Lancaster University Management School
Working Paper
2003/098

The Accuracy of a Procedural Approach to Specifying Feedforward Neural Networks for Forecasting

Robert Fildes and Kua-Ping Liao

The Department of Management Science
Lancaster University Management School
Lancaster LA1 4YX
UK

The Accuracy of a Procedural Approach to Specifying Feedforward Neural Networks for Forecasting

Kua-ping Liao* and Robert Fildes**

*Department of Industrial Management, Kun Shan University of Technology,
Tainan County, Taiwan, R.O.C.
Email: kpliao@yahoo.co.uk

**Corresponding author:
Department of Management Science, The University of Lancaster,
Lancaster LA1 4YX, U.K.
Email: R.Fildes@Lancaster.ac.uk

Revised December 2003

Scope and Purpose

The forecasting capability of neural networks (NN), when applied to economic time series, remains moot with recent tests of comparative performance disappointing and dependent on the particular specification selected. Earlier research has established that the performance of a NN model depends quite critically on the process by which it is constructed. However, most previous evaluations have not developed systematic procedures to choose the specification. In order to evaluate the forecasting effectiveness of NN, it is important to ensure they are adequately and objectively built. This paper has two purposes: to propose an effective and computationally viable approach to objectively specifying the structure of NN and, secondly, to evaluate its success by a thorough examination of its forecasting performance when compared to various alternative statistical forecasting methods.

Abstract

The comparative accuracy of feedforward neural networks (NN) when applied to time series forecasting problems remains uncertain. This is because most studies suffer from either of two defects – they choose the NN from a wide range of alternatives in order to present the forecast accuracy results in the best light, or they do not compare the results with suitable benchmarks. In order to overcome both these objections this paper proposes an objective procedure for specifying a feedforward neural network models and evaluates its effectiveness by examining its forecasting performance compared with established benchmarks. After the selection of input nodes based on cross-validation, a 3-Stage procedure is proposed here which consists of sequentially selecting first the learning rate followed by the number of nodes and the initial weights. This paper shows that neural networks only perform robustly if they are built by considering these three factors jointly. In an empirical demonstration of the strength of the approach, those neural network models, built by considering all three factors, performed better than other competitive statistical methods when evaluated rigorously on a standard test data set.

Keywords: backpropagation; neural networks; comparative forecasting accuracy; model specification

1. Introduction

Neural network approaches to time series forecasting are becoming more accepted as part of the armory of forecasting techniques that are used in many application areas. The computer science community has been an effective publicist of these new methods, despite the limited evidence of their effectiveness compared to established methods (see e.g. Chatfield, [1]). However, more recently, there has been increasing evidence that the accuracy of neural network methods is comparable to existing methods [2] although in standard time series comparisons their performance still remains disappointing [3]. This paper further contributes to this growing literature by examining the sensitivity of forecast accuracy to network specification. In so doing, it aims to overcome one key weakness in much of the earlier work, the lack of an objective approach to model construction [4].

The first stage in defining a procedural approach to specifying and estimating a feedforward neural network is to select a limited number of input nodes from the set of inputs. A number of techniques have been proposed including structural stabilization, regularization and Bayesian approaches [5]. Here we will use a selection procedure based on cross-validation. Once the input variables are specified the following important choices must be made[1]: the number of hidden nodes, the learning rates, and the initial values assigned to the weights and biases of the neural network. A 3-Stage

[1] Of course, other choices must also be made including for example, the number of hidden layers. In empirical forecasting applications a single hidden layer has always provided sufficient flexibility.

Training approach is proposed here to resolve these choices and train a neural network automatically, ensuring an objective and replicable approach to model selection.

In order to evaluate the value of this 3-Stage approach rigorously two issues need to be faced. First the 3-stage approach must be shown to lead to an adequate and computationally practical NN compared with alternative specifications and the procedure underlying its construction justified, and second, the forecasting performance of the resulting neural network models must be compared with other established forecasting approaches. This has been achieved by using a standard data set, the telecommunications data set [6] and the benchmarks adopted for making the comparisons with neural nets were those developed in Fildes et al. [7]. The organization of the paper is as follows. The next section discusses the process of building a backpropagation neural net, the choice of input variables and the proposed 3-Stage procedural approach to its specification and training. Section 3 discusses the evaluation of neural net forecasting methods. Section 4 gives a brief summary of the data used and the benchmark forecasting methods while the results are presented in sections 5 and 6. Looking ahead to the conclusions we find that when the neural net is built using appropriate and objective methods it can outperform the best benchmark alternative forecasting methods.

2. Building a Neural Network: The 3-Stage Training Approach

Backpropogation neural network methods applied to time series forecasting has now an established pedigree (see Zhang et al. [2] for a summary). Before estimating a backpropagation neural network forecasting model (and after defining the set of possible input variables), a number of decisions must be made to fully specify the proposed network. Neuneier and Zimmerman [8], for example, give a full discussion of the various elements. These include deciding which particular inputs to include, the choice of the number of hidden nodes, the learning rates, and an initializing procedure for the weights and biases of the neural network etc. Some earlier researchers have shown the effects on forecasting accuracy of some of the choices researchers must make. For example, Farraway and Chatfield [9] examined the effects of the choice of input nodes and the number of hidden neurones, concluding with the comment 'great care is needed to choose an appropriate set of input variables, an appropriate architecture …and an appropriate numerical procedure for fitting an NN model'. Some of these choices are more critical than others, although there is no clear consensus, so for example, Zhang et al. [10] examined the number of input nodes and the number of hidden nodes (as well as sample size) in a simulation experiment, concluding that the number of input nodes was much more important. However, in their experiments, so long as the inputs were sufficient to characterize the non-linear equations being simulated, there was nothing to be gained from including additional inputs and sometimes there was a (minor) loss. Thus, the choice of the input set needs to be sufficiently broad to characterize the data series and the domain of application but not too large (relative to sample size) to undermine pruning/ model simplification algorithms. In addition the approach used to limit the input nodes should not be overly conservative. Here, a 3-Stage training procedure is proposed to specify a NN structure in order to build and estimate a forecasting model automatically, once the set of possible inputs is selected. The approach ensures the forecasting model building process is almost entirely data-driven but, because of its sequential nature, it is also conservative in its use of computer resources. Simultaneous estimation of the parameters demands much more computer time (see appendix 1). The sequential 3-stage procedure also attempts to enhance the robustness of the chosen neural network model and to convert the complexity and subjectivity inherent in neural network model building into a flexible and more objective methodology. Figure 1 presents an overview of the process of specifying a neural net suitable for forecasting when the 3-Stage Training approach is adopted.

Figure 1 here. The process of specifying a neural network forecasting model when using 3-Stage Training

Stage 1: Pilot Training

The estimation (or 'learning') of a specific neural network relies on the non-linear estimation of the parameters and the optimizing search routine depends on a pre-selected learning rate. The main goal of this first stage is to set up the learning rate of the neural network.

The first stage requires the input set to be specified, based on the modeler's knowledge of the problem area and the known time series characteristics, bearing in mind data limitations. Once the potential input variables have been selected, the data set available for model building and estimation is then split into a 'training' data set and a 'validation' set (according to the usual practice when building a neural network) in order to overcome the overfitting problem [11]. In order to derive a reasonable learning rate efficiently, simple network structures that contain a small number of hidden nodes and a single hidden layer are trained to predict the outputs of the validation data set by minimizing a pre-selected error measure, usually MSE. The choice of learning rate etc, then can be designed as a small-scale competition (1 or 3 hidden nodes were used here in the empirical comparison to reflect the limited data availability). The optimal learning rate is obtained when a network structure can use that learning rate to predict best the outputs of the validation data.

The effect of using only simple structures in choosing the learning rate effectively advantages them compared to more complex alternatives. This has been used as an alternative to a penalty approach, which penalizes more complex structures quite heavily. The approach proposed here only gives the simpler structures the advantage of choosing the learning rate. Once the learning rate is chosen, it is used in training all the models at the later stages. The use of an adaptive learning rate with pre-specified parameters such as those given by Smith [11] is an equally well-specified approach (and in the empirical comparison was shown to have similar performance characteristics).

<u>Stage 2: Competition Training</u>

At this stage, neural networks with all the structures included from the set under consideration are trained with the learning rate determined as at the first stage. The network structure that best predicts the outputs of the validation data set is recorded and will be used at the third stage.

Stage 3: Reward Training

The recorded optimal structure is retrained several times. Each time, in initialization a different set of random numbers are given to the weights and the biases of the neural network. That set of random numbers, which best predicts the outputs of the validation data set, is selected. The neural network model is now specified and can be used to forecast ahead.

A full description of these three stages which lead to an objectively specified structure and estimation procedure is given in appendix 1 together with the parameters chosen including data transformations.

3. The Evaluation of Neural Network Forecasting Models

The forecasting evaluations of neural networks in the literature have taken one of two approaches: either a limited set of model structures (with a fixed learning rate etc) have been considered and the best chosen using a pre-selected fit criterion, or alternatively, a broader set was considered and one complete specification from among the many is lauded for its strong performance. While the former places artificial constraints on the potential of NNs, because of the limited structures considered, the latter flatters their achievements because no objective selection method is specified, i.e. the model structure is chosen retrospectively to produce the best performance. In order to investigate whether neural network forecasting models are effective, the out-of-sample forecasting performance of objectively and adequately built neural networks should be evaluated. To carry out this task fairly, there are various issues which must be considered. In a study by Adya and Collopy [4], guidelines first described in Collopy et al. [12] were used to evaluate the effectiveness of the validation component. These required comparisons with well-accepted models, and the use of ex ante validation data based on a reasonable sample of forecasts. Expanding on these guidelines, seven issues will be addressed here.

(1) To evaluate the forecasting performance of neural network models, benchmarks methods should be used to demonstrate how performance of the network model differs from relevant competitors [1]. By using competitive models as benchmarks, we can evaluate the performance of neural networks rigorously. Benchmarks should be chosen based on past research in a similar area or the results of forecasting competitions. In the present study, the benchmarks used are those methods which have been evaluated in the study by Fildes et al. [7] as they apply directly to the data under analysis. In that study, the Robust Trend method, designed specifically for the telecoms data [6], was shown to perform convincingly better than all the other methods used. The inclusion of methods such as Holt's has the added advantage that they have been widely tested and shown to perform well across a wide variety of series [13].

(2) Since relative forecasting performance can be influenced by the evaluation criteria adopted [6], the criteria should be carefully chosen before model estimation. This can prevent biasing the evaluation by using some specific evaluation criteria, chosen ex post. Performance evaluation criteria should be able to provide a summary of the forecast error distribution and/or to indicate the actual costs incurred by forecasting errors. The evaluation criteria used in the present study are those criteria which were adopted in the study by Fildes et al. [7] and evaluated in Fildes [14]. These include MAPE, GMRAE and MdRAE and are defined in appendix 2.

(3) The 'test' or out-of-sample data set should only be used to measure forecasting performance, after model specification and estimation is complete. It should not be used in the model estimation or model selection process. This ensures that real 'forecasting' performance is evaluated.

(4) If the aim is to establish generalisable conclusions about the performance of neural network methods, the number of series selected from the target population should be adequate for the problem under consideration. Using a small number of series increases the risks that a particular specification that 'fits' does not represent the typical data generating processes of the target population of series. Moreover, by using more series, we can test neural networks' capabilities of adapting themselves to model different data generating processes within a particular class. Here 261 monthly non-seasonal series are used.

(5) Forecasting performance should be evaluated over different forecasting horizons. The M-competition [15] showed that "the performance of various methods depends upon the length of the forecasting horizon." [7]. In applications it is important to establish whether neural networks can forecast well for short and/or long forecasting horizons. Because of their non-linearity it might be expected that performance over the longer horizon would be relatively stronger. In the present study, the forecast horizon is 18 months.

(6) For time series forecasting applications, different forecasting time origins should be used. This is because, for each series or data set, the forecasting performance of a model can be influenced by the sampling variability of time origins and the effects of the estimation/ test samples sizes [7]. In the present study, 5 time origins are used. This permits the examination of the reliability of the different methods over time as well as sample size effects.

(7) A clear, objective and adequate procedure for building neural network forecasting models should be adopted so that the models built can be replicated by others. While most of the studies surveyed by Adya and Collopy [4] fail this test, Balkin and Ord's [3] evaluation on the M3 data is fully objective although their process is more limited than that employed here – they only specify a criterion for selecting the number of hidden nodes, rather than the 3-Stage training approach adopted here.

4. The Data, Alternative Forecasting Methods and Error Measures

In choosing the test data on which to evaluate the proposed procedure a number of issues need to be borne in mind. The relatively poor performance of a forecasting method on heterogeneous data such as that in the M3-Competition where NN performed disappointingly does not imply equally poor performance on a more homogeneous data set [14,7]. The data used in this study are 261 non-seasonal time series taken from a telecommunications data set. Thus, our choice of a homogeneous data set removes one source of experimental variability.

There are 71 data points in each time series. Some have argued that NN require a large sample size to perform effectively but there is no consensus on this issue [4]. While multicollinearity in the parameter estimates is inevitable with small samples, the forecasting accuracy of the resulting model is not necessarily damaged. If sample size proves to be important we would expect to see NN performing better when a larger training set is used. This is evaluated in the experimental comparisons we describe by comparing performance over forecast origins. Strong performance when the 'training' data set is short would give support to those who argue that NN is a robust forecasting method in most circumstances.

Each time series represents the monthly data of the number of special circuits in service and is non-seasonal. For a more detailed description of the data set, its statistical features and an exploratory data analysis of its features, reference can be made to [6, 7, 14]. The key feature of the data are its non-stationarity with a dominant negative trend, limited autoregressive information and a considerable number of outliers. Thus, Grambsch and Stahel [6] had proposed a robust technique which modelled the differenced data and included the median of the first differences.

The core Neural Network training method used in this study is the backpropagation neural network (see for example, Rumelhart et al. [16]), built by using the 3-Stage training approach. The set of possible inputs consisted of three autoregressive lags of the first differenced data , given the limited data. In addition, based on the earlier research, the median (of the first differenced data), was included as this feature has been shown by Grambsch and Stahel [6] and Fildes [14] to have potential predictive power. The forecasting performances of the feedforward neural network models was then compared with a variety of statistical forecasting methods that had earlier been used by Fildes et al. [7], ensuring a fair and thorough evaluation. The methods used were:

i. Holt's exponential smoothing with linear trend model

ii. Damped Smoothing [17]

iii. ARIMA

iv. ARARMA [18]

v. Robust Trend model [5]

Researchers have concluded no single forecast error measure contains all the interesting information relevant to evaluating forecasting accuracy and that relative performance between methods depends on the error criteria used (as discussed in (2) above). Two basic measures were used here: the absolute percentage error (APE) and the Relative Absolute Error. The following were used to satisfy the various criteria of robustness, reliability and scale independence promulgated by Armstrong and Collopy [19] and Fildes [14]. Definitions and formulae are given in appendix 2.

(i) mean absolute percentage error (MAPE)

– this suffers unduly from the effects of outliers (e.g. when an observed value is close to zero leading to very large APE)

(ii) median absolute percentage error (MdAPE)

- this has greater applicability and robustness than MAPE but discards much of the information in the error distribution. This is because a target value of zero in the test period can make the calculation of the percentage error unmanageable.

(iii) Geometric mean and median relative absolute error

- Both measures have proved robust to outliers with a more normal distribution than other error measures while the former contains information on the relative magnitude of all the errors.

Following the recommendation of Adya and Collopy [4] the models were evaluated using 5 forecast time origins, periods 23, 31, 38, 45 and 53. The NNs (and the benchmark statistical methods) were therefore specified and trained on the data interval [1,23] etc and the subsequent 18 points (for forecasting lead times, 1-18) used to evaluate each method. Thus, for any given forecasting origin and forecast horizon there were 261 errors to be summarised. These error measures were then summarised across the 5 forecast time origins, using means, medians, geometric means (appropriate for ratios and percentages) and ranks. Hu et al [19] describe this out-of-sample cross-validation in greater detail.

5. The Effects of Choosing the Learning Rate, the Number of Hidden Nodes, and the Initial Weights

As figure 1 points out, various decisions must be made for any NN to be operational. In order to be confident that the results we report here are competitive with a number of alternative, pre-structured NN specifications we have carried out a full experimental design examining the three factors of learning rate, number of hidden nodes and initial weights. We have also evaluated the effects of using the full-set of input nodes compared to the cross-validation approach we have used to simplify the model structure (an issue we explore more fully in section 5.1). An additional element to complete the specification included the choice of transfer function. The logistic function was used with the inputs scaled in the interval (.35,.65). Full details of the procedure are given in appendix 1. By demonstrating the 3-stage approach performed well we can be better assured that the range of settings we considered for these factors is appropriate to deliver effective forecasting accuracy when we go on to compare this automatic NN specification with conventional forecasting techniques. This ensured the NN results are not handicapped in the comparisons we make with the statistical methods in section 6.

A number of different variants of the 3-Stage Training Approach have been considered and these are shown in Table 1. ‘NN’ represents the Neural Network structure when the full 3-stage Training Approach is adopted which specifies the learning rate, number of hidden nodes and the initial weights through the validation data set. To give a second example, ‘NN3’ is where the number of hidden nodes is pre-determined as 3.

Table 1. The Neural Network Models under Consideration

(Cross-validation is used to determine the number of lags to be used as inputs to each of the models. The number considered ranges from 1 to 3. All the models except NN_AR(1-3) include the median as one of the inputs.)

NN	Stage 1:The set of the numbers of hidden nodes tried is {1,3}. The set of the learning rates tried is {0.01,0.1,1}. Stage 2: The set of the numbers of hidden nodes tried is {1,3,5}. Stage 3: The set of the initial ranges of weights between the input layer and the hidden layer is {[-0.001,0.001],[-0.01,0.01],[-0.1,0.1]}.
NN1	Only stage 1and stage 3 are used. The number of hidden nodes is 1. Stage 1:The set of the learning rates tried is {0.01,0.1,1}. Stage 3: The set of the initial ranges of weights between the input layer and the hidden layer is {[-0.001,0.001],[-0.01,0.01],[-0.1,0.1]}.
NN3	Only stage 1and stage 3 are used. The number of hidden nodes is 3. Stage 1:The set of the learning rates tried is {0.01,0.1,1}. Stage 3: The set of the initial ranges of weights between the input layer and the hidden layer is {[-0.001,0.001],[-0.01,0.01],[-0.1,0.1]}.
NN5	Only stage 1and stage 3 are used. The number of hidden nodes is 5. Stage 1:The set of the learning rates tried is {0.01,0.1,1}. Stage 3: The set of the initial ranges of weights between the input layer and the hidden layer is {[-0.001,0.001],[-0.01,0.01],[-0.1,0.1]}.
NN3_0.01	Only Stage 3 is used. The number of hidden nodes is 3. The learning rate is 0.01. Stage 3: The set of the initial ranges of weights between the input layer and the hidden layer is {[-0.001,0.001],[-0.01,0.01],[-0.1,0.1]}.
NN3_0.1	Only Stage 3 is used. The number of hidden nodes is 3. The learning rate is 0.1. Stage 3: The set of the initial ranges of weights between the input layer and the hidden layer is {[-0.001,0.001],[-0.01,0.01],[-0.1,0.1]}.
NN3_1	Only Stage 3 is used. The number of hidden nodes is 3. The learning rate is 1. Stage 3: The set of the initial ranges of weights between the input layer and the hidden layer is {[-0.001,0.001],[-0.01,0.01],[-0.1,0.1]}.
NN3_0.1_0.001	The number of hidden nodes is 3, the learning rate is 0.1, and the initial range of the weights between the input layer and the hidden layer is [-0.001, 0.001].
NN3_0.1_0.01	The number of hidden nodes is 3, the learning rate is 0.1, and the initial range of the weights between the input layer and the hidden layer is [-0.01, 0.01].
NN3_0.1_0.1	The number of hidden nodes is 3, the learning rate is 0.1, and the initial range of the weights between the input layer and the hidden layer is [-0.1, 0.1].
NN_AR(1-3)	The 3-Stage training used in NN is also used in this model, however the median is not included as an input to the neural network.

We first examine the effects on forecasting performance of the choice of the number of hidden nodes, the learning rate and the initial weight range using a 3-factor factorial design. For each data series to be forecast, the accuracy using the test out-of-sample data set and measured by MAPE (and GRMSE), is potentially affected by the above 3 factors.

The comparisons have all been carried out using the three greater forecast origins, 38, 45, and 53 to minimize possible sample size effects. Because the sample size available in the empirical problem we considered was relatively small, neural network specifications that lead to large numbers of parameters to estimate could not be sensibly considered. As a consequence only relatively simple structures have been included as part of the procedure. With more data both the input set and the number of hidden nodes could be enlarged though the principle of parsimony and the results of the various forecasting competitions [13,15] suggest that too complex a structure will lead to noise being interpreted as signal with correspondingly poor out-of-sample forecasting performance. The set of the numbers of hidden nodes considered was therefore limited to {1,3,5}. The set of learning rates used was {0.01,0.1,1}. The set of the initial ranges of the weights between the input layer and the hidden layer was [-0.001,0.001],[-0.01,0.01],[-0.1,0.1]}. These small positive and negative weights were used to avoid extreme outputs of the hidden nodes and thus avoid slow learning caused by small error derivatives [11]. Overall, the parameters that have been used to specify the 3-Stage procedure have been pre-selected because of the known characteristics and size of the data set. Table 2 summarises the results of the ANOVA carried out when the error measure used is MAPE and the forecasting horizon is 1. The learning rate is found to be significant. This holds also when GRMSE is used as the error measure . More generally, no matter whether MAPE or GMRSE is used as the error measure and no matter whether the forecasting horizon is 1, 6, 12 or 18, the learning rate is significant. (The time origin block effect is also significant, unsurprisingly.)

Table 2 ANOVA Results for the MAPE Data When the Forecasting Horizon is 1

(N: the number of hidden nodes; LR: the learning rate; W: the initial ranges of the weights between the input layer and the hidden layer; 3 time origins (38,45,53) are treated as different blocks)

Source of Variation	Degrees of Freedom	Mean Square	F
Blocks	2	0.087134	14.6822**
N	2	0.000174	0.02930
L	2	0.059665	10.05362**
W	2	0.000006	0.0011
Interaction NL	4	0.000080	0.0135
Interaction NW	4	0.000021	0.0036
Interaction LW	4	0.000003	0.0004
Interaction NLW	8	0.000020	0.0034
Error	52	0.005935	
Total	80		

**significant at the 1% significance level

Fisher's least significant difference (LSD) procedure was used to investigate further where the differences occur. The data set used were based on the three longer forecast origins to limit any sample size effects. The three treatments considered below were selected

Table 3 The treatments of the three factors investigated in the factorial experiment

(N: the number of hidden nodes; LR: the learning rate; W: the initial ranges of the weights between the input layer and the hidden layer)

Factor	Treatment 1	Treatment 2	Treatment 3
N	1	3	5
LR	0.01	0.1	1
W	[-0.001,0.001]	[-0.01,0.01]	[-0.1,0.1]

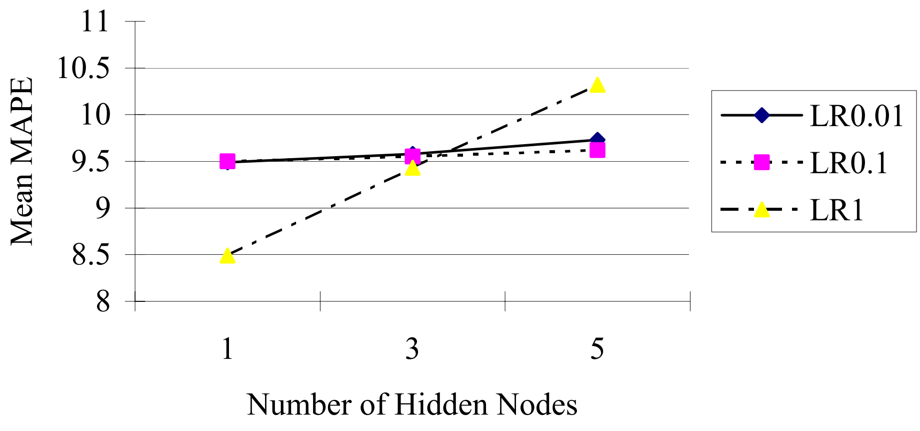
Table 4 Results of LSD procedure at the 1% significance level

Evaluation Criterion	Horizon	Factor	Mean for Treatment 1 (LR=0.01)	Mean for Treatment 2 (LR=0.1)	Mean for Treatment 3 (LR=1)	LSD	Performance Order*
MAPE	1	LR	1.25	1.24	1.16	0.06	Tr.3,(Tr.2,Tr.1)
	6	LR	5.45	5.44	4.37	0.14	Tr.3,(Tr.2,Tr.1)
	12	LR	10.46	10.44	7.95	0.22	Tr.3,(Tr.2,Tr.1)
	18	LR	15.83	15.77	12.32	0.53	Tr.3,(Tr.2,Tr.1)
GMRAE	1	LR	1.51	1.51	1.39	0.03	Tr.3,(Tr.1,Tr.2)
	6	LR	0.51	0.51	0.44	0.03	Tr.3,(Tr.1,Tr.2)
	12	LR	0.45	0.45	0.36	0.02	Tr.3,(Tr.2,Tr.1)
	18	LR	0.46	0.46	0.38	0.03	Tr.3,(Tr.2,Tr.1)

*For performance order, treatments in parenthesis are not different at the 1% significance level.

Table 4 shows that a learning rate of 1 performs better than the neural networks with a learning rate of 0.01 or 0.1. However, establishing the effects of the number of hidden nodes (N) and the choice of weights are more complicated. When cross-validation is used to select the suitable input nodes, the effect is uncertain. When the time origin is 53, the full input set is invariably used and, with the forecasting horizons treated as different blocks, the ANOVA results show that the number of hidden nodes, the learning rate and their interaction are significant no matter whether MAPE or GMRSE is used as the error measure. Figure 2 shows the interaction between the number of hidden nodes and the learning rate based on MAPE.

Figure 2 The interaction between the number of hidden nodes
and the learning rate based on the MAPE data



The conclusion we draw from this experimental analysis is the need to ensure the proposed NN algorithm leads to a careful choice of number of hidden nodes and the learning rate. No a priori specification is likely to be dominant. As a consequence the proposed 3-stage approach should have the facility to evaluate and choose between a wide range of alternative specifications.

One further comparison has been carried out: a test of whether the cross-validation pruning of the input nodes has any significant impact on accuracy. The results (omitted for reasons of space) for both error measures showed, when aggregated over lead times and forecast horizons, that there was no significant difference between keeping the inputs fixed (at three) or allowing the selection algorithm to choose the most appropriate inputs. Only in the case of longer horizons and MAPE as the error measure did the selection method of input variables affect performance (so long as the initial set is broad enough as in Zhang et al [10]).

Whilst the ANOVA demonstrates the significance of the effective choice of the two factors, it does not show whether the effects are important (as measured by the various accuracy statistics). Tables 5 through 7 summarise the results for a range of NN specifications including the 3-Stage Approach, for the various lead times and for the various error measures. The best and second best performers are shaded. The corresponding results for the Geometric Mean calculations are omitted as they are very similar to those shown. Results for the individual forecast origins support the conclusions and are available from the authors.

Table 5: MAPE summarised across 261 series for all neural network methods,

summarised across time origins by Mean and over forecast horizon, by Overall Sum of Ranks.

		Forecast Horizon			Overall Sum
Method	**1**	**6**	**12**	**18**	**of Ranks**
NN	1.171	4.056	7.551	11.743	11
NN1	1.173	4.113	7.621	12.033	24
NN3	1.170	4.090	7.620	11.974	17
NN5	1.168	4.050	7.547	11.757	8
NN3_0.01	1.253	4.889	9.313	14.399	38
NN3_0.1	1.251	4.880	9.285	14.358	31
NN3_1	1.171	4.097	7.516	11.826	15
NN3_0.1_0.001	1.254	4.858	9.374	14.382	39
NN3_0.1_0.01	1.254	4.857	9.373	14.381	35
NN3_0.1_0.1	1.254	4.857	9.369	14.373	33
NN_AR(1-3)	1.167	4.058	7.511	11.767	8

Key

▓	Best performing method
░	Second best method

Table 6 MAPE summarised across 261 series, for all neural network methods,

summarised across time origins by Median and over forecast horizon, by Overall Sum of Ranks.

		Forecast Horizon			Overall Sum
Method	**1**	**6**	**12**	**18**	**of Ranks**
NN	1.142	3.830	7.351	11.585	11
NN1	1.146	3.844	7.502	12.213	24
NN3	1.139	3.808	7.380	11.756	13
NN5	1.140	3.835	7.325	11.596	11
NN3_0.01	1.235	4.875	9.382	13.840	34
NN3_0.1	1.226	4.876	9.378	13.827	29
NN3_1	1.140	3.834	7.339	11.708	13
NN3_0.1_0.001	1.230	4.886	9.480	14.110	39
NN3_0.1_0.01	1.230	4.886	9.478	14.108	36
NN3_0.1_0.1	1.231	4.883	9.479	14.112	40
NN_AR(1-3)	1.138	3.831	7.359	11.657	11

N.B. To calculate the overall sum of ranks, each method is ranked for each horizon (by MAPE) and the results summed to give an overall picture of performance.

Table 7: MAPE summarised across 261 series, for all neural network methods,

summarised across forecast origins by sum of ranks and over forecast horizon, by Overall Sum of Ranks

	Forecast Horizon				Overall Sum
Method	1	6	12	18	of Ranks
NN	27	16	18	12	73
NN1	21	30	26	29	106
NN3	16	16	19	19	70
NN5	17	15	18	15	65
NN3_0.01	46	48	46	51	191
NN3_0.1	35	45	41	45	166
NN3_1	22	19	13	18	72
NN3_0.1_0.001	43	44	47	43	177
NN3_0.1_0.01	45	43	45	41	174
NN3_0.1_0.1	44	42	44	42	172
NN_AR(1-3)	10	12	13	15	50

Almost all the summary tables (as well the robust geometric mean results) point to the success of the 3-Stage approach. However it is clear that the use of a fixed number of hidden nodes (NN1, NN3 and NN5) is less important than the arbitrarily fixing of the other two parameters with the weights the least important (as measured by the range of overall ranks). The various error measures all show that neural network model performs badly when the smallest learning rate of 0.01 used. The differences are also substantial (shown in Tables 5 & 6) so these results support earlier research that the choice of the learning rate has an important effect on the forecasting performance of a feedforward neural network. The results from the three error measures are highly correlated, showing that NN performs well in both the magnitude it beats its competitors and the frequency.

5.1. Sensitivity to Choice of Input Nodes

A potentially critical issue in using NN as an extrapolative forecasting method is the choice of input variables. In the experiments described above we have examined two alternative procedures applied to an input set that includes the last three lags and the median of the differenced data (i) where cross-validation is used to select a subset of the inputs appropriate for each series individually, (ii) where all the inputs are used. In addition we considered the case where the median is excluded despite it being seen as necessary based on the preliminary analysis of Grambsch and Stahel [5]. Table 8 summarises the performance for three later forecast origins. The Wilcoxon signed rank test was used to compare the forecasting performance of NN (which includes the median) and the NN-AR(1_3) (without the median). Differences of zero were ignored. Where some differences were equal, they were assigned average ranks. The test was based on the 4 lead time and 5 forecast time origin comparisons. The results of testing whether the median of the MAPEs of the NN_AR(1_3) model is greater than that of the NN model (the H_1 hypothesis holds), versus H_0 (these two medians are equal) are shown in Table 8 for the MAPE and the GMRAE. At 1% significance level we conclude that the forecasting performance of the NN model is equal to that of the NN-AR(1_3) model and we conclude that the choice of input variables is not critical to performance in this particular case. Table 8 also shows the significance of the choice of learning rate. However, as noted in Appendix 1, the validation testing was based on MAPE (while the training used a MSE criterion). When MSE was used for both validation and training, the NN_AR(1_3)_MSE model, performance was substantially worse (as shown in Table 7). This is suggestive that forecasting performance is improved by using MAPE for validation when the forecast criteria using the test out-of-sample data are based on relative error.

Table 8 Wilcoxon signed rank tests comparing different NN specifications

(at the 1% significance level: null hypothesis of equal medians vs first named method less accurate; R^- less than the critical value leads to rejecting the null)

Error Measure	Competing Models	Pairs of observations with different values	Critical value for one-sided test	R^-
MAPE	NN_AR(1-3) versus NN	15	19	64.5
	NN3_0.01 versus NN3_1	19	37	0
	NN3_0.1 versus NN3_1	19	37	0
	NN_AR(1-3)_MSEv versus NN_AR(1-3)	20	43	1
GMRAE	NN_AR(1-3) versus NN	8	1	18.5
	NN3_0.01 versus NN3_1	19	37	0
	NN3_0.1 versus NN3_1	20	43	1.5
	NN_AR(1-3)_MSEv versus NN_AR(1-3)	19	37	19

5.2. Experimental Summary

To summarise the results so far, the 3-Stage Training Procedure has been shown to be at least the equal of the alternative pre-specified networks (although there may of course be feedforward networks which, when trained more intensively, e.g. in parallel rather than sequentially perform better). The key question we now address is whether this simple procedure for structuring and estimating a NN shows adequate performance characteristics when compared to standard forecasting benchmarks.

6. Comparative Forecasting Performance

A number of comparisons have been carried out, in particular the 3-stage NN has been compared with the methods already evaluated in Fildes et al. [7]. The results are shown in Tables 9-12, where the forecasting performances of the neural network model (NN from the evaluation in the previous section) is comparable with the best performing Robust Trend Method. Both models outperform all the other models consistently. From Table 10 and Table 11, we can see that the neural network model performs best based on the evaluation criterion GMRAE or MdRAE.

For the relative error measures, the performance of Robust Trend is very similar to that of Holt's model. Both these two models perform clearly worse than the neural network model. Using MAPE, for a short forecasting horizon of between 1 and 12 months, Robust Trend performs better than the neural network model (Table 9). But the difference in forecasting performance is negligible when the forecasting horizon is 12 months with the neural network model better for the longer horizon. This suggests some slight non-linearity in the trend that the linear method, robust trend, cannot pick up. The two more robust measures, MdAPE shown in Table 10, and GMRAE in Table 12 show NN as uniformly more accurate which suggests the MAPE results of Table 6 arise from a few outlying values.

Table 9 The Means of the MAPEs of six different forecasting methods

(summarised across a sample of 5 forecasting time origins)

	Forecast Horizon							Sum of Ranks
Method	1	6	12	18	1-6	1-12	1-18	
Holt	1.36	5.28	9.82	15.05	3.37	5.66	8.05	28
Damped	1.35	5.77	12.29	19	3.58	6.41	9.72	33
Robust-T	1.11	3.95	7.54	11.8	2.54	4.28	6.19	8
ARIMA	1.51	6.51	14.41	22.92	3.98	7.29	11.38	42
ARARMA	1.37	4.48	9.79	14.82	2.94	5.03	7.69	23
NN	1.17	4.06	7.55	11.74	2.64	4.37	6.24	13

NB: NN denotes the neural network model specified through the 3-stage approach.

Table 10 The Medians of the MAPEs of 6 forecasting methods

(summarised across a sample of 5 forecasting time origins)

	Forecast Horizon							Sum of Ranks
Method	1	6	12	18	1-6	1-12	1-18	
Holt	1.25	5.3	10.77	16.2	3.54	6.09	8.8	28
Damped	1.25	5.74	11.84	18.83	3.45	6.31	9.51	32
Robust-T	1.02	4.18	7.71	12.11	2.46	4.27	6.25	12
ARIMA	1.53	6.76	13.94	22.27	4.17	7.16	11.28	42
ARARMA	1.28	4.38	8.9	14.07	2.97	4.94	7.26	23
NN	1.14	3.83	7.35	11.58	2.47	4.2	6.08	9

Overall, NN performs at least as well as Robust Trend whether measured through the MAPE or the MdAPE. Using the latter measure, the neural network model performs better than the Robust Trend model except when the forecasting horizon is 1 month.

Table 11 The Ranks of the MAPEs of 6 forecasting methods

(summarised across a sample of 5 forecasting time origins)

	Forecast Horizon							Sum of Ranks
Method	1	6	12	18	1-6	1-12	1-18	
Holt	19	20	17	16	20	19	17	128
Damped	22	23	23	25	23	23	24	163
Robust-T	8	8	9	9	8	9	9	60
ARIMA	27	29	30	30	29	30	30	205
ARARMA	17	16	18	17	16	17	18	119
NN	12	9	8	8	9	7	7	60

Table 12 The means of the GMRAEs of 6 forecasting methods

(summarized across a sample of 5 forecasting time origins)

Numbers in parentheses indicate the rankings. *These values are affected by the Winsorizing Process.

Method	Forecast Horizon			
	1*	6	12	18
Holt	(3) 1.35	(3) 0.72	(2) 0.60	(3) 0.54
Damped	1.39	0.81	0.78	0.77
Robust-T	(3) 1.35	(2) 0.71	(3) 0.61	(2) 0.52
ARIMA	(2) 1.33	0.81	0.82	0.84
ARARMA	1.38	0.75	0.67	0.6
NN	(1) 1.13	(1) 0.48	(1) 0.4	(1) 0.39

7. Conclusions

A 3-Stage sequential training procedure has been used to build a feedforward neural network model objectively. Cross-validation was used intensively both to speed up the training process (as the processing times included in Appendix 1 show) and to help build a neural network with better generalisation capability to overcome the bias/ variance trade-off. This is important when building a model based on a general specification using a more intricate training approach than the 3-Stage approach adopted here. From the results of this study, we can see that the neural networks built using the 3-Stage training procedure is at least the equal of neural networks with a pre-specified structure. In fact, performance turns out to be critically affected by the decision on the learning rate and to a lesser extent the number of hidden nodes. The choice of input variables (within a plausible initial set) was found not to be important.

Neural network methods have not clearly demonstrated their ability to match the performance of standard statistical forecasting methods on economic and business time series as the survey by Zhang et al. [2] shows, in particular previous studies have failed to meet the criteria laid down by Adya and Collopy [4]. This research has added to the evidence, avoiding the pitfalls that earlier studies have fallen into. A variety of error measures have been considered, rigorous benchmarks employed and an objective and replicable approach to neural network models adopted. Neural network methods were shown to perform as well as the best performing benchmark on the test data but this required the use of an objective specification procedure.

Like all empirical studies, the evaluation of the 3-Stage Procedure has its limitations in that it has been applied to one specific homogeneous data set and the sample sizes used in training were small. Other more complex architectures could have offered even better performance. The fact that the performance of NNs under these conditions was strong is significant, especially when compared to methods such as exponential smoothing which has often proved its effectiveness. It demonstrates, perhaps for the first time convincingly, that NN can be a cost-effective winner when compared to rigorous benchmark methods. Further work is needed on a data set such as M3 (where NN performed poorly) and on simulated data, where the 3-Stage procedure can be compared to other methods of specifying the structure of the Neural Net. In addition, the 3-Stage procedure could itself be generalised to include those factors that we pre-specified such as the number of hidden layers, with cross- validation used to ensure parsimony and good out-of-sample performance characteristics.

8. Appendix 1: Neural Network Model Building

The complete process of building a neural network model for forecasting the telecommunications data will be described below.

1. Defining the Training and Validation Data Set

Suppose the time origin of forecasting is T. Each original time series y(t), t=1, 2, 3, ..., T, will be first differenced to get a series d(t), t=2, 3, 4, ..., T. The series d(t) is then be scaled to a series s(t), t=2, 3, 4, ...T, the values of which are in a range between 0.35 and 0.65 in order to avoid slow learning at the extremes where the error derivatives are very small. This also ensures the output is within the target range.

The most recent 3 lagged values of the series s(t) and the median (m(t)) of all the past lagged values of the series s(t) are used as the inputs to the neural network. That is to say m(t) is the median of s(2),s(3), ..., s(t).

2. Splitting the Data Set into a Validation Data Set and a Training Set

One third of the data will be used as the validation data and two thirds of the data will be used as the training data.

3. Building a Neural Network by Using 3-Stage Training

Throughout the logistic transfer function is used.

3.1. Pilot Training

The number of hidden nodes used is $\{1,3\}$. The set of learning rates used is $\{0.01,0.1,\}$.

3.2. Competition Training

The number of hidden nodes considered is $\{1,3,5\}$. Since the subset $\{1,3\}$ were used in pilot training, on 5 hidden nodes are further considered.

3.3. Reward Training

Two more sets of initial weights and biases are considered.

4. Forecasting Ahead on Test Data

The same notation that was used to describe the design of the whole data set will be used here. At forecast origin T, the three most recent values of the scaled first differenced time series are s(T-2), s(T-1), s(T). m(T) is the median of s(2), s(3), ..., s(T). To forecast one period ahead, the inputs to the neural network built are s(T-1), s(T), s(T-2) and m(T). Suppose that the output from the neural network is $\hat{s}(T+1)$. To forecast another period ahead, the inputs to be used are s(T-1), s(T), $\hat{s}(T+1)$ and $\hat{m}(T+1)$. $\hat{m}(T+1)$ is the median of s(2), s(3), ..., s(T) and $\hat{s}(T+1)$. This process of forecasting ahead will continue until all the values to be forecasted are produced. Suppose the forecasting horizon is h. $\hat{s}(T+1)$, $\hat{s}(T+2)$, $\hat{s}(T+3)$, ..., $\hat{s}(T+h-1)$ and $\hat{s}(T+h)$ are the forecasted values of the scaled first differenced series s(t). These values will be scaled back to get $\hat{d}(T+1)$, $\hat{d}(T+2)$, $\hat{d}(T+3)$, ..., $\hat{d}(T+h-1)$ and $\hat{d}(T+h)$, which are the forecasted values of the first differenced time series d(t). Suppose $\hat{y}(T+p)$ is the forecasted value of the original time series p period ahead. The forecasted values of the original time series y(t) can be obtained by using the following formulae:

$\hat{y}(T+1) =$ y(T) + $\hat{d}(T+1)$

$\hat{y}(T+p) = \hat{y}(T+p-1) + \hat{d}(T+p)$, where p =2, 3, 4, ……, h.

For an Intel Pentium III 733 MHz CPU, the average time needed for building a neural network model (NN model in this study) for forecasting a time series increases linearly from 2.51 second per series for 23 data points rising to 6.96 for 53 points. It should be noted that the NN model was programmed in C and the measured time includes all the data processing time. In this study, the method of cross-validation is used and network training is stopped if after 200 iterations the validation error can not be reduced. The maximum number of iterations allowed is 1000. The neural network selected is the one that produces the smallest validation error.

Different combinations of the learning rate, the number of hidden nodes and the initial ranges of the weights and biases were also tried to find the optimal one. NNpo was built by using this kind of

Parallel Optimization method. The set of the numbers of hidden nodes used was {1,3,5}. The set of learning rates used was {0.01,0.1,1}. The set of the initial ranges of the weights (and biases) between the input layer and the hidden layer was [-0.001,0.001],[-0.01,0.01],[-0.1,0.1]}. The time needed for building a parallel model was about 3 times as long as the time needed for building an NN model confirming Hill et al's observation [21] about the time consuming nature of a full search. (At the 5% significance level we accept that NNpo performs better than NN based on the error measure of GMRAE, however the difference between their performances was very small.) No matter whether MAPE or GMRAE was used as the error measure, the difference between their performances proved negligible.

9. Appendix 2: Error Measures

Let $Y_{j,t}$ represent the tth observation on the jth series. Let $e_{i,j,t}(l)$ be the error made in forecasting *l* periods for the jth series from forecast origin t using method *i*.

***MAPE**$_{i,k}$**(l)**:* =

$$\frac{1}{n}\sum_{j=1}^{n}\frac{\left|e_{i,j,t_k}(l)\right|}{Y_{j,t_k+l}}$$

(Mean Absolute Percentage Error)

where summation is over the j=1to 261 series for each time origin, k=1 to 5, for each method, *i* and for lead *l*.

These are further summarized across the time origins using the following error measures:
(a) Mean over k of $MAPE_{i,k}(l)$ for each forecast horizon *l* and each method *i*
(b) Median over k of $MAPE_{i,k}(l)$ for each forecast horizon *l* and each method *i*
(c) Rank over *i* for each forecast horizon

The relative performance is also ranked with the results for each horizon summarised by the sum of the ranks.

The second method used is the geometric mean of the relative absolute errors relative to a base line naïve method.

***GMRAE**$_{i,k}$**(l)**=*

$$\exp(\frac{1}{n_i}\sum_{j=1}^{n}\ln\left(\frac{\left|e_{i,j,t_k}(l)\right|}{\left|e_{0,j,t_k}(l)\right|}\right)$$

(Geometric Mean of Relative Absolute Error)

As above these are summarised across the time origins k using
(a) the Geometric Mean of ***GMRAE**$_{i,k}$**(l)***
(b) the Median.

The relative performance is also ranked with the results for each horizon summarised by the sum of the ranks.

10. References:

[1] Chatfield, C. Neural networks: Forecasting breakthrough or passing fad. International Journal of Forecasting 1993;9:1-3.

[2] Zhang G, Patuwo BE, Hu MY. Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting 1998;14:35-62.

[3] Balkin SD, Ord JK. Automatic neural network modeling for univariate time series. International Journal of Forecasting 2000;16:509-515.

[4] Adya M, Collopy F. How effective are neural networks at forecasting and prediction? A review and evaluation. Journal of Forecasting 1998;17:481-495.

[5] Bishop CM. Neural Networks for Pattern Recognition. Oxford University Press, 1995.

[6] Grambsch P, Stahel WA. Forecasting demand for special telephone services. International Journal of Forecasting 1990;6:53-64.

[7] Fildes R, Hibon M, Makridakis S, Meade N. Generalising about univariate forecasting methods: Further empirical evidence. International Journal of Forecasting 1998;14:339-358.

[8] Neuneier R, Zimmermann HG. How to train neural networks. In: Orr GB, Muller KR (Eds.). Neural networks: Tricks of the trade. Springer Verlag, 1998. p.373-423.

[9] Faraway J, Chatfield C. Time series forecasting with neural networks. Applied Statistics 1998;47:231-250.

[10] Zhang G., Patuwo BE, Hu MY. A simulation study of artificial neural networks for nonlinear time-series forecasting. Computers and Operations Research 2001;28:381-396.

[11] Smith M. Neural networks for statistical modeling. Van Nostrand Reinhold, 1993.

[12] Collopy F, Adya M, Armstrong JS. 1994. Principles for examining predictive validity: The case of information systems spending forecasts. Information Systems Research 1994;5:2:170-179.

[13] Makridakis S, Hibon M. The M3 competition. International Journal of Forecasting

2000;16:451-476.

[14] Fildes R. The evaluation of extrapolative forecasting methods. International Journal of Forecasting 1992;8:81-98.

[15] Makridakis S, Anderson A, Carbone R, Fildes R, Hibon M, Lewandowski R., Newton J, Parzen E, Winkler R. The accuracy of extrapolation (time series) methods: Results of a forecasting competition. Journal of Forecasting 1982;1:111-153.

[16] Rumelhart DE, Hinton G.E, Williams RJ. Learning internal representations by error propagation. In Rumelhart DE, McClelland JL, and the PDP Research Group (Eds.). Parallel distributed processing Vol.1. Cambridge , MA : MIT Press, 1986. p.318-362.

[17] Gardner ESJr, McKenzie ED. Forecasting trends in time series. Management Science 1985;31:1237-1246.

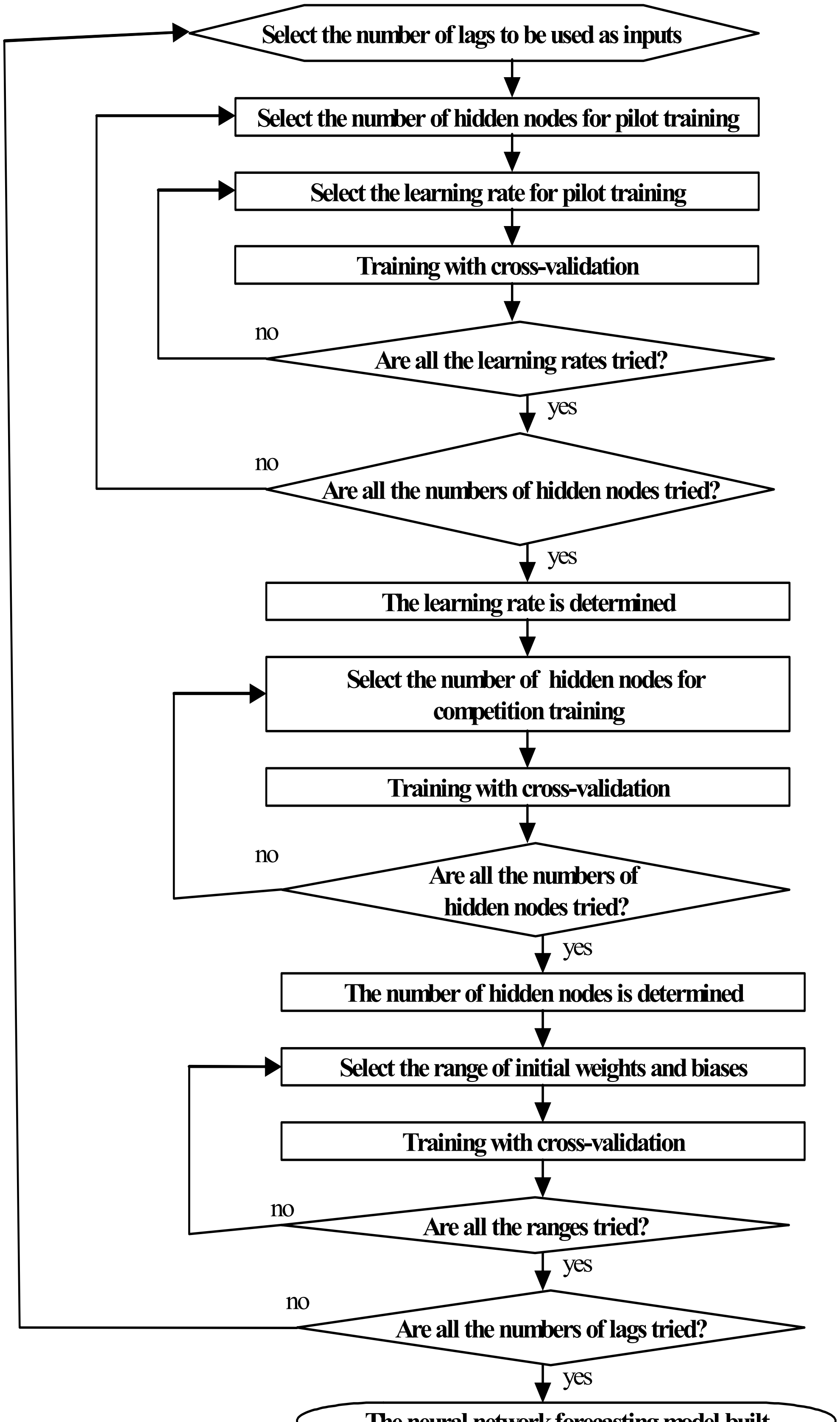
[18] Parzen E. ARARMA models for time series analysis and forecasting. Journal of Forecasting 1982;1:67-82.

[19] Armstrong JS, Collopy F. Error measures for generalising about forecasting methods: Empirical comparisons. International Journal of Forecasting 1992;8:69-80.

[20] Hu MY, Zhang G., Jiang CX. and Patuwo BY. A cross-validation analysis of neural network out-of-sample performance of exchange rate forecasting. Decision Sciences 1999;1 Winter:197-216.

[21] Hill T., O'Connor M. and Remus W. Neural network models for time series forecasts. Management Science 1996:42, 1082-1092.

Figure 1 The process of specifying a neural network forecasting model by using 3-Stage training


Select the number of lags to be used as inputs
Select the number of hidden nodes for pilot training
Select the learning rate for pilot training
Training with cross-validation
no
Are all the learning rates tried?
yes
no
Are all the numbers of hidden nodes tried?
yes
The learning rate is determined
Select the number of hidden nodes for competition training
Training with cross-validation
no
Are all the numbers of hidden nodes tried?
yes
The number of hidden nodes is determined
Select the range of initial weights and biases
Training with cross-validation
no
Are all the ranges tried?
yes
no
Are all the numbers of lags tried?
yes
The neural network forecasting model built