Review of Economic Sciences, 6, TEI of Epirus, pp. 161-176

THE USE OF NEURAL NETWORKS IN FORECASTING

Dimitrios Maditinos
Applications Professor

Prodromos Chatzoglou
Associate Professor

ABSTRACT

Finance and investing are one of the most frequent areas of neural network (NN) applications. Some of the most representative problems being solved by NNs are bankruptcy predictions, risk assessments of mortgage and other loans, stock market predictions (stock, bond, and option prices, capital returns, commodity trade, etc.), financial prognoses (returns on investments) and others. Chase Manhattan Bank, Peat Marwick, American Express are only a few of many companies that efficiently apply NNs in solving their financial and investing problems. The objective of this paper is to provide a review of the literature on NNs applied to finance problems, focusing mainly on the modelling process.

ΠΕΡΙΛΗΨΗ

Η χρηματοοικονομική και οι επενδύσεις αποτελούν τις συχνότερες περιοχές με τις περισσότερες εφαρμογές των Νευρωνικών Δικτύων (ΝΔ). Μερικά από τα πιο αντιπροσωπευτικά προβλήματα που λύνονται με την υιοθέτηση των ΝΔ είναι οι προβλέψεις πτωχεύσεων των επιχειρήσεων, η αξιολόγηση του κινδύνου διαφόρων ειδών δανείων, οι προβλέψεις στο χρηματιστήριο (τιμές μετοχών, ομολογιακών δανείων, προθεσμιακών συμβολαίων, κεφαλαιακές αποδόσεις, τιμές εμπορευμάτων, κλπ), και η αξιολόγηση των επενδύσεων. Η τράπεζα Chase Manhattan, η εταιρία ορκωτών ελεγκτών Peat Marwick, και η εταιρία American Express είναι μόνο λίγες από τις επιχειρήσεις που χρησιμοποιούν, με μεγάλη αποτελεσματικότητα, τα ΝΔ για την λύση διαφόρων χρηματοοικονομικών προβλημάτων που αντιμετωπίζουν καθημερινά. Ο σκοπός του άρθρου αυτού είναι η βιβλιογραφική αναφορά των εφαρμογών των ΝΔ στους τομείς της χρηματοοικονομικής, κυρίως στο θέμα των προβλέψεων, και η παρουσίαση της διαδικασίας μοντελοποίησης με την χρήση των ΝΔ.

JEL Classification (C00)
Key Words: Neural networks, forecasting, finance.

Review of Economic Sciences, 6, TEI of Epirus, pp. 161-176

THE USE OF NEURAL NETWORKS IN FORECASTING

Dimitrios Maditinos
TEI of Kavala
Department of Business Administration
Agios Loukas-65404 Kavala
Tel.:2510-462219
Dmadi@teikav.edu.gr

Prodromos Chatzoglou
Democritus University of Thrace
Department of Management and Production Engineering
Kimmeria – 67 100 Xanthi
2510-462299
pdchatz@yahoo.com

1. Introduction

Numerous research and applications of NNs in business have proven their advantage in relation to classical methods that do not include artificial intelligence. According to Wong *et al.* (1995), the most frequent areas of NN applications in past 10 years are production/operations (53.5%) and finance (25.4%).

Predicting the future behaviour of real world time series using NNs has been extensively investigated (e.g., Chakraborty *et. al.,*1992; Theriou and Tsirigotis, 2000) because neural networks can learn nonlinear relationships between inputs and desired outputs. Integration of knowledge and NNs has also been extensively investigated, because such integration holds great promise in solving complicated real-world problems. One method is to insert prior knowledge into the initial network structure and refine it with learning by examples (Giles and Omlin, 1993). Another method is to represent prior knowledge in the form of error measurements for training neural networks (Abu-Mostafa, 1993).

In the past few years, many researchers have used ANNs to analyse traditional classification and prediction problems in accounting and finance.

Numerous articles have appeared recently that surveyed journal articles on ANNs applied to business situations. Wong *et al.* (1997) surveyed 203 articles from 1988 through 1995. They classified the articles by year of publication, application area (accounting, finance, etc.), journal, various decision characteristics means of development, integration with other technologies, comparative technique (discriminant analysis, regression analysis, logit and IDS), and major contribution. The survey included five articles in accounting and auditing, and 54 articles in finance

O'Leary (1998) analysed 15 articles that applied ANNs to predict corporate failure or bankruptcy. For each study, he provided information about the data, the ANN model and software (means of development), the structure of the ANN (input, hidden and output layers) training and testing, and the alternative parametric methods used as a benchmark.

Zhang *et al.* (19981 surveyed 21 articles that addressed modelling issues when ANNs are applied for forecasting, and an additional 11 studies that compared the relative performance of ANNs with traditional statistical methods. For the modelling issues, they addressed the type of data, size of the training and test samples, architecture of the model (number of nodes in each layer and transfer function), training algorithm used, and the method of data normalization.

Vellido *et* a/. (1999), surveyed 123 articles from 1992 through 1998. They included 8 articles in accounting and auditing, and 44 articles in finance (23 on bankruptcy prediction, 11 on credit evaluation, and 10 in other areas). They provided information on the ANN model applied, the method used to validate training of the model, the sample size and number of decision variables, the comparative parametric / linear technique used as a benchmark, and main contribution of the article.

Analytically, we could say that there is an extensive literature in financial applications of ANNs (Trippi and Turban, 1993; Azoff, 1994; Refenes, 1995; Gately, 1996). ANNs have been used for forecasting bankruptcy and business failure (Odom and Sharda, 1990; Coleman et al., 1991; Salchenkerger et al., 1992; Wilson and Sharda, 1994), foreign exchange rate (Weigend et al., 1992; Refenes, 1993; Borisov and Pavlov, 1995; Hann and Steurer, 1996), stock prices (White, 1988; Kimoto et al., 1990; Bergerson and Wunsch, 1991; Grudnitski and Osburn, 1993), and others (Dutta and Shekhar, 1988; 1993; Refenes et al., 1994; Kaastra and Boyd, 1995; Chiang et al., 1996; Kohzadi et al., 1996; Theriou and Tsirigotis, 2000).
The objective of this paper is to provide a review of the literature on ANNs applied to finance problems, focusing on the modelling issues. It is more like a tutorial on modelling issues than a critical analysis. The second section will review the basic foundation of ANNs to provide a common basis for further elaboration. For a more detailed description of ANNs, we refer the reader to numerous other articles that provide insights into various networks (Anderson and Rosenfeld, 1988; Hecht-Nielsen, 1990; Hertz *et al,* 1991; Hoptroff *et al,* 1991; Rumelhart and McClelland, 1986; Wasserman, 1989).
The third section of the paper discusses the development of ANN modelling process.

2. The basic foundation of NN

ANNs are structures of highly interconnected elementary computational units. They are called *neural* because the model of the nervous systems of animals inspired them. Each computational unit (see Figure 1) has a set of input connections that receive signals from other computational units and a bias adjustment, a set of weights for each input connection and bias adjustment, and a transfer function that transforms the sum of the weighted inputs and bias to decide the value of the output from the computational unit. The sum value for the computational unit (node *j)* is the linear combination of all signals from each connection *(A_i)* times the value of the connection weight between node *j* and connection *i* ($W_{j\ i}$) (equation (1)). Note that equation (1) is similar to the equation form of multiple regression: Y' = BO + Σi [Bi * Xi]. The output for node *j* is the result of applying a transfer function *g* (equation (2)) to the sum value (Sum_i)

$$Sum_j = \Sigma_i [W_{ji} * A_i] \qquad (1)$$

$$O_j = g(Sum_j) \qquad (2)$$

If the transfer function applied in equation (2) is linear, then the computational unit resembles the multiple regression model. If the transfer function applied in equation (2) is the sigmoid, then the computational unit resembles the logistic regression model. The only difference between the ANN and regression models is the manner in which the values for the weights are established. ANNs employ a dynamic programming approach to iteratively adjust the weights until the error is minimized while the

regression models compute the weights using a mathematical technique that minimizes the squared error.
Most ANNs applied in the literature are actually a network of these computational units (hereafter referred to as nodes) interconnected to function as a collective system.

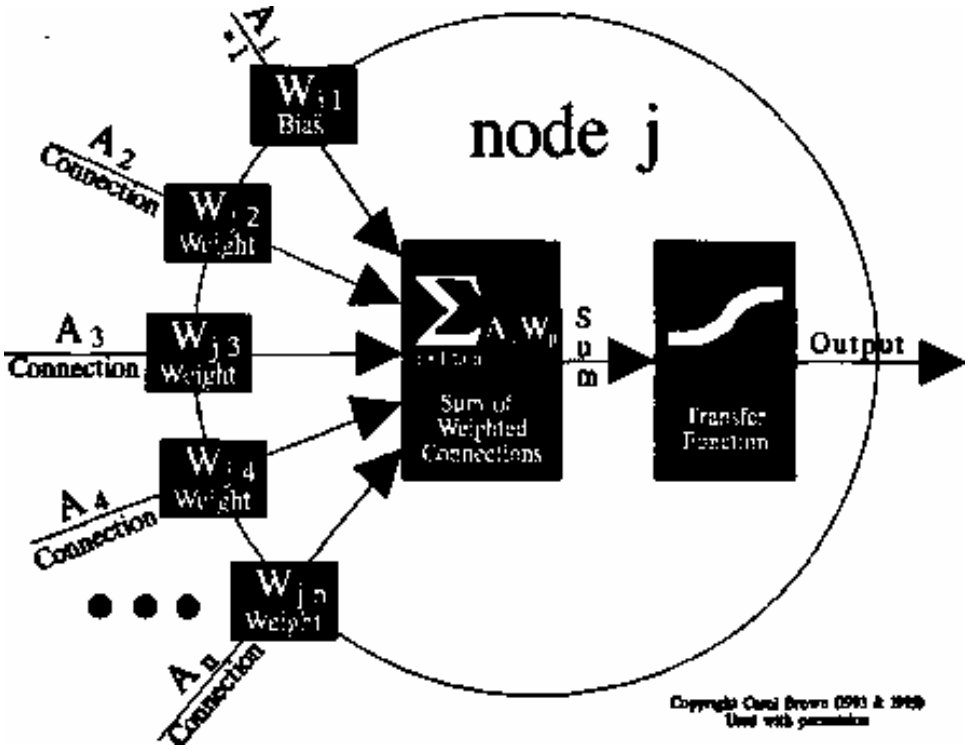

Figure 1: Structure of a computational unit (node y)
(Coakley and Brown, 2000)

The *architecture* of the network defines how the nodes in a network are interconnected. A multi-layer, feed-forward architecture is depicted in Figure 2. The nodes are organized into a series of layers with an input layer, one or more hidden layers, and an output layer. Data flows through this network in one direction only, from the input layer to the output layer.

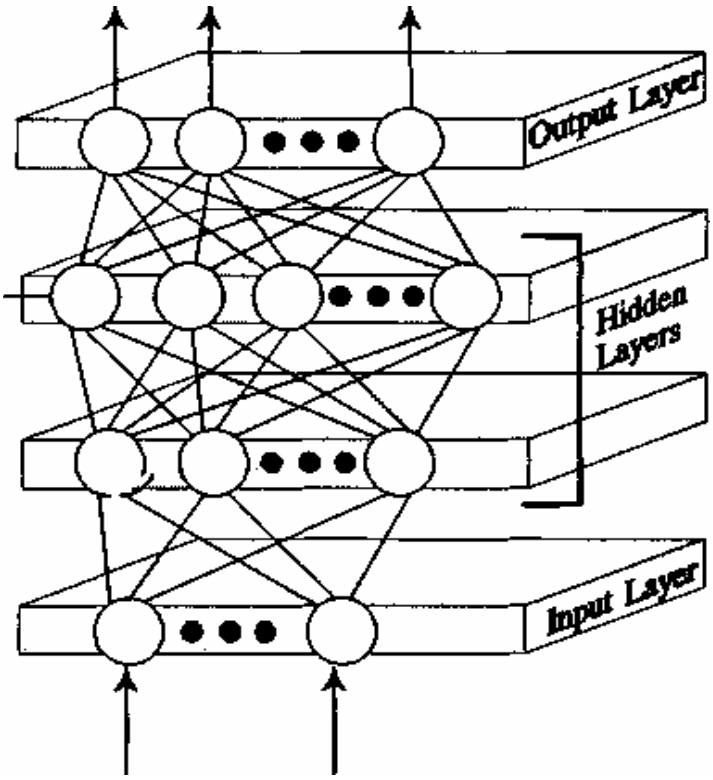

Figure 2: Feed-forward neural network structure with two hidden layers (Coakley and Brown, 2000)

Before an ANN can be used to perform any desired task, it must be trained to do so. Basically, training is the process of determining the arc weights, which are the key elements of an ANN. The knowledge learned by a network is stored in the arcs and nodes in the form of arc weights and node biases. It is through the linking arcs that an ANN can carry out complex nonlinear mappings from its input nodes to its output nodes. A multiplayer network's training is a supervised one in that the desired response of the network (target value) for each input pattern (example) is always available.

The training input data is in the form of vectors of input variables or training patterns. Corresponding to each element in an input vector is an input node in the network input layer. Hence the number of input nodes is equal to the dimension of input vectors. For a causal forecasting problem, the number of input nodes is well defined and it is the number of independent variables associated with the problem. For a time series' forecasting problem, however, the appropriate number of input nodes is not easy to determine. Whatever the dimension, the input vector for a time series forecasting problem will be almost always composed of a moving window of fixed length along the series. The total available data is usually divided into a training set (in-sample data) and a test set (out-of-sample or hold-out sample). The training set is used for estimating the arc weights while the test set is used for measuring the generalization ability of the network.

The training process is usually as follows. First, examples of the training set

are entered into the input nodes. The activation values of the input nodes are weighted and accumulated at each node in the first hidden layer. The total is then transformed by an activation function into the node's activation value. It in turn becomes an input into the nodes in the next layer, until eventually the output activation values are found. The training algorithm is used to find the weights that minimize some overall error measure such as the sum of squared errors (SSE) or mean squared errors (MSE). Hence the network training is actually an unconstrained nonlinear minimization problem.

3. Issues in ANN modelling for forecasting

Despite the many satisfactory characteristics of ANNs, building a neural network forecaster for a particular forecasting problem is a nontrivial task. Modelling issues that affect the performance of an ANN must be considered carefully. One critical decision is to determine the appropriate architecture, that is, the number of layers, the number of nodes in each layer, and the number of arcs, which interconnect with the nodes. Other network design decisions include the selection of activation functions of the hidden and output nodes, the training algorithm, data transformation or normalization methods, training and test sets, and performance measures.
In this section we survey the above-mentioned modelling issues of a neural network forecaster. Since the majority of researchers use exclusively fully-connected-feedforward networks, we will focus on issues of constructing this type of ANNs.

3.1. The network architecture

An ANN is typically composed of layers of nodes. In the popular multi-layer models, all the input nodes are in one input layer, all the output nodes are in one output layer and the hidden nodes are distributed into one or more hidden layers in between. In designing such a model, one must determine the following variables:

- the number of input nodes.
- the number of hidden layers and hidden nodes.
- the number of output nodes.

The selection of these parameters is basically problem-dependent. Although there exists many different approaches such as the pruning algorithm (Sietsma and Dow, 1988; Karnin, 1990; Weigend et al., 1991; Reed, 1993; Cottrell et al., 1995), the polynomial time algorithm (Roy et al., 1993), the canonical decomposition technique (Wang et al., 1994), and the network information criterion (Murata et al., 1994) for finding the optimal architecture of an ANN, these methods are usually quite complex in nature and are difficult to implement. Furthermore none of these methods can guarantee the optimal solution for all real forecasting problems. To date, there is no simple clear-cut method for determination of these parameters.

Guidelines are either heuristic or based on simulations derived from limited experiments. Hence the design of an ANN is more of an art than a science.

3.1.1. *The number of hidden layers and nodes*

The hidden layer and nodes play very important roles for many successful applications of neural networks. It is the hidden nodes in the hidden layer that allow neural networks to detect the feature, to capture the pattern in the data, and to perform complicated nonlinear mapping between input and output variables. Theoretical work has shown that a hidden layer is sufficient for ANNs to approximate any nonlinear function with any desired accuracy (Cybenko, 1989; Hornik et al., 1989). Thus, most authors use only one hidden layer for forecasting purposes. Two hidden layer networks may provide more benefits for some type of problems (Barron, 1994). Several authors address this problem and consider more than one hidden layer (usually two hidden layers) in their network design processes. Srinivasan et al. (1994) use two hidden layers and this results in a more compact architecture, which achieves a higher efficiency in the training process than one hidden layer networks. Some authors simply adopt two hidden layers in their network modelling without comparing them to the one hidden layer networks (Vishwakarma, 1994; Grudnitski and Osburn, 1993; Lee and Jhee, 1994). The issue of determining the optimal number of hidden nodes is a crucial yet complicated one. In general, networks with fewer hidden nodes are preferable as they usually have better generalization ability and less overrating problem. But networks with too few hidden nodes may not have enough power to model and learn the data. There is no theoretical basis for selecting this parameter although a few systematic approaches are reported. For example, both methods for pruning out unnecessary hidden nodes and adding hidden nodes to improve network performance have been suggested. Gorr et al. (1994) propose a grid search method to determine the optimal number of hidden nodes.

The most common way in determining the number of hidden nodes is via experiments or by trial-and-error. Several rules of thumb have also been proposed, such as, the number of hidden nodes depends on the number of input patterns and each weight should have at least ten input patterns (sample size). To help avoid the overfitting problem, some researchers have provided empirical rules to restrict the number of hidden nodes. Lachtermacher and Fuller (1995) give a heuristic constraint on the number of hidden nodes. In the case of the popular one hidden layer networks, several practical guidelines exist. These include using *"2n +1"* (Lippmann, 1987; Hecht-Nielsen, 1990), *"2n"* (Wong, 1991), *"n"* (Tang and Fishwick, 1993), "n/2" (Kang, 1991), where *n* is the number of input nodes. However none of these heuristic choices works well for all problems.

3.1.2. The number of input nodes

The number of input nodes corresponds to the number of variables in the input vector used to forecast future values. For causal forecasting, the number of inputs is usually transparent and relatively easy to choose. In a

time series forecasting problem, the number of input nodes corresponds to the number of lagged observations used to discover the underlying pattern in a time series and to make forecasts for future values. However, currently there is no suggested systematic way to determine this number. Recently, genetic algorithms have received considerable attention in the optimal design of a neural network (Miller et al., 1989; Guo and Uhrig, 1992; Jones, 1993; Schiffmann et al., 1993). Genetic algorithms are optimisation procedures which can mimic natural selection and biological evolution to achieve more efficient ANN learning process (Happel and Murre, 1994). Due to their unique properties, genetic algorithms are often implemented in commercial ANN software packages.

3.1.3. The number of output nodes

The number of output nodes is relatively easy to specify as it is directly related to the problem under study. For a time series forecasting problem, the number of output nodes often corresponds to the forecasting horizon. There are two types of forecasting: one-step-ahead (which uses one output node) and multi-step-ahead forecasting. Two ways of making multi-step forecasts are reported in the literature. The first is called the iterative forecasting as used in the Box-Jenkins model in which the forecast values are iteratively used as inputs for the next forecasts. In this case, only one output node is necessary. The second called the direct method is to let the neural network have several output nodes to directly forecast each step into the future.

3.2. The activation function

This function determines the relationship between inputs and outputs of a node and a network. In general, the activation function introduces a degree of nonlinearity that is valuable for most ANN applications. Chen and Chen (1995) identify general conditions for a continuous function to qualify as an activation function. Loosely speaking, any differentiable function can qualify as an activation function in theory. In practice, only a small number of activation functions are used. These include:

1. The sigmoid (logistic) function:

$$f(x)=(1+\exp(-x))^{-1};$$

2. The hyperbolic tangent (tanh) function:

$$f(x) = (\exp(x) - \exp(-x))/(\exp(x) + \exp(-x));$$

3. The sine or cosine function:

$f(x) = sin(x)$ or $f(x) = cos(x)$;

4. The linear function: $f(x) = x$.

Among them, logistic transfer function is the most popular choice.

3.3. Training algorithm

The neural network training is an unconstrained nonlinear minimization problem in which arc weights of a network are iteratively modified to minimize the overall mean or total squared error between the desired and actual output values for all output nodes over all input patterns. The existence of many different optimisation methods (Fletcher, 1987) provides various choices for neural network training. There is no algorithm currently available to guarantee the global optimal solution for a general nonlinear optimisation problem in a reasonable amount of time. The most popularly used training method is the **back propagation algorithm**. A back propagation NN uses a feedforward topology, supervised learning, and the back propagation algorithm (Rumelhart, Hinton, and Williams, 1986). **Recurrent back propagation** is a network with feedback or recurrent connections. By adding recurrent connections to a back propagation network enhances its ability to learn temporal sequences without fundamentally changing the training process, thus, in general, performs better than the regular back propagation network on time-series problems.

3.4. Scaling and Data normalization

Another transformation involves the more general issue of scaling data for presentation to the neural network. Most neural network models accept numeric data only in the range of 0.0 to 1.0 or -1.0 to +1.0, depending on the activation functions used in the neural processing elements. Consequently, data usually must be scaled down to that range.
Scalar values that are more or less uniformly distributed over a range can be scaled directly to the 0 to 1.0 range. If the data values are skewed, a piece-wise linear or a logarithmic function can be used to transform the data, which can then be scaled into the desired range. Discrete variables can be represented by coded types with 0 and 1 values, or they can be assigned values in the desired continuous range.

Vectors or arrays of numeric data can sometimes be treated as groups of numbers. In these cases, we might need to normalize or scale the vectors as a

group. There are several ways of doing this. Perhaps the most common vector normalization method is to sum the squares of each element, take the square root of the sum, and then divide each element by the norm. This is called the Euclidean norm. A second way to normalize vector data is to simply sum up all of the elements in the vector and then divide each number by the sum. In this way, the normalized elements sum to 1.0, and each takes on a value representing the percentage of contribution they make. A third way to normalize vector data is to divide each vector element by the maximum value in the array. Data normalization is often performed before the training process begins. As mentioned earlier, when nonlinear transfer functions are used at the output nodes, the desired output values must be transformed to the range of the actual outputs of the network. Even if a linear output transfer function is used, it may still be advantageous to standardize the outputs as well as the inputs to avoid computational problems (Lapedes and Farber, 1988), to meet algorithm requirement (Sharda and Patil, 1992), and to facilitate network learning (Srinivasan et al., 1994). Normalization of the output values is usually independent of the normalization of the inputs. Only for the time series forecasting problems, the normalization of inputs is typically performed together with the outputs. It should be noted that, as a result of normalizing the output values, the observed output of the network will correspond to the normalized range. Thus, to interpret the results obtained from the network, the output must be rescaled to the original range.

3.5. Training sample and test sample

As we mentioned earlier, a training and a test sample are typically required for building an ANN forecaster. The training sample is used for ANN model development and the test sample is adopted for evaluating the forecasting ability of the model. Sometimes a third one called the validation sample is also utilized to avoid the overfilling problem or to determine the stopping point of the training process (Weigend et al., 1992). It is common to use one test set for both validation and testing purposes particularly with small data sets

The first issue here is the division of the data into the training and test sets. Although there is no general solution to this problem, several factors such as the problem characteristics, the data type and the size of the available data should be considered in making the decision. It is critical to have both the training and test sets representative of the population or underlying mechanism. This has particular importance for time series forecasting problems. The literature offers little guidance in selecting the training and the test sample. Most authors select them based on the rule of 90% vs. 10%, 80% vs. 20% or 70% vs. 30%, etc. Granger (1993) suggests that for nonlinear forecasting models, at least 20 percent of any sample should be held back for the final evaluation (testing) of the forecasting results.

Another closely related factor is the sample size. No definite rule exists for the requirement of the sample size for a given problem. The amount of data

for the network training depends on the network structure, the training method, and the complexity of the particular problem or the amount of noise in the data on hand. In general, as in any statistical approach, the sample size is closely related to the required accuracy of the problem. The larger the sample size, the more accurate the results will be. Nam and Schaefer (1995) test the effect of different training sample size and find that as the training sample size increases, the ANN forecaster performs better. Kang (1991) finds that ANN forecasting models perform quite well even with sample sizes less than 50 while the Box-Jenkins models typically require at least 50 data points in order to forecast successfully.

3.6. Performance measures

Although there can be many performance measures for an ANN forecaster like the modelling time and training time, the ultimate and the most important measure of performance is the prediction accuracy it can achieve beyond the training data. However, a suitable measure of accuracy for a given problem is not universally accepted by the forecasting academicians and practitioners. An accuracy measure is often defined in terms of the forecasting error which is the difference between the actual (desired) and the predicted value. There are a number of measures of accuracy in the forecasting literature and each has advantages and limitations (Makridakis et al., 1983). The most frequently used are

- the mean absolute deviation (MAD)
- the sum of squared error (SSE)
- the mean squared error (MSE)
- the root mean squared error (RMSE)
- the mean absolute percentage error (MAPE).

4. Conclusions

We have presented a review of the current state of the use of artificial neural networks for forecasting application. This review is comprehensive but by no means exhaustive, given the fast growing nature of the literature. The important findings are summarized as follows:

- The unique characteristics of ANNs - adaptability, nonlinearity, arbitrary function mapping ability - make them quite suitable and useful for forecasting tasks. Overall, ANNs give satisfactory performance in forecasting.
- A considerable amount of research has been done in this area. The findings are inconclusive as to whether and when ANNs are better than classical methods.
- There are many factors that can affect the performance of ANNs.

However, there are no systematic investigations of these issues. The shot-gun (trial-and-error) methodology for specific problems is typically adopted by most researchers, which is the primary reason for inconsistencies in the literature.

ANNs offer a promising alternative approach to traditional linear methods. However, while ANNs provide a great deal of promises, they also embody a large degree of uncertainty. Like statistical models, ANNs have weaknesses as well as strengths. While ANNs have many desired features, which make them quite suitable for a variety of problem areas, they will never be the panacea.

5. References

Abu-Mostafa, Y., (1993), "A method for learning from hints", in Hanson, S. *et al.* (eds), *Advances in Neural Information Processing Systems*, **5**, San Mateo, CA: Morgan Kaufmann.

Anderson JA, Rosenfeld E. 1988. *Neurocomputing: Foundations of Research.* MIT Press: Cambridge, MA.

Azoff, E.M., 1994. Neural Network Time Series Forecasting of Financial Markets. John Wiley and Sons, Chichester.

Barron, A.R., 1994. A comment on "Neural networks: A review from a statistical perspective". Statistical Science 9(1), 33-35.

Bergerson, K., Wunsch, D.C., 1991. A commodity trading model based on a neural network-expert system hybrid. In: Proceedings of the IEEE International Conference on Neural Networks, Seattle, WA, pp. 1289-1293.

Bigus, J. P., (1996), *Data Mining with Neural Networks*, New York: McGraw-Hill.

Borisov, A.N., Pavlov, V.A., 1995. Prediction of a continuous function with the aid of neural networks. Automatic Control and Computer Sciences 29 (5), 39-50.

Chakraborty, K., Mehrotra, K.,Mohan, C. and Ranka, S., (1992), "Forecasting the behaviour of multivariate time series using neural networks", *Neural Networks*, **5** : 961-70.

Chen, C.H., 1994. Neural networks for financial market prediction. In: Proceedings of the IEEE International Conference on Neural Networks, 2, pp. 1199-1202.

Chen, T., Chen, H., 1995. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. IEEE Transactions on Neural Networks 6 (4), 911-917.

Chiang, W.-C., Urban, T.L., Baldridge, G.W., 1996. A neural network approach to mutual fund net asset value forecasting. Omega 24, 205-215.

Coakley, J. R. and Brown, C. E., (2000), "Artificial Neural Networks in Accounting and Finance: Modelling Issues", *Intelligent Systems in Accounting, Finance and Management*, **9** (2): 119-144.

Coleman, K.G., Graettinger, T.J., Lawrence, W.F., 1991. Neural networks for bankruptcy prediction: The power to solve financial problems. AI Review 5, July/August, 48-50.

Cottrell, M., Girard, B., Girard, Y, Mangeas, M., Muller, C., 1995. Neural modeling for time series: a statistical stepwise method for weight elimination. IEEE Transactions on Neural Networks 6 (6), 1355-1364.

Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. Mathematical Control Signals Systems 2, 303-314.

Cybenko, G., (1988), " Continuous valued neural networks with two hidden layers are sufficient", *Technical Report*, Department of Computer Science, Tufts University, Medford, MA.

Dutta, S., Shekhar, S., 1988. Bond rating: A non-conservative application of neural networks. In: Proceedings of the IEEE International Conference on Neural Networks. San Diego, California, 2, pp. 443-450.

Fletcher, R., 1987. Practical Methods of Optimization, 2nd ed. John Wiley, Chichester.

Gately, E., 1996. Neural Networks for Financial Forecasting. John Wiley, New York.

Giles, C. and Omlin, C., (1993), "Rule refinement with recurrent neural networks", *Proceedings of International Conference on Neural Networks*, San Francisco, pp. 801-6.

Gorr, W.L., Nagin, D., Szczypula, J., 1994. Comparative study of artificial neural network and statistical models for predicting student grade point averages. International Journal of Forecasting 10, 17-34.

Granger, C.W.J., 1993. Strategies for modelling nonlinear time-series relationships. The Economic Record 69 (206), 233-238.

Grudnitski, G., Osburn, L., 1993. Forecasting S and P and gold futures prices: An application of neural networks. The Journal of Futures Markets 13 (6), 631-643.

Guo, Z., Uhrig, R., 1992. Using genetic algorithm to select inputs for neural networks. In: Proceedings of the Workshop on Combinations of Genetic Algorithms and Neural Networks, COGANN92, pp. 223-234.

Hann, T.H., Steurer, E., 1996. Much ado about nothing? Exchange rate forecasting: Neural networks vs. linear models using monthly and weekly data. Neurocomputing 10, 323-339.

Happel, B.L.M., Murre, J.M.J., 1994. The design and evolution of modular neural network architectures. Neural Networks 7, 985-1004.

Hecht-Nielsen, R., 1990. Neurocomputing. Addison-Wesley, Menlo Park, CA.

Hertz J, Krogh A, Palmer RG. 1991. *Introduction to the Theory of Neural Computation.* Addison-Wesley: Reading, MA.

Hoptroff R, Hall T, Bramson MJ. 1991. Forecasting economic turning points with neural nets. *Proceedings of the International Joint Conference on Neural Networks.* IEEE Service Center: Piscataway, NJ, II.347-II.352.

Hornik, K., Stinchcombe, M, White, H., 1989. Multilayer feedforward networks are universal approximators. Neural Networks 2, 359-366.

Jones, A.J., 1993. Genetic algorithms and their applications to the design of neural networks. Neural Computing and Applications 1, 32-45.

Kaastra, 1., Boyd, M.S., 1995. Forecasting futures trading volume using neural networks. The Journal of Futures Markets 15 (8), 953-970.

Kang, S., 1991. An Investigation of the Use of Feedforward Neural Networks for Forecasting. Ph.D, Thesis, Kent State University.

Karnin, E.D., 1990. A simple procedure for pruning back-propagation trained neural networks. IEEE Transactions on Neural Networks 1 (2), 239-245.

Kimoto, T., Asakawa, K., Yoda, M., Takeoka, M., 1990. Stock Market prediction system with modular neural networks. In: Proceedings of the IEEE International Joint Conference on Neural Networks. San Diego, California, 2, pp. 11-16.

Kohzadi, N., Boyd, M.S., Kermanshahi, B., Kaastra, I., 1996. A comparison of artificial neural network and time series models for forecasting commodity prices. Neurocomputing 10, 169-181.

Lachtermacher, G., Fuller, J.D., 1995. Backpropagation in time-series forecasting. Journal of Forecasting 14, 381-393.

Lapedes, A., Farber, R., 1988. How neural nets work. In: Anderson, D.Z., (Ed.), Neural Information Processing Systems, American Institute of Physics, New York, pp. 442-456.

Lapedes A, Farber R. 1987. *Nonlinear Signal Processing Using Neural Networks: Prediction and System Modeling.* Los Almos National Laboratory Report LA-UR-87-2662.

Lawrence J. 1991. *Introduction to Neural Networks.* California Scientific Software: Grass Valley, CA.

Lee, J.K., Jhee, W.C., 1994. A two-stage neural network approach for ARMA model identification with ESACF. Decision Support Systems 11, 461-479.

Lippmann, R.P., 1987. An introduction to computing with neural nets, IEEE ASSP Magazine, April, 4-22.

Makridakis, S., Wheelwright, S.C., McGee, V.E., 1983. Forecasting: Methods and Applications, 2nd ed. John Wiley, New York.

Miller, G.F., Todd, P.M., Hegde, S.U., 1989. Designing neural networks using genetic algorithms. In: Schaffer, J.D. (Ed.), Proceedings of the Third

International Conference on Genetic Algorithms. Morgon Kaufman, San Francisco, pp. 370-384.

Murata, N., Yoshizawa, S., Amari, S., 1994. Network information criterion-determining the number of hidden units for an artificial neural network model. IEEE Transactions on Neural Networks 5 (6), 865-872.

Nam, K., Schaefer, T., 1995. Forecasting international airline passenger traffic using neural networks. Logistics and Transportation 31 (3), 239-251.

Odom, M.D., Sharda, R., 1990. A neural network model for bankruptcy prediction. In: Proceedings of the IEEE International Joint Conference on Neural Networks. San Diego, CA, 2, pp. 163-168.

O'Leary DE. 1998. Using neural networks to predict corporate failure. *International Journal of Intelligent Systems in Accounting, Finance and Management 7:* 187-197.

Reed, R., 1993. Pruning algorithms - A survey. IEEE Transactions on Neural Networks, 4 (5), 740-747.

Refenes, A.N., 1993. Constructive learning and its application to currency exchange rate forecasting. In: Trippi, R.R., Turban, E. (Eds.), Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real-World Performance. Probus Publishing Company, Chicago.

Refenes, A.N., 1995. Neural Networks in the Capital Markets. John Wiley, Chichester.

Refenes, A.N., Zapranis, A., Francis, G., 1994. Stock performance modeling using neural networks: A comparative study with regression models. Neural Networks 7 (2), 375-388.

Roy, A., Kim, L.S., Mukhopadhyay, S., 1993. A polynomial time algorithm for the construction and training of a class of multilayer perceptrons. Neural Networks 6, 535-545.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by backpropagating errors. Nature 323 (6188), 533-536.

Rumelhart DE, McClelland JL. 1986 *Parallel Distributed Processing, Explorations in the Microstructure of Cognition, Volume 1: Foundations.* MIT Press: Cambridge, MA.

Salchenkerger, L.M., Cinar, E.M., Lash, N.A., 1992. Neural networks: A new tool for predicting thrift failures. Decision Science 23 (4), 899-916.

Schiffmann, W., Joost, M., Werner, R., 1993. Application of genetic algorithms to the construction of topologies for multilayer perceptron. In: Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms, pp. 675-682.

Sharda, R., Patil, R.B., 1992. Connectionist approach to time series prediction: An empirical test. Journal of Intelligent Manufacturing 3, 317-323.

Sietsma, J., Dow, R., 1988. Neural net pruning-Why and how? In: Proceedings of the IEEE International Conference on Neural Networks, 1, pp. 325-333.

Srinivasan, D., Liew, A.C., Chang, C.S., 1994. A neural network short-term load forecaster. Electric Power Systems Research 28, 227-234.

Tang, Z., Fishwick, PA., 1993. Feedforward neural nets as models for time series forecasting. ORSA Journal on Computing 5 (4), 374-385.

Theriou, N. G. and Tsirigotis, G. (2000): "The Construction of an Anticipatory Model for the Strategic Management Decision Making Process at the Firm Level", *International Journal of Computing Anticipatory Systems*, **9**: 127-142.

Trippi, R.R., Turban, E., 1993. Neural Networks in Finance and Investment: Using Artificial Intelligence to Improve Real-world Performance. Probus, Chicago.

Velido A, Lisboa PJG, Vanghan J. 1999. Neural networks in business: a survey of applications (1992-1998). *Expert Systems with Applications* 17: 51-70.

Vishwakarma, K.P., 1994. A neural network to predict multiple economic time series. In: Proceedings of the IEEE International Conference on Neural Networks, 6, pp. 3674-3679.

Waite T, Hardenbergh H. 1989. Neural nets. *Programmer's Journal 7:* No. 3, 10-22.

Wang, Z., Massimo, C.D., Tham, M.T., Morris, A.J., 1994. A procedure for determining the topology of multilayer feedforward neural networks. Neural Networks 7 (2), 291-300.

Wasserman PD. 1989. *Neural Computing: Theory and Practice.* Van Nostrand Reinhold: New York.

Weigend, A.S., Huberman, B.A., Rumelhart, D.E., 1992. Predicting sunspots and exchange rates with connectionist networks. In: Casdagli, M., Eubank, S. (Eds.), Nonlinear Modeling and Forecasting. Addison-Wesley, Redwood City, CA, pp. 395-432.

Weigend, A.S., Rumelhart, D.E., Huberman, B.A., 1991. Generalization by weight-elimination with application to forecasting. Advances in Neural Information Processing Systems 3, 875-882.

White, H., 1988. Economic prediction using neural networks: The case of IBM daily stock returns. In: Proceedings of the IEEE International Conference on Neural Networks, 2, pp. 451-458.

Wilson, R., Sharda, R., 1994. Bankruptcy prediction using neural networks. Decision Support Systems 11, 545-557.

Wong BK, Bodnovich TA, Selvi Y. 1997. Neural network applications in business. A review and analysis of the literature (1988-95). *Decision Support Systems* 19: 301-320.

Wong, B.K., Bodnovich, T.A., Selvi, Y., 1995. A bibliography of neural networks business application research: 1988-Sep-tember 1994. Expert Systems 12 (3), 253-262.

Wong, F.S., 1991. Time series forecasting using backpropagation neural networks. Neurocomputing 2, 147-159.

Wu, B., 1995. Model-free forecasting for nonlinear time series (with application to exchange rates). Computational Statistics and Data Analysis 19, 433-459.

Zhang, G., Patuwo, E. B., Hu, M. Y., (1997), "Wavelet neural networks for function learning", *IEEE Transactions on Signal Processing*, **43** (6): 1485-1497.

Zhang G, Patuwo E. B., Hu M. Y., (1998), "Forecasting with artificial neural networks: The state of the art", *International Journal of Forecasting* **14**: 35-62.