness.

4.3 Model

In the previous studies, for evaluating MFs efficiency, different models of DEA have been used. Based on Murthi et al. (1997), Basso and Funari (2001), Anderson et al. (2004) and Nguyen-Thi-Thanh (2004) studies, the original DEA model invented by Charnes, Cooper, and Rhodes (1978) (known as CCR model) is the most popular method for evaluating MFs. One concern is the type of variables, as explained in Emrouznejad and Amin (2009) if there is a ratio variable in the assessment model, then DEA may produce incorrect results. To avoid such a problem either one has to use a non-linear model as proposed by Emrouznejad and Amin (2009) or to make sure that the DMUs in the DEA assessment are about the same size to avoid convexity problem. This has been considered in the selection of 1,150 MFs among 10,049 MFs in the USA.

4.4 Assessment procedure

Generally, before training begins in any neural network, the user must decide on the network topology by specifying the number of variables in the input and output layer, the number of hidden layers, and the number of neurons in any hidden layer. In the NNDEA
we use resources and outcomes in the corresponding DEA model as variables in the input layer and DEA efficiency amount as the only variable in the output layer.

The selection of number hidden layers in feed-forward neural network topology often requires an engineering opinion, because there is not any defined rule for determining the ‘best’ number of hidden layers. Generally, the concept of ‘the more, the better’ can use as a policy for determining hidden layer numbers, since number of hidden layers and numbers of neurons within them control the flexibility of mapping in the network. However, using large numbers of hidden layers may not be suitable (for more details, see Schalkoff, 1997).

Therefore, it can be said that the number of hidden layers, which influences on network correctness, is proportional to the kind of problem and trial and error process. Thus, primary values of weights influence on obtained correctness (error rate). When network is trained and its correctness is not so important, it is recommend repeating training process with different structure of neural network.

Determination of available neuron numbers in each layer is also very important. Gaborski (1990) indicated that training time is affected by hidden layers neuron numbers. Over increase in the hidden layer neurons, leads to significant increase in the training time, in return of it, only little level of mapping correctness can be improved. To explain the function of this algorithm, a set of data contains 10,049 MFs (DMUs) is considered. A randomised sample as big as 1,150 MF, is selected among above MFs for training and testing the function of NNDEA as explained above. Then this sample was divided into two sets of 600 and 550 MFs, respectively for training and testing NNDEA. Analysis performed in two stages:

1) training NNDEA using first set
2) testing NNDEA using the second set.

Further, we compared the efficiency of MFs obtained using NNDEA with the efficiency scores obtained from conventional DEA in the test set. Estimated efficiency amount by NNDEA have been calculated by MATLAB programs.

In the first stage, i.e., NNDEA training, three networks without hidden layer, with one hidden layer and with two hidden layers were examined. To evaluate these networks, using the first group data, these three networks were trained in order to obtain a better correctness in testing and in training each of these networks, by changing $E_{\text{max}}$, our aim was minimise the error. Then, by comparing the training time and correctness in these networks, a network with one hidden layer was recognised appropriate, because it had the least error and the shorter training time.

Afterward, in the second stage, the efficiency of MFs in the second group (test set) was assessed. This was done by multiplying the obtained weights of training network in the inputs and outputs of the DEA. On the other hand, for comparison study the efficiency of MFs in second group were obtained using the DEA method.

5 Discussion and results

Comparison of scores obtained by the two methods, conventional DEA and NNDEA, is illustrated in Figure 2.
In order to validate the NNDEA model we compared the results obtained from NNDEA with those of DEA. In this procedure, we use correctness and time as two key criteria for comparing the scores.

5.1 Correctness

Figure 2 illustrates a comparison of the efficiency scores as calculated by NNDEA and the actual efficiency scores as measured by the DEA model. For further analysis, we also calculated the error as the absolute difference between NNDEA and DEA scores which is shown in Figure 3. It can be seen that the NNDEA predictions for efficiency amount appear to be a good estimate for the majority of cases.
The first similarity criterion between obtained amounts from this model and the ones of conventional DEA model is distribution of the error estimation. As see in Figure 3, 86.7% estimated MFs efficiency have been associated with less than 0.02 error units compared to the real efficiency amounts from conventional DEA. Also, 92.9% estimated efficiency amounts have error less than 0.04 units of error.

The second criterion of similarity in the two approaches is in terms of the rank ordering of the DMUs by the two methods of the estimation. At first, this was tested by calculating Spearman’s $P$ that evaluates the linear correlation of the measures and criterion. This analysis was done in SPSS software and the results are given in Table 1. All correlations are at 0.01 confidence level (two-tail test).

One important point in using Spearman’s $\rho$ criterion on studying rank correlation is that this criterion only considers linear correlation of the ranks, while there may be some ranks with non-linear correlation. For solving this problem, we use Kendall’s $\rho$ criterion which evaluate both linear and non-linear correlations. The result is reported in Table 1. The two models have rank correlation of 99% confidence level. So, we can conclude that the rank orders of DMUs are very similar in both DEA and NNDEA models.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Comparison of NNDEA and conventional DEA: Spearman’s and Kendall’s correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NNDEA</td>
</tr>
<tr>
<td>Kendall’s tau_b</td>
<td>Correlation coefficient</td>
</tr>
<tr>
<td></td>
<td>Sig. (two-tailed)</td>
</tr>
<tr>
<td></td>
<td>N</td>
</tr>
<tr>
<td>DEA</td>
<td>Correlation coefficient</td>
</tr>
<tr>
<td></td>
<td>Sig. (two-tailed)</td>
</tr>
<tr>
<td></td>
<td>N</td>
</tr>
<tr>
<td>Spearman’s rho</td>
<td>Correlation coefficient</td>
</tr>
<tr>
<td></td>
<td>Sig. (two-tailed)</td>
</tr>
<tr>
<td></td>
<td>N</td>
</tr>
<tr>
<td>DEA</td>
<td>Correlation coefficient</td>
</tr>
<tr>
<td></td>
<td>Sig. (two-tailed)</td>
</tr>
<tr>
<td></td>
<td>N</td>
</tr>
</tbody>
</table>

Note: **Correlation is significant at the .01 level (two-tailed).

5.2 Time

As it is noted, we can obtain conventional DEA efficiency scores and estimated efficiency scores by NNDEA, through two distinctive programmes which are written in MATLAB software. Comparison of performing time of these two programmes can be a good criterion for comparing these two models from the point of speed. The time comparison is reported in Table 2. These results indicate the considerable time reduction in NNDEA in comparison to DEA. It is clear from this table that the solving time for NNDEA to estimate efficiency of 10,049 MFs is far less than corresponding conventional DEA model.
Table 2  

<table>
<thead>
<tr>
<th>Method</th>
<th>Time(s)</th>
<th>Computer</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEA</td>
<td>128.0588</td>
<td>Intell Pentum III Processor</td>
<td>MATLAB</td>
</tr>
<tr>
<td>NNDEA</td>
<td>0.0886</td>
<td>797 MHZ</td>
<td>240 MB of RAM</td>
</tr>
</tbody>
</table>

5.3 Empirical analysis

In this section, we focus only on a selected set of MFs. Usually, investors are looking to invest in the most efficient MFs, hence we selected ten MFs with the highest efficiency score for details evaluation, from the ones which were obtained out of NNDEA. This set of MFs is shown in Table 3 with both efficiency scores and ranks obtained in NNDEA and conventional DEA.

Table 3  

<table>
<thead>
<tr>
<th>Fund</th>
<th>NNDEA</th>
<th>NNDEA-R</th>
<th>DEA</th>
<th>DEA-R</th>
<th>ERROR</th>
<th>A&amp;P</th>
<th>A&amp;P-R</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMUAX</td>
<td>0.9938</td>
<td>1</td>
<td>1</td>
<td>1–2</td>
<td>0.0062</td>
<td>2.3195</td>
<td>1</td>
</tr>
<tr>
<td>DFIHX</td>
<td>0.9132</td>
<td>2</td>
<td>1</td>
<td>1–2</td>
<td>0.0368</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>JASBX</td>
<td>0.4183</td>
<td>3</td>
<td>0.3409</td>
<td>3</td>
<td>0.0774</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>PTSHX</td>
<td>0.3091</td>
<td>5</td>
<td>0.2563</td>
<td>4</td>
<td>0.0528</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>STSMX</td>
<td>0.2885</td>
<td>4</td>
<td>0.2407</td>
<td>5</td>
<td>0.0478</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>AIBAX</td>
<td>0.2659</td>
<td>7</td>
<td>0.2399</td>
<td>6</td>
<td>0.026</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>VCAIX</td>
<td>0.2597</td>
<td>8</td>
<td>0.2286</td>
<td>7</td>
<td>0.0311</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>PRWIBX</td>
<td>0.2776</td>
<td>6</td>
<td>0.2181</td>
<td>8</td>
<td>0.0595</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>FLT MX</td>
<td>0.1873</td>
<td>11</td>
<td>0.202</td>
<td>9</td>
<td>0.0147</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>SCMTX</td>
<td>0.1725</td>
<td>12</td>
<td>0.1852</td>
<td>10</td>
<td>0.0127</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Note: The rankings with two numbers mean that two efficiencies from MFs are equal and have the same ranking.

Our first observation is that the conventional DEA does not distinguish between the first two MFs, as it assigns efficiency score of 1 to both SMUAX and DFIHX. In order rank these two efficient MFs one possibility is to use super efficiency model of Andersen and Peterson (1993) (A&P). This was done and ranking is added in the last column of Table 3 which is matched with the ranking given by NNDEA.

It is clear that three out of ten MFs have the same rank in both NNDEA and DEA.

While in the whole data set the error is low enough (92.9% lower than 0.04 error) to use NNDEA, but in the selected set the error is even much lower with less than 1% overall, see column 2 of Table 3.

6 Conclusions

DEA is non-parametric method that is used for measuring relative efficiency of the MFs. In measuring efficiency of MFs speed is an important issue for investors who want to get the results less than a second. However the standard DEA needs to solve one linear
programming for each MFs, hence it may takes some time to obtained the results of large datasets even with fast computers. This paper proposed a NNDEA model for such problem which requires far less time than standard DEA to obtain the same results.

Applicability of the proposed model is examined for evaluating efficiency of large set of MFs in USA. After applying the proposed algorithm, the validity of the model was discussed using distribution errors and correlation of the two efficiency scores. The results show that NNDEA estimates MFs’ relative efficiency with high accuracy and it is very fast compared to conventional DEA method.

The selection of variables, determination of the number of hidden layers and neurons in NNDEA are subjects of future studies.

References
Neural network DEA for measuring the efficiency of mutual funds


---

**Appendix**

Back propagation DEA algorithm (neural network with one hidden layer shown in Figure 1).

<table>
<thead>
<tr>
<th>Training sample (net input)</th>
<th>Desirable output (DEA output)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x^1)</td>
<td>(d^1)</td>
</tr>
<tr>
<td>(x^2)</td>
<td>(d^2)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>(x^R)</td>
<td>(d^R)</td>
</tr>
</tbody>
</table>
Neural network DEA for measuring the efficiency of mutual funds

Step 1  Initialise learning rate $C > 0$, maximum level of error criterion function $E_{\text{max}}$, initial amounts of weights of $V, W$ randomly about zero. Set $E = 0$, $r = 1$, $q = 1$ that $r$: training sample counter, $q$: training period counter.

Step 2  Give the training sample to the network ($d \rightarrow d', x \rightarrow x'$). Then calculate these values and compare them with related threshold and then calculate output values of each layer.

\[ Y_j = x^{T}.v_j^{i} \]
\[ y_j = f(Y_j) \]
\[ Z = y^{T}.w \]
\[ z = f(Z) \]

$v^{i}$ is $i^{th}$ column vector of matrix $V$. In other words, weights lead to hidden layer of $i^{th}$ neuron.

Step 3  After calculating the network’s output ($z$) for the given sample, compute output error.

\[ E \rightarrow E + \frac{1}{2} \sum_{k=1}^{l} (d - z)^2 \]

Step 4  Calculate error signals of $\delta_z, \delta_Z$

\[ \delta_z = z(1-z)(d-z) \]
\[ \delta_{y_j} = y_j \left(1 - y_j \right) \left( \delta_z.w_j \right); \ j = 1, 2, 3, \ldots, n \]

Step 5  Correcting the weights of output layer

\[ w_j \rightarrow w_j + C.\delta_Z.y_j; \ j = 1, 2, 3, \ldots, n \]

Step 6  Correcting the weights of hidden layer

\[ v_{y_j} \rightarrow v_{y_j} + C.\delta_x.x_i; \ i = 1, 2, 3, \ldots, m, \ j = 1, 2, 3, \ldots, n \]

Step 7  If $r < R$, then add one nit to $r$ and go to 2, else go to 8.

Step 8  If $E < E_{\text{max}}$, training period is completed and show $V, W, q, E$ in output, but if, $E < E_{\text{max}}$ set $E = 0$, $r = 1$, $q = q + 1$ and start new training period again from step 2.