Automatic modelling of neural networks for time series prediction – in search of a uniform methodology across varying time frequencies

Nikolaos Kourentzes and Sven F. Crone

Lancaster University Management School, Department of Management Science
Bailrigg campus, Lancaster, LA1 4YX, United Kingdom

Abstract. In time series prediction, modelling neural networks poses multiple challenges in specifying suitable input vectors, network architectures, and training parameters depending on the underlying structure of the time series data. The data properties are often determined by the frequency in which the time series is measured, such as low frequency data of yearly, quarterly or monthly observations, or high frequency data of weekly, daily, hourly or even shorter time intervals. As different time frequencies require distinct modelling heuristics, employing neural networks to predict a set of time series of unknown domain, which may exhibit different characteristics and time frequencies, remains particularly challenging and limits the development of fully automated forecasting methodologies for neural networks. We propose a methodology that unifies proven statistical modelling approaches based upon filters and best practices from previous forecasting competitions into one framework, providing automatic forecasting without manual intervention by inferring all information from the data itself to model a diverse set of time series of varying time frequency, like the ESTSP'08 dataset.

1 Introduction

Artificial neural networks (NN) have found increasing consideration in forecasting research and practice, leading to successful applications in time series prediction and explanatory forecasting [1]. However, despite their theoretical capabilities of non-parametric, data driven approximation of any linear or nonlinear function directly from the dataset, NN have not been able to confirm their potential in forecasting competitions against established statistical methods, such as ARIMA or Exponential Smoothing [2]. As NN offer many degrees of freedom in the modelling process, from the selection of activation functions, adequate network topologies of input, hidden and output nodes, to learning algorithms and parameters and data pre-processing in interaction with the data, their valid and reliable use is often considered as much an art as science. Previous research indicates that the parsimonious identification of input variables to forecast an unknown data generating process without domain knowledge poses one of the key problems in model specification of NN [3, 4]. While literature provides guidance in selecting the number of hidden layers of a NN using wrapper approaches [5, 6], selecting the correct contemporaneous or lagged realisation of the dependent variable, and / or multiple explanatory variables, remains a challenge [7].

The issue of input variable and lag selection becomes particularly important, as the input vector needs to capture all characteristics of complex time series, including the components of deterministic or stochastic trends, cycles and seasonality, interacting in a linear or nonlinear model with pulses, level shifts, structural breaks and different distributions of noise. While some components may be addressed in a univariate model using only lagged realisations of the dependent variable, others may require the integration of explanatory dummy-variables with adequate time-delays. Although a number of methodologies have been developed to support the valid and reliable identification of the input vector for NNs, they do not perform well consistently [8], there have been no comparative evaluations between them [4] and consequently there is currently no consensus on what methodology should be applied under which circumstances and time series frequency. Furthermore, it is argued [9, 10] that these methodologies to specify the input vector do not apply to high frequency data of weekly or higher frequency, like those datasets provided for the 2008 ESTSP competition. In addition to identifying a methodology to specify the input vector for a given time series frequency, this raises a more substantial challenge associated with the variety of modelling methodologies: the challenge of developing a valid and reliable methodology for a set of time series of different frequency, which ultimately prohibits the generation of a fully automated NN forecasting system.

To address this challenge, this paper suggests a methodology founded on established best practices from previous time series forecasting competitions for NN and proven statistical methods. The resulting approach can be applied automatically, without the need of manual intervention from a human expert, producing forecasts for sets of time series of unknown domain and different frequencies. Finally, through the necessary research that led to the development of this modelling methodology, a set of problems associated with modelling NN on high frequency data were encountered and explored. These are discussed in contrast to the challenges of modelling on low frequency time series, revealing the increasing complexity of high frequency data and pointing to potential for future research. The paper is organized as follows. First, we briefly introduce NN in the context of time series forecasting. Methodologies for selecting the input vector and the number of hidden nodes are also discussed. Section 3 presents the experimental design and the results obtained. A discussion of the problem arising from the transition from low to high frequency time series is done in section 4. Finally, we provide conclusions and future work in section 5.

2 Methods

2.1 Forecasting with multilayer perceptrons

Forecasting with NNs provides many degrees of freedom in determining the model form and input variables to predict a dependent variable $\hat{y}$. Due to the large degrees of freedom in modelling NN for forecasting, we present a brief introduction to specifying feedforward NN for time series modelling; a general discussion is given in [11, 12]. Through specification of the input vector of n lagged realisations of only the dependent variable y a feedforward NN can be configured for time series forecasting as $\hat{y}_{t+1} = f(y_t, y_{t-1}, \ldots, y_{t-n+1})$, or by including i explanatory variables x_i of metric or nominal scale for causal forecasting, estimating a functional relationship of the form $\hat{y} = f(x_1, x_2, \ldots, x_z)$. By extending the model form through lagged realisations of the independent variables $x_{i,t-n}$ and dependent variable y_{t-n} more general dynamic regression and autoregressive (AR) transfer function models may be estimated. To extend the autoregressive model forms of feed-forward architectures to other stochastic processes, recurrent architectures incorporate moving average components (MA) of past model errors into the model, in analogy to the ARIMA-Methodology of Box and Jenkins [13]. Forecasting time series with NN is conventionally based on modelling a feed-forward topology in analogy to an non-linear autoregressive AR(*p*) model using a Multilayer Perceptron (MLP) [1, 14]. The architecture of a MLP of arbitrary topology is displayed in figure 1.

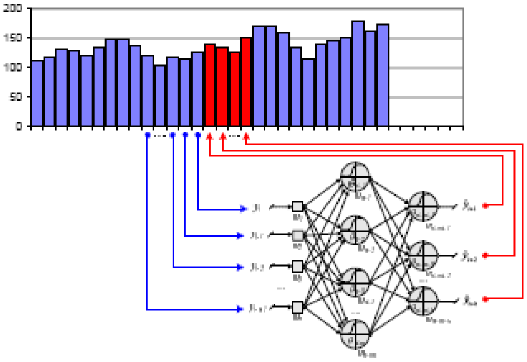

Fig. 1: Autoregressive MLP for time series forecasting

In time series prediction, at a point in time *t* a one-step ahead forecast $\hat{y}_{t+1}$ is computed using $p=n$ observations y_t, y_{t-1},…, y_{t-n+1} from *n* preceding points in time *t, t-1, t-2, …, t-n+1*, with *n* denoting the number of input units of the NN. Data is presented to the MLP as an overlapping set of input vectors formed as a sliding window over the time series observations. The task of the NN is to model the underlying generator of the data during training, so that a valid forecast is made when the trained NN is subsequently presented with a new input vector value [15]. The network paradigm of MLP offers extensive degrees of freedom in modelling for prediction tasks. Structuring the degrees of freedom, each expert must decide upon the selection and sampling of datasets, the degrees of data pre-processing, the static architectural properties, the signal processing within nodes and the learning algorithm in order to achieve the design goal, characterized through the objective function or error function. For a detailed discussion of these issues and the ability of NN to forecast univariate time series, the reader is referred to [1]. The specification of the input vector has been identified as being particularly crucial to achieving valid and reliable results followed by the specification of the number of hidden nodes [16, 17]. Both will be examined in the next section.

2.2 **Input variable selection for multilayer perceptrons**

The identification of relevant input variables and variable lags aims at capturing the relevant components of the data generating process in a parsimonious form. In time series modelling, it is closely related to identifying the underlying time series components of trend and seasonality and capturing their deterministic behaviour in lags of the dependent variable. A simple visual analysis of the time series components frequently fails to reveal the complex interactions of autoregressive and moving average components, multiple overlying and interacting seasonality of different cycle lengths and nonlinear patterns. Several methodologies have been suggested for input variables selection of the significant lags in forecasting, most originating from linear statistics and engineering. However, there exists no uniformly accepted approach to identify linear or nonlinear input variables [4]. After reviewing the alternative methodologies suggested in literature for specifying the input vector of MLPs, the most widespread approach was found to be a form of stepwise regression [18-20]. The approach employs a conventional stepwise regression to identify the significant lags of the dependent variable and uses them as inputs for the MLP, with straightforward extensions of this approach for multivariate modelling [19]. Conventionally, the parametric approach of linear stepwise regression assumes a stationary time series, which must not be satisfied for trended or seasonal time series patterns. However, no consensus exists on whether a time series with identified trend should be detrended, and whether a seasonal time series should be deseasonalised first to enhance the accuracy of NN predictions [3, 21, 22]. Alternatively seasonality or trend be incorporated in the NN structure using additional model terms and explanatory variables [23-25]. As removing trending and / or seasonality prior to identifying significant lags may impact on the structure of the identified input vector, we evaluate three candidates of stepwise regression using (a) the original time series, (b) the detrended time series, and (c) deseasonalised versions of it. The resulting input vectors were different in structure and length, and were used as competing candidates to specify the input vector for the original, undifferenced time series.

A problem largely neglected but directly related to identifying significant lags from the time series is setting a maximum number of lags into the past the input vector should be explored for significance. The common practice involves the use of an arbitrary heuristic, e.g. using lags up to three seasons and hence 36 lags, or through an iterative trial and error process during modelling similar to the ARIMA-methodology. While both approaches may be feasible for low frequency data, they fail for high frequency time series where the large sample size for each lag induces low significance bounds. As a result most lags in the past become statistically significant and should be included in the model, although a particular seasonality may be best captured by including only the relevant lag of the true season. As the significance of lags further in the past does not fade away as with low frequency data, all lags up to an arbitrary maximum would be included, creating very large input vectors. Despite its universal relevance for NN, Regression and ARIMA-modelling, this issue has not been explored in literature, with the exception of one paper noting the issue in the context of forecasting low frequency time series with MLPs [26]. As a solution to determine the maximum lag number that is required for high-frequency time series, we propose a method based on the Euclidean distance of a seasonal year-on-year-plot. Assuming no prior seasonal information, the time series of length n is split into n / s ‘seasons’ of different length, with $s = \{2, 3, 4, \dots , n / 2)$, and the Euclidean distance between all observations across seasonal sub- series is calculated. The seasonal length s^* that minimises the Euclidean distance indicates the minimum possible deviation (in squared error terms) of the seasons in a seasonal plot, thus providing an indication of seasonality and an upper limit of using 3**s* as a maximum lag length. The global minimum identifies the strongest single seasonality, while local minima found in this sense reflect the minimum distances as seasonality increases, indicating seasonality or multiples of seasonality. The identified seasonality then provides relevant modelling information of single or multiple seasonality to be incorporated into the input vector [25], using the Euclidean distance. (For example, assuming a daily time series, which exhibits both day of the week and day of the year seasonality, the Euclidean distance will reveal both seasonalities, with the weekly being 7 observations and the annual being 365 observations.)

Regarding the selection of the number of hidden layers, theory regarding universal approximation [5, 6] suggests that one hidden layer is sufficient to invoke the universal approximation properties of the NN. Therefore, the question on specifying the network topology may be simplified to specifying the number of hidden nodes to include in the single layer. For time series of varying frequency, prior research indicates that a different number of hidden nodes maybe required depending on the pre-specified input vector of different length [27]. To reflect this, we employ a wrapper with a constant grid size to select the correct number of hidden units, which reflects the most popular approach to specifying the NN architecture [4].

3 Experimental design

3.1 Exploratory data analysis and input vector specification

The ESTSP 08 competition provided three time series without any information on the domain of origin nor on the time series frequency, which are displayed in fig.2. As each time series may contain different characteristics, they are explored using the Seasonal Euclidian distance and the Augmented Dickey-Fuller (ADF) test for trend.

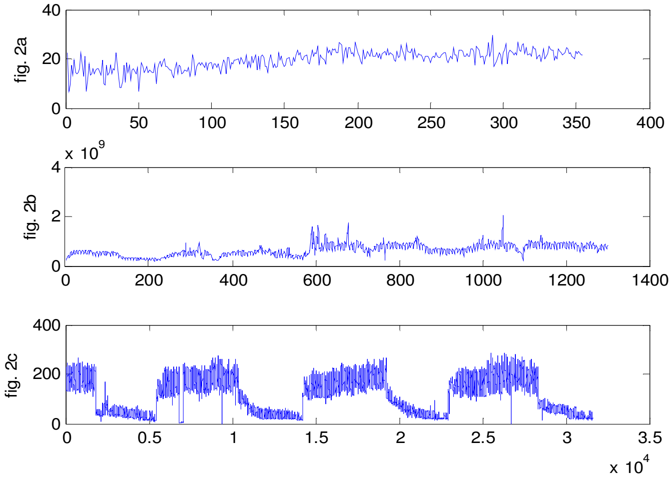

Fig. 2: The three ESTSP'08 time series

The first time series of the competition, plotted in fig. 2a, is comprised of 354 observations plus two explanatory time series to aid with the modelling of the time series. No domain knowledge on the time series is provided. The objective is to forecast the next 18 values. Applying the Euclidean distance approach a seasonality of 12-observations is identified; hence the time series is treated as monthly data containing 29.5 years of data. The abundance of data allows the use of three full seasons to identify the input vector. The ADF-test indicates the absence of trend, leaving only two options to model the input vector regarding pre-processing of the time series: both the original time series (1.a) and the time series after taking a 12th order difference to remove the seasonality (1.b) is used. Note that the seasonal differenced time series is used only for the stationary identification of the input vector - the NNs are modelled only on the original time series.

The second time series of the competition contains 1,300 observations without any explanatory time series. The objective is to forecast the next 100 values. Using the ADF-test an instationary time series with trend is identified. However, careful visual inspection of the time series and the seasonal series plot indicates a structural break in the form of a single level shift, visible in fig. 2b, rather than a continuous trend. As a consequence no 1st order differencing is required. Using the Euclidean distance approach seasonality of 7 and 365 observations are identified and we may infer that the time series contains daily observations. The significant lags are identified on the original time series (2.a), applying a 7th order differencing (2.b), a 365th order differencing (2.c) and both differences (2.d) in order to identify possible input vectors candidates.

The third time series of the competition, plotted in fig. 2c, contains 31,614 observations; the objective is to predict the 200 next values. The ADF-tests identifies no significant trend. Using the Euclidian distance approach we identify three potentially overlaying seasonalities of 24, 168 and 8,760 observations, indicating an hourly time series with hour of the day, day of the week, and day in the year seasonality. This provides several alternative input vectors by applying different levels of seasonal differencing, including the original series (3.a), the 24th (3.b), the 168th (3.c), and the 8,760th differenced series (3.d), plus four combinations or the differences. All identified candidate input vectors will be constructed and evaluated in a set of NN candidates, which are specified in the next section.

3.2 Artificial Neural Network models

We construct a set of conventional MLPs using a consistent methodology, where all modelling parameters are identical but the choice of the input vector and the number of hidden nodes, as specified above. In addition to the lagged inputs of the dependent variable and possible explanatory time series identified through the stepwise regression analysis, where applicable, a set of additional inputs was as a set of candidates for all time series. A single integer variable was used to code a deterministic seasonality, in contrast to conventional s-1 binary dummies that substantially increase the size of the input vector. This was done in order to capture additional aspects of the seasonality in addition to the AR(p) terms modelled through time lagged realisations, as suggested from previous studies [24]. Also, binary dummies were introduced to code level shifts for time series 2.

All MLPs apply a single output node with the identity activation function for a on-step-ahead prediction of t+1. Due to the possible interaction of the input vector size with the number of hidden nodes in a single hidden layer we evaluate different NN models for every input vector candidate using a stepwise grid-search with 2, 4, 6, 8, 10 and 12 hidden nodes to be considered for model selection. All hidden nodes for time series 1 and 2 apply a hyperbolic tangent as the activation function, while time series 3 uses the logistic activation function. This choice was made due to problems discovered during training of the 3rd time series, most probably due to the length of the time series resulting in a large number of training examples and the high degrees of freedom of the relevant neural network candidates. Each MLP is trained using simple back-propagation with momentum for 1,000 epochs or until an early stopping criterion is satisfied. For the early stopping criterion the mean squared error (MSE) is evaluated every epoch, and training is halted if no improvement was made for hundred epochs. The initial learning rate is set to η=0.5, applying a cooling factor $\Delta\eta$ to reduce the learning rate by 0.01 per epoch; the momentum term is kept constant at φ=0.4. All data is pre-processed using linear scaling into the interval of [-0.6, 0.6] and presented to the MLP using random sampling without replacement. Each MLP candidate is initialised 40 times with random starting weights in the interval of [-0.6, 0.6] in order to avoid local minima during the training and to provide an adequate error distribution using sufficient results.

3.3 Model selection

Given the large number of different alternative candidate models created, applying a different number of hidden nodes, input vectors and across the 40 initialisations used in training, model selection of the MLP candidate which promises the best out-of-sample performance on unseen data can be very challenging. The limited prior performance of NN, and, in particular, their low consistency and robustness of performance across homogeneous datasets in time series prediction [8] can in part be contributed to suboptimal model selection using a simple 1-fold cross validation. In contrast to selecting the best performing MLP candidate, we consider an ensemble of diverse candidates to generate average predictions. In addition to substantial evidence in classification that ensembles of simple methods perform well, this has long been confirmed for time series prediction, e.g. at the M competition, where a simple average of all competing methods performed better than each of the competing methods itself[2]. Based on this finding we rank all the MLP candidates for each time series, select the 10 best models and average their forecasts for each future horizon. These ensemble forecasts circumvent aspects of the challenges in model selection, however pose additional problems in evaluating different ensemble schemes. The ESTSP'08 competition assesses the accuracy of the models using a normalised mean squared error (NMSE) for each time series averaged over all three series. In order to align the performance metric for parameterisation and model selection with the final metric, a MSE proportional to the final metric was used during model development.

4 Experimental results

The composite ensemble forecasts for time series 1, 2 and 3 are given in fig. 3.

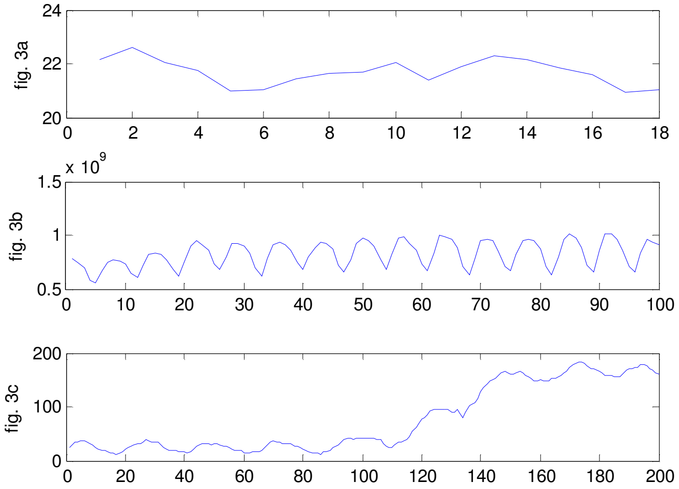

Fig. 3: Forecasts for the ESTSP'08 time series

Due to the large number of different MLP candidates evaluated and the space constraints it is infeasible to provide a comprehensive overview of the experimental results and architectures of the individual candidate models. Therefore we will restrict our discussion to some generalised findings: 80% of the top 10 candidate models (which were used to create the composite forecasts) uses an integer dummy variable to code each deterministic seasonality of different length, implying that this strategy aids the model to capture the complex overlying seasonal forms. For the candidate approaches for which the input vector was identified both on the original and the differenced time series, both model forms were always selected to be within the top 10 across all time series, implying that these approaches are complementary. An unexpected finding was that the univariate models for time series 1 outperformed all multivariate models using the two additional explanatory time series. This reduced the complexity of creating the final forecasts, as no predictions for the explanatory time series were required and no accumulation of the errors due to inaccurate forecasts of the explanatory variables could be introduced into the final forecasts of time series 1.

5 Challenges in modelling high frequency data

One elementary characteristic of high frequency time series data is the increase in length of the time series given a constant time interval, and the resulting increase in training vectors. For the frequencies employed in the ESTSP'08 competition the hourly time series would be 24 and 720 times longer than the daily and the monthly time series, had an identical time period been used. This difference in the sample size creates several challenges in the three frequency domains even though an identical same modelling procedure is followed. The most important implications for this set of experiment, handling the degrees of freedom, input vector length in model identification and computational time, are outlined in Table 1 and discussed below.

Time Series	Frequency	Average no. of inputs	% Difference in inputs	Maximum time lag	% Difference in lag
1	Monthly	7	-	t-36	-
2	Daily	30	328.57%	t-392	988.89%
3	Hourly	354	4957.14%	t-9072	25100.00%

Table 1: Average number of inputs and maximum time lag per time series.

5.1 Degrees of freedom

This analysis employed several ways to identify the input vector for each time series. These derived from different options on performing seasonal differencing in the presence of a single or multiple seasonalities, or not. Across all number of input vectors determined for each candidate we compute the average number of inputs for each time series. The findings listed in table 1 exemplify the magnitude of the increase in both the size of the number of inputs used for time series of increasing frequency, and of the resulting increase of the degrees of freedom purely from the number of input nodes. This illustrates the increased complexity of training a MLP as the data frequency increases. Taking into consideration the number of hidden nodes, a candidate model developed for the hourly time series would use 2,478 parameters on average, in comparison to only 49 for the monthly time series. The implications this has for the training are apparent, as well as the difficulty of solving such a complex optimisation problem. Further interactions seem to exist also with alternative modelling choices: for time series 3 the architectures using a hyperbolic tangent activation function in the hidden layer could not be trained using backpropagation, as the optimiser could not cope with the degrees of freedom. This suggests the need for future research regarding NN topology, not only with regard to predictive accuracy but also with regard to robustness and consistency of the architecture.

5.2 Model identification

In addition to the increase in input vector size, our experiments identified a positive correlation between the frequency of the time series and the size of the search space required to find suitable input lags. This is again illustrated in table 1, where the maximum lag that was evaluated for each time series is provided. Not only does the input vector for time series of higher frequency increase in size, the maximum time lag to be considered also moves further into the past. Most methodologies to identify the input vector based upon wrappers, grid search, exhaustive random search, genetic algorithms and other meta-heuristics based on computational force are bound to encounter constraints in providing results in a reasonable time frame. In contrast, the filter approach based upon an iterative stepwise regression equally requires long computation times to identify the appropriate lags to use, proportional to the increase in the search space. On the other hand, filter approaches utilising the autocorrelation and the partial autocorrelation information of the time series are limited in their accuracy to provide useful information for model identification due to the increased number of significant lags resulting from a growing sample size and tight confidence intervals. Consequently, modelling time series of higher frequencies requires the careful consideration of the trade-of between compute power, filter and wrapper based approaches.

5.3 Computational time

The regression approach employed here appears to be adequate for the time series frequencies in question, providing solid identification of the relevant time lags for forecasting in an acceptable time. However, computational time varied substantially, ranging from virtually instantaneous for the time series 1 and 2 to several hours for time series 3. Experiments for the first two time series were computed on a 2.2 GHz INTEL dual core processor with 3 GB of RAM, running 2-3 hours. For the third time series initial computations identified resource problems. As a consequence the experiments were computed on a high performance cluster with two dual core processors at 2.4 GHz with 10GB of RAM dedicated for this task, which required several days. It appears that experiments on high frequency data require additional computational power beyond the scope of normal personal computers, in particular for multiple architectures and model ensembles. Alternatively, these may provide the requirements for developing alternative training methods to perform well for large datasets under the current computational resources constraints.

6 Conclusions

This paper proposes an initial methodology for automatic modelling of time series with arbitrary time frequencies, seasonalities and trends, using the true ex ante predictions of the ESTSP'08 competition. The principle of the model is to compute competing candidate models of MLPs with different input vectors utilising varying temporal information on trends, stochastic and deterministic seasonality through autoregressive (AR) and / or integer dummy variables respectively. In order to omit the need of manual intervention we employ a composite ensemble forecast from the 10 best models on the in sample performance of each time series. Ways to avoid arbitrary modelling decisions are described, concerning the selection of the input vector, number of hidden layers and the hidden nodes. The proposed methodology, which is based on established tools and methods, manages to surpass the problems that trouble most neural network methodologies in literature when facing sets of time series of varying time granularity and frequency.

The analysis finishes with identifying some of the main problems encountered in the extension of the methodology towards high frequency data. Given the computational resources, high frequency data remain to be extremely demanding and limit the amount of ad-hoc experimentation. Unique problems arise that are beyond the scope of this paper, requiring further research. There is an apparent need to explore the possibility of training the MLPs in a way that the sheer amount of data will not require unreasonably long time and can cope with the increased degrees of freedom of the neural network models.

References

1. G. Zhang, B.E. Patuwo, and M.Y. Hu, “Forecasting with artificial neural networks: The state of the art,” *International Journal of Forecasting*, vol. 14, no. 1, 1998, pp. 35-62.
2. S. Makridakis, and M. Hibon, “The M3-Competition: results, conclusions and implications,” *International Journal of Forecasting*, vol. 16, no. 4, 2000, pp. 451-476.
3. T. Hill, M. O'Connor, and W. Remus, “Neural network models for time series forecasts,” *Management Science*, vol. 42, no. 7, 1996, pp. 1082-1092.
4. G.Q. Zhang, B.E. Patuwo, and M.Y. Hu, “Forecasting with artificial neural networks: The state of the art,” *International Journal of Forecasting*, vol. 14, no. 1, 1998, pp. 35-62.
5. K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators ” *Neural Networks*, vol. 2, no. 5, 1989, pp. 359-366.
6. K. Hornik, “Approximation capabilities of multilayer feedforward networks,” *Neural Networks*, vol. 4, no. 2, 1991, pp. 251-257.
7. B. Curry, and P.H. Morgan, “Model selection in neural networks: Some difficulties,” *European Journal of Operational Research*, vol. 170, no. 2, 2006, pp. 567-577.
8. J.S. Armstrong, “Findings from evidence-based forecasting: Methods for reducing forecast error,” *International Journal of Forecasting*, vol. 22, no. 3, 2006, pp. 583-598.
9. C.W.J. Granger, “Extracting information from mega-panels and high-frequency data,” *Statistica Neerlandica*, vol. 52, no. 3, 1998, pp. 258-272.
10. J.W. Taylor, L.M. de Menezes, and P.E. McSharry, “A comparison of univariate methods for forecasting electricity demand up to a day ahead,” *International Journal of Forecasting*, vol. 22, no. 1, 2006, pp. 1-16.
11. C.M. Bishop, *Neural networks for pattern recognition*, Oxford University Press, 1995.
12. S.S. Haykin, *Neural networks : a comprehensive foundation*, Prentice Hall, 1999.
13. G.E.P. Box, and G.M. Jenkins, *Time series analysis: forecasting and control*, Holden-Day, 1970.
14. A. Lapedes, and R. Farber, “How neural nets work,” *Neural Information Processing Systems*, D.Z. Anderson ed., American Institute of Physics, 1988, pp. 442-456.
15. G. Lachtermacher, and J.D. Fuller, “Backpropagation in Time-Series Forecasting,” *Journal of Forecasting*, vol. 14, no. 4, 1995, pp. 381-393.
16. G.P. Zhang, B.E. Patuwo, and M.Y. Hu, “A simulation study of artificial neural networks for nonlinear time-series forecasting,” *Computers & Operations Research*, vol. 28, no. 4, 2001, pp. 381-396.
17. G.P. Zhang, “An investigation of neural networks for linear time-series forecasting,” *Computers & Operations Research*, vol. 28, no. 12, 2001, pp. 1183-1202.
18. N.R. Swanson, and H. White, “Forecasting economic time series using flexible versus fixed specification and linear versus nonlinear econometric models,” *International Journal of Forecasting*, vol. 13, no. 4, 1997, pp. 439-461.
19. M. Qi, and G.S. Maddala, “Economic factors and the stock market: A new perspective,” *Journal of Forecasting*, vol. 18, no. 3, 1999, pp. 151-166.
20. C.M. Dahl, and S. Hylleberg, “Flexible regression models and relative forecast performance,” *International Journal of Forecasting*, vol. 20, no. 2, 2004, pp. 201-217.
21. M. Nelson, T. Hill, W. Remus, and M. O'Connor, “Time series forecasting using neural networks: Should the data be deseasonalized first?,” *Journal of Forecasting*, vol. 18, no. 5, 1999, pp. 359-367.
22. G.P. Zhang, and M. Qi, “Neural network forecasting for seasonal and trend time series,” *European Journal of Operational Research*, vol. 160, no. 2, 2005, pp. 501-514.
23. S.F. Crone, “A new perspective on forecasting seasonal time series with artificial neural networks,” *25th International Symposium on Forecasting*.
24. S.F. Crone, and N. Kourentzes, “Input variable selection for time series prediction with neural networks-an evaluation of visual, autocorrelation and spectral analysis for varying seasonality,” *European Symposium on Time Series Prediction*, pp. 195-205.
25. B. Curry, “Neural networks and seasonality: Some technical considerations,” *European Journal of Operational Research*, vol. 179, no. 1, 2007, pp. 267-274.
26. S.D. Balkin, and J.K. Ord, “Automatic neural network modeling for univariate time series,” *International Journal of Forecasting*, vol. 16, no. 4, 2000, pp. 509-515.
27. K.P. Liao, and R. Fildes, “The accuracy of a procedural approach to specifying feedforward neural networks for forecasting,” *Computers & Operations Research*, vol. 32, no. 8, 2005, pp. 2151-2169.