Multi-step Learning Rule for Recurrent Neural Models: An Application to Time Series Forecasting

INÉS M. GALVÁN AND PEDRO ISASI
Dep. of Computer Science, Universidad Carlos III, Av. de la Universidad 30, 28911 Leganés, Madrid, Spain

Abstract. Multi step prediction is a difficult task that has attracted increasing interest in recent years. It tries to achieve predictions several steps ahead into the future starting from current information. The interest in this work is the development of nonlinear neural models for the purpose of building multi step time series prediction schemes. In that context, the most popular neural models are based on the traditional feedforward neural networks. However, this kind of model may present some disadvantages when a long term prediction problem is formulated because they are trained to predict only the next sampling time. In this paper, a neural model based on a partially recurrent neural network is proposed as a better alternative. For the recurrent model, a learning phase with the purpose of long term prediction is imposed, which allows to obtain better predictions of time series in the future. In order to validate the performance of the recurrent neural model to predict the dynamic behaviour of the series in the future, three different data time series have been used as study cases. An artificial data time series, the logistic map, and two real time series, sunspots and laser data. Models based on feedforward neural networks have also been used and compared against the proposed model. The results suggest than the recurrent model can help in improving the prediction accuracy.

Key words: multi step prediction, neural networks, time series, time series modelling

1. Introduction

Time series prediction is a major goal in many areas of research, e.g. biology, physics, business and engineering. The ability to forecast the behaviour of a system hinges, generally, on the knowledge of the laws underlying a given phenomenon. When this knowledge is expressed as a solvable equation, one can predict the behaviour along the future once the initial condition is given. However phenomenological models are often unknown or extremely time consuming. Nevertheless, it is also possible to predict the dynamic behaviour of the system along the future by extracting knowledge from the past. We are interested in time series processes, which can be viewed as generalized nonlinear autoregressive models, also named NAR models. In this case, the time series behaviour can be captured by expressing the value $x(k+1)$ as a function of the d previous values of the time series, $x(k), \ldots, x(k-d)$, that is:

$$x(k+1) = F(x(k), \ldots, x(k-d)), \tag{1}$$

where k is the time variable and F is some function defining a very large and general class of time series.

In many time series applications, one-step prediction schemes are used to predict the next sample of data, $x(k+1)$, based on previous samples. However, one-step prediction may not provide enough information, especially in situations where a broader knowledge of the time series behaviour can be very useful or in situations where it is desirable to anticipate the behaviour of the time series process.

The present study deals with long-term or multi-step prediction, i.e. to obtain predictions several steps ahead into the future $x(k+1), x(k+2), x(k+3), \ldots$ starting from information at instant k. Hence, the goal is to approximate the function F such that the model given by Equation (1) can be used as a multi-step prediction scheme.

Many different methods have been developed to deal with nonlinear time series prediction. Among them neural networks occupy an important place being able to adequately model the nonlinearity and nonstationarity while being simple to train and to implement. Since the initial works, neural networks have been proved to be a powerful method in accuracy for time prediction, exceeding conventional methods by orders of magnitude [1]. The power of neural networks in time series prediction is based in some special features:

- Neural networks make no assumptions about the nature of the distribution of the data and are not therefore, biased in their analysis. Neural networks develop an internal representation of the relationship between the variables [2].
- Neural networks are the best method at discovering non-linear relationships [3, 4].
- Neural networks perform well with missing or incomplete data [5].
- The forecasting period is shorter than traditional models [5].

The neural models most widely used in time series applications are built up using the standard multilayer feedforward neural networks [6 8]. Models based on feedforward neural networks, also called in this work classical neural models, can be used for the purpose of multi-step prediction. They consist of approximating the function F by a multilayer feedforward network. When the training is finished, the output of the network is fed back into the input and the model is used to predict the behaviour of the time series along the interval $[k+1, k+h+1]$, where h is a natural number named prediction horizon. However, the static mapping realised by this kind of networks may provide poor predictions of the time series along the future because they are trained for the purpose of one-step prediction.

It has been shown that the modelling capacity of feedforward neural networks can be improved if the iteration of the network is incorporated into the learning process [9, 10]. It seems to be rather natural to use recurrent neural networks because they have, in addition to the static mapping between input and output, also the internal states of the network. These states work as a short-term memory and they are able to represent information about the preceding inputs [11]. That information takes an important place when the goal is long-term prediction.

In this Letter a Multi-Step Recurrent Neural model (MSRN) is used for modelling the behaviour of the time series, which is based on a partially recurrent neural network. The use of this network is motivated by two factors: first, the goal dealt with in this work, multi-step prediction; and, second, to solve the troubles found in feedforward neural models in the context of multi-step prediction.

The partially recurrent neural network used to built up the MSRN model consists of adding feedback connections from the output neurone to the input layer which allow the neural network to memorise previous prediction values. In this case, the parameters of the MSRN model are determined to minimise the error along interval $[k+1, k+h+1]$. Thus, the model is trained for the purpose of long-term prediction and better predictions than classical feedforward neural models may be expected.

The Letter is organised as follows. In Section 2, the classical neural models are reviewed analysing their disadvantages when they are used for multi-step prediction purpose. In Section 3, the MSRN model proposed in work is presented. The architecture of the partially recurrent neural network is briefly described and the learning procedure is explained. The experimental results are shown in Section 4. Both classical and MSRN models are applied to three different time series. An artificial series, the logistic map, and two real data time series, sunspots data and data measured in a physics laboratory representing some behaviour of a laser. Finally, in Section 5 the conclusions drawn from this work are analysed.

2. Feedforward Neural Model for Multi-step Prediction

As has previously been mentioned, the neural models most widely used in time series applications are based in feedforward neural networks with backpropagation learning algorithm. These models consist of approximating the function F appearing in Equation (1) by a multilayer feedforward neural network. Introducing the vector $(x(k), \ldots, x(k-d))$ as the kth network input pattern, the one-step predicted value by the neural model can be written as follows:

$$\tilde{x}(K+1) \quad \tilde{F}(x(k), \ldots, x(k-d), W_1), \tag{2}$$

where W_1, is the parameter set of the neural model, which is obtained using the backpropagation algorithm [12]. The update of the parameter set is based on the local error between the measured and predicted values, i.e.:

$$e(k+1) \quad \tfrac{1}{2}(x(k+1) - \tilde{x}(k+1))^2. \tag{3}$$

When the model given by Equation (2) has to forecast the values of time series at instants $k+1, k+2, \ldots, k+h+1$, this is along the interval $[k+1, k+h+1]$, its structure has to be modified because the sequence of measured values are not available. Therefore, the predicted network output at instant $k+1$ must be fed back as an input for the next step prediction and all the remaining input neurone values are shifted back one unit. This strategy must be repeated in the next prediction steps until the instant $k+h+1$ is reached. Hence, at each instant k, the predictions on the interval $[k+1, k+h+1]$ must be calculated by the following equations:

$$\tilde{x}(k+1) \quad \tilde{F}(x(k), \ldots, x(k-d), W_1) \tag{4}$$

$$\begin{aligned} \tilde{x}(k+2) \quad & \tilde{F}(\tilde{x}(k+1), x(k), \ldots, x(k-d+1), W_1) \\ & \vdots \end{aligned} \tag{5}$$

$$\tilde{x}(k+h+1) \quad \tilde{F}(\tilde{x}(k+h), \ldots, \tilde{x}(k+1), x(k), \ldots, x(k-d+h), W_1). \tag{6}$$

During the prediction, the parameter set W_1, remains fixed; it has been estimated in a previous phase by training the neural model given by Equation (2) using the local difference between the measured and predicted values Equation (3).

The main disadvantage of the model given by Equations (4) (6) in the context of multi-step prediction is precisely the way of computing the parameter set W_1,. As has just been said, they have been obtained to minimise the local errors given by Equation (3), this is, for the purpose of one-step prediction. During the training phase, the parameters capture the relation between the available observations of the original time series at the current time, $x(k), \ldots, x(k-d)$, and the next sampling time, $x(k+1)$. However, when the model is acting as a multi-step prediction scheme, a group of the input neurones gather the previous approximated values (see Equations (4) (6)), $\tilde{x}(k+h+1) \quad \tilde{F}(\tilde{x}(k+h), \ldots, \tilde{x}(k+1), x(k), \ldots, x(k-d+h), W_1)$.

As a consequence, the performance of the classical neural model depends on the capability of multilayer feedforward neural networks to approximate perturbations of input patterns. Generally, multilayer feedforward networks filter small perturbations in their inputs; however, when the perturbations in the inputs increase the answer of the network may not be appropriate because the input patterns to the network differ from patterns used during the training procedure.

In addition, an error at some instant $k+1$ is propagated into the next approximations and the network will have to filter more and more high errors. In this case, it is not possible to expect appropriate predictions because errors occurred at some instant are propagated and magnified to future sampling times. Thus, the capability of classical neural models to predict the future of the time series may decrease.

3. Recurrent Neural Model for Multi-step Prediction

The multi-step recurrent neural model proposed in this paper is presented as an alternative to classical neural models when the goal is to predict the future behaviour of the time series. In the previous section, it has been concluded that the predictive capability of the neural model given by Equations (4) (6) could be destroyed since the parameters are determined to solve one-step prediction problem. This implies that the input vectors to the model, $(\tilde{x}(k+1)x(k), \ldots, x(k-d+1), \ldots,$ $(\tilde{x}(k+h), \ldots, \tilde{x}(k+1), x(k), \ldots, x(k-d+h))$, are not used during the training phase and the network will have to filter perturbations on its inputs. In order to guarantee an adequate prediction in the future, the parameters of the model should be estimated using those input patterns and for the purpose of multi-step prediction. The MSRN model proposed in this work is based on these ideas. Basically, it consists of imposing a special learning phase for the purpose of long-term prediction using the predicted outputs as input variables.

In order to build up the MSRN model, a partially recurrent neural network [13] is used. The recurrent network is constructed by starting from a multilayer feedforward neural network and by adding feedback connections from the output neurone to the input layer as it is shown in Figure 1.

The neurones in the recurrent neural network are divided in the input, hidden and output layer, as usual. The input layer is composed by two groups of neurones. The first group acts as the external input to the network gathering the original or measured time series data. The second group is formed by the context neurones, which memorise previous outputs of the network. Introducing the vector $C(k) \quad (C_1(k), \ldots, C_h(k))$ to indicate the activation of context neurones, each component is calculated as,

$$C_i(k) \quad Z^{\ i}(\tilde{x}(k+h+1) \quad \tilde{x}(k+h+1-i)\ i \quad 1, \ldots, h, \tag{7}$$

where $\tilde{x}$ represents the network answer, and $Z^{\ i}$ is an operator that delays by i terms the network output sequence.

The activation of the remaining neurones in the network (hidden and output neurones) follow the same equations that the neurone activation's in the multilayer

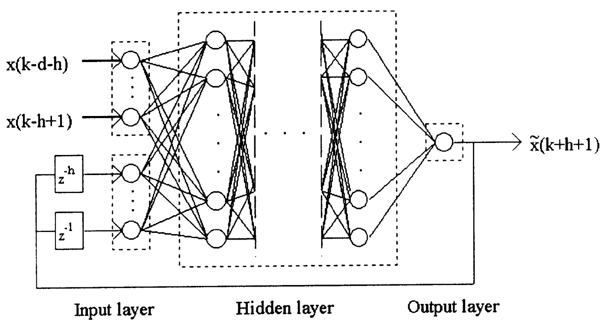

Figure 1. Recurrent network architecture.

feedforward network, this is, the sigmoidal function applied to the weighted sum of the neurone activation's in the previous layer.

Due to the presence of the feedback connections, the training of the partially recurrent neural network is based on a class of learning algorithms, called dynamic backpropagation algorithms [14]. In [13] a dynamic learning rule is inferred for the recurrent architecture. However, since the internal structure of the partially recurrent network is like a feedforward neural network, the training may be also realised using the traditional backpropagation algorithm, as is discussed in [13]. The computational effort required by the dynamic learning rule may not be recommendable in practical applications, principally, when the number of context neurones is high. In these cases, the traditional backpropagation algorithm may provide a suitable convergence.

3.1. DESCRIPTION OF MULTI STEP RECURRENT NEURAL MODEL

When the partially recurrent neural network previously described is used for the prediction task, the number of context neurones and the number of input units receiving the predicted and measured time series values, respectively, depend on the prediction horizon h. The number of context neurones is changing every sampling time, instead of being a fixed number as in other applications [13].

Assuming that the prediction horizon is fixed to h and assuming that at instant k the goal is to predict the time series values at instants $k+1, k+2, \ldots, k+h+1$, the number of input units decrease from $d+1$ to $d+1-h$ and the number of context neurones increase from 0 to h, respectively. Thus, the sequences received by the external inputs and the context neurones, at every instant k, are given by the following sequence:

- The number of context neurones is initialised to zero and the external inputs receive the sequence: $x(k), \ldots, x(k-d)$.
- The future instants $k+i$ for $i \quad 2, \ldots, h+1$ are not real, but simulated. Now the input units receive the vector $x(k+i), \ldots, x(k-d-i)$, and the $(i-1)$-th context neurones memorise the previous $i-1$ outputs of the network, i.e.:

$$
\begin{array}{ll}
C_1(k) & \tilde{x}(k+i-1) \\
 & \vdots \\
C_{i\ 1}(k) & \tilde{x}(k+1)
\end{array}
$$

- After that, the external inputs and the context neurones are resettled.

As was said before, the number of external input units decreases from $d+1$ to $d+1-h$. Hence, it is important to point out that when the prediction horizon, h, is higher than $d+1$, in the last instants of the interval $[k+1, k+h+1]$ all neurones in the inputs are context units and no measured time series value are fed into the network. In that cases, the network do not receive information from the external world and the prediction task becomes more complicate because the network will only use information from itself. Thus, the performance of the neural network may decrease when the prediction horizon is higher than $d+1$.

3.1.1 *Training Procedure*

Below the complete training procedure of the MSRN model for the purpose of multi-step time series prediction is described. At each instant k, starting with $k \quad d$:

Step 1. The number of context neurones is initialised to zero. $d+1$ external input neurones are set receiving the measured values of the time series, $x(k), \ldots, x(k-d)$. The output of the network is given by the following equation:

$$\tilde{x}(k+1) \quad \tilde{F}x(k), \ldots, x(k-d), W_2). \tag{8}$$

Step 2. The number of context neurones is increased in one unit and the number of external units is decreased also in one unit. The context neurone memorises the output of the network, previously calculated, $\tilde{x}(k+1)$. Thus, the prediction at the simulated instant $k+2$ is given by:

$$\tilde{x}(k+2) \quad \tilde{F}(\tilde{x}(k+1), x(k), \ldots, x(k-d+1), W_2). \tag{9}$$

Step 3. Step 2 is repeated until h context neurones are achieved. The outputs of the recurrent model at simulated instants $k+3, \ldots, k+h+1$ are given by the following equations, respectively:

$$\tilde{x}(k+3) \quad \tilde{F}(\tilde{x}(k+1), x(k), \ldots, x(k-d+2), W_2) \tag{10}$$

$$\vdots$$

$$\tilde{x}(k+h+1) \quad \tilde{F}(\tilde{x}(k+h), \ldots, \tilde{x}(k+1), x(k), \ldots, x(k-d+h), W_2). \tag{11}$$

Step 4. At this moment, the parameter set of the model, W_2, is updated. In order to impose a training phase for the purpose of long-term prediction, the learning is based on the sum of the local errors along the prediction horizon, i.e. along the interval $[k+1, k+h+1]$. Hence, the parameter set W_2 is updated following the negative gradient direction of the error function given by:

$$e(k+1) \quad \tfrac{1}{2}\Sigma_i^h \;{}_1(x(k+i+1) - \tilde{x}(k+i+1))^2. \tag{12}$$

In order to avoid the long computational effort required by dynamic backpropagation rules when the prediction horizon is high, the updating of the parameters is realised using the traditional backpropagation learning rule.

Step 5. At this point the time variable k is increased in one unit and the procedure returns to Step 1.

The procedure is repeated for the complete training set until to reach the convergence.

Once the training of the MSRN model is finalised, it can be used for the purpose of multi-step prediction. In this case, the Steps 1, 2, 3 and 5 are carried out.

The structure of the MSRN model (Equations (8) (11)) is identical to the structure of classical neural model when it is used for prediction (Equations (4) (6). However, there exists an important difference between them: the way to obtain the parameter sets of the models. That is, the learning procedure of the system.

As was said before, the parameter set W_1, of the classical neural model is obtained training a multilayer feedforward network and remains fixed during the prediction phase. This means that the parameter set W_1, is updated using the local error measured at each instant (Equation (3)). When the MSRN model previously described is used, the update of the parameters is based on the measured error along the prediction interval $[k+1, k+h+1]$. Thus, the set of parameters W_2 is determined to minimise the error in the future (Equation (12)). The model is trained in such a way that it acts as a multi-step prediction scheme as opposed to classical neural model, which is trained to predict exclusively the next sampling time (one-step prediction scheme).

In other hand, the predicted network outputs are used as patterns during the learning procedure of the recurrent model. Thus, the MSRN model can capture the relationship between the patterns that will be used during the prediction task.

Due to the recurrent structure of the proposed model, errors occurred at the same instant are propagated into the next sampling time as in the classical neural model. However, in the MSRN model the propagated errors are reduced during the training phase because the learning is carried out using the predicted output at previous time steps. Thus, the errors are corrected and better predictions in the future may be expected.

4. Experimental Verification

The experimental verification has been conducted by three different time series: an artificial time series described by the logistic map and two real data time series. The first real series is given by sunspot data and the second one represents the behaviour of the laser measured in a physics laboratory.

For each dynamic system, first we have fixed the structure of NAR model, this is the number of input variables of the models. Second, we have used the forecasting models studied in this paper, the classical feedforward neural model and the MSRN model. Finally the performance of different models is compared and analysed.

In order to measure the ability of neural models to predict the future, the following mean square error, also called prediction error, has been used:

$$E \quad \frac{1}{2N}\Sigma_{k\ 0}^{N\ h}(x(k+h+1)-x(k+h+1))^2, \tag{13}$$

where h is the prediction horizon; $x(k+h+1)$ and $\tilde{x}(k+h+1)$ are the real and prediction values of the time series at instant $k+h+1$; and N is the number of patterns. In the three cases, logistic, sunspot and laser time series, several prediction horizons have been considered.

4.1. PREDICTING THE LOGISTIC MAP

The logistic map is given by the following equation:

$$x(k+1) \quad \lambda x(k)(1-x(k)). \tag{14}$$

When $\lambda \quad 3.97$ and $x(0) \quad 0.5$ the map describes a strongly chaotic time series. That kind of chaotic time series are good test beds for multi-step prediction because an error in previous steps of prediction is strongly propagated in further predictions.

In order to train the neural models, data of the logistic time series from $t \quad 0$ to $t \quad 100$ are used. A different data set values from $t \quad 100$ to $t \quad 500$ has also been used as test patterns.

From Equation (14) it follows that the logistic map at instant $k+1$ depends on the series value at instant k. However, as the ultimate goal in this work is to predict the future in an horizon greater than one, it is suitable to consider NAR models that own more information about the past behaviour of the time series [15]. Thus, in this work the logistic map is represented by the following NAR model:

$$x(k+1) \quad F(x(k), x(k-1), x(k-2)). \tag{15}$$

The parameters of the previous model have been determined to approximate the immediate sampling time, $x(k+1)$, using a multilayer feedforward neural network with 3, 10 and 1 input, hidden and output units, respectively; after that, the model has been used to forecast the logistic map at the prediction horizon $k+h+1$, for $h \quad 0, 1, 2, 3$. The second experiment has consisted of using the recurrent neural network with 10 hidden units and h context neurones, from $h \quad 0$ to $h \quad 3$. The MSRN models have been trained up to complete the convergence using the algorithm described in the Section 3.1.

The prediction errors (Equation (13)) on the training set for both models, classical and MSRN models, can be seen in Table I. The prediction errors over the test set have been also evaluated. The results are shown in Table II.

In these tables, it can be observed that the MSRN model has provided a considerable accuracy on the training and test sets for each prediction horizon. The improvement of MSRN model over classical one is more relevant when the prediction horizon is increased ($h \quad 2$ and $h \quad 3$). In those cases, the predictions provided by the classical neural model are very poor, while the MSRN model is able to obtain convenient long-time predictions. For short prediction horizons ($h \quad 1$), the approximations provided by the classical neural model may be adequate; although even in this case the MSRN obtain the smallest prediction errors.

Table I. Logistic Time Series: Prediction Errors over the Training Data Set

Prediction horizon	Classical Neural Model	MSRN Model
$h = 0$	0.00089	0.00089
$h = 1$	0.00790	0.00310
$h = 2$	0.03925	0.00571
$h = 3$	0.05510	0.00701

Table II. Logistic Time Series: Prediction Errors over the Test Data Set

Prediction horizon	Classical Neural Model	MSRN Model
$h = 0$	0.00152	0.00152
$h = 1$	0.00904	0.00464
$h = 2$	0.04807	0.00784
$h = 3$	0.07827	0.01123

The predictions at two, three and four sampling times (h 1, 2, 3) of the logistic time series provided by the classical neural model and the MSRN model are shown in Figures 2, 3 and 4 respectively. In these figures is easy to appreciate as the adjustment between the real and predicted data is much higher in the recurrent than in the classical model.

4.2. PREDICTING THE SUNSPOT TIME SERIES

The sunspot series was the first time series studied with autoregressive models [16 18], and thus has served as a benchmark in the forecasting literature. The underlying mechanism for sunspot appearances is not exactly known. No first-principles theory exists, although it is known that sunspots are related to other solar activities, such as magnetic field cycles which in turn influence the meteorological conditions on the earth.

Sunspot were first observed around 1610, shortly after the invention of the telescope. The sunspot data have been recorded since 1700. In the most of works related to forecast the sunspot series, the yearly averages of the sunspot data tabulated from around 1700 to 1979 have been used to estimate the parameters of the autoregressive models [6, 19]. In these cases, each input to the model represents the average of the measured values along one year. This may provide to the forecasting model poor information about the behaviour of the sunspot time series because data exhibit strong irregularities during the year. In order to build up forecasting models owing more information about the sunspot series behaviour, we believed it convenient to consider sunspot data corresponding to monthly data instead of yearly data, as usual. In addition, if the inputs to the models represent monthly sunspot values, the models can be used to forecast the behaviour of the sunspot series through the months providing a more comprehensive description of the sunspot series performance.

In our case, the monthly mean of daily relative sunspot numbers from January 1749 to March of 1977 have been used to build up and validate the neural models

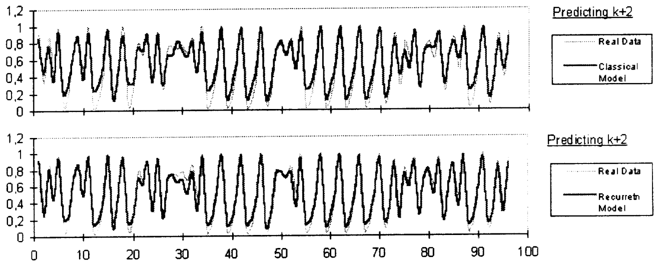

Figure 2. Classical and MSRN models predicting the logistic map at the horizon $h = 1$.

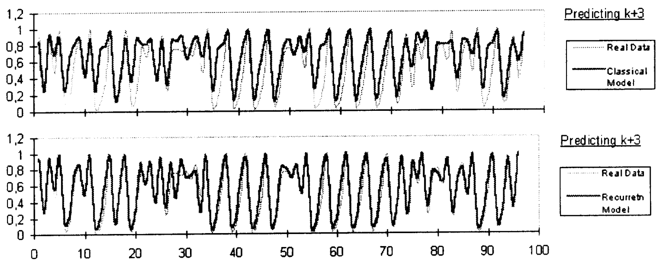

Figure 3. Classical and MSRN models predicting the logistic map at the horizon $h = 2$.

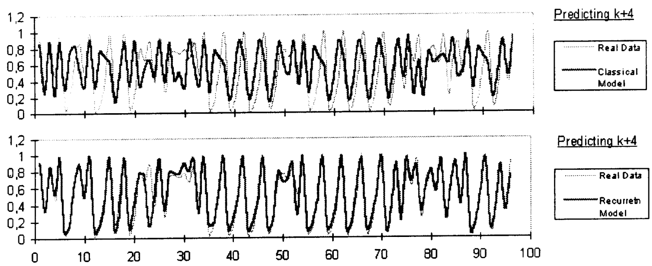

Figure 4. Classical and MSRN models predicting the logistic map at the horizon $h = 3$.

studied in this work. We have used monthly sunspot data from January 1749 through December 1919 for training, and the data from January 1929 to March 1977 have been used to evaluate the generalisation capability of predictive neural models. The data have been normalised in the range [0, 1].

After some simulations, it is concluded that data related with the previous 24 months are enough to predict sunspot data in the future. Thus, the NAR model is given by the following equation:

$$x(k+1) \quad F(x(k), x(k-1), \ldots, x(k-23)). \tag{16}$$

Simulations varying the number of input variables have also been realised. However, the results have proven that the inclusion of number of input variables does not improve the capability of the model to predict the future. Also, if some input variables are removed, the performance of neural models to predict the future might decrease.

Once the structure of NAR model has been fixed, the parameters of the neural models are estimated. First, the functional F in Equation (16) is approximated using a multilayer feedforward neural network with 24 neurones in the input, 30 units in the hidden layer and one unit in the output layer. After that, the model has been used to predict the sunspot time series for several prediction horizons, h 0, h 3, h 7, h 11 and h 17, i.e. one, four, eight, twelve and eighteen months, respectively. The functional F has also been approximated using the recurrent neural network presented in Section 3. The MSRN model has been trained to predict the sunspot series at one, four, eight, twelve and eighteen months in the future; the network had 30 hidden units and the number of context neurones is varying form 0 to 4, 8, 12 and 18, respectively.

The prediction errors (Equation (13)) over the training data obtained with the classical neural model and the MSRN model proposed in this work are presented in Table III. The performance of neural models over the test set is shown in Table IV.

From these results it concludes that the prediction qualities of the MSRN model are better than those provided by the classical model, and the significant differences appears for prediction horizons further than one step in the future.

However, the superiority of the MSRN model is less appreciable over the test set than over the training set. As it is possible to observe in Table IV, the prediction errors on the test set do not decrease as much as for the training set when the MSRN model is used to predict the future. This is mainly due to two factors: first, the output of the network is fed back to the context neurones and second, the measured monthly sunspot data present strong irregularities as it is possible to observe, for example, in Figure 3. These irregularities joined to the fact that previous outputs are used as inputs to the model mean that the model finds difficulties correctly generalising the sunspot data. Both factors contribute to reduce the capability of MSRN models to predict the future on the test set.

In other applications included in this work, logistic and laser time series, it is possible to observe that the performance of the MSRN model on the test set is similar to the performance on the training set. In the sunspot case, the performance is reduced to the nature of the data.

The strong irregularities of data make the generalisation much more difficult, as can be seen in Figures 5 8. However, the MSRN model is able to capture, almost

Table III. Sunspot Time Series: Prediction Errors over the Training Data Set

Prediction horizon	Classical Neural Model	MSRN Model
$h = 0$	0.003262	0.003262
$h = 3$	0.005358	0.002858
$h = 7$	0.007301	0.003382
$h = 11$	0.009790	0.004713
$h = 17$	0.015676	0.008205

Table IV. Sunspot Time Series: Prediction Errors over the Test Data Set

Prediction horizon	Classical Neural Model	MSRN Model
$h = 0$	0.005112	0.005112
$h = 3$	0.009648	0.008376
$h = 7$	0.011775	0.007813
$h = 11$	0.015874	0.010801
$h = 17$	0.025705	0.014645

perfectly, the sudden variations of the solar activity. Otherwise, the classical model tends to approximate the peaks by an average. Obviously, the MSRN generalisation power decreases as higher prediction horizons are considered.

4.3. PREDICTING THE LASER TIME SERIES

The laser time series is an univariate time record of a single observed quantity, measured in a physics laboratory experiment. These data have been chosen because they are a good example of the complicated behaviour that can be seen in a clean, stationary, low-dimensional nontrivial physical system. In addition, the data representing the behaviour of the laser do not present the irregularities observed in the sunspot data. Thus, it may be an appropriate real example in order to prove the ability of the MSRN model as in the training phase as in the test phase.

The laser data were recorded from a Far-Infrared-Laser in a chaotic state. The measurements were made on an 81.5-micron 14NH3 cw (FIR) laser, pumped optically by the P (13) line of an N2O laser via the vibrational aQ(8,7) NH3 transition. The basic laser setup can be found in [20]. The intensity data was recorded by a LeCroy oscilloscope. No further processing happened. The experimental signal to noise ratio was about 300 which means slightly under the half bit uncertainty of the analog to digital conversion.

Approximately 11,000 points referred to the laser time series were provided. However, for the prediction task, not so many points are needed. From the supplied data, two sets   training and test data   have been extracted. The first 1000 points have been used to train the neural models and the test set is composed by the following 1000 points. The aim was to consider data representing different behaviour of the laser time series in order to study the capability of neural models to predict transition states. The data have been normalised in the range [0, 1].

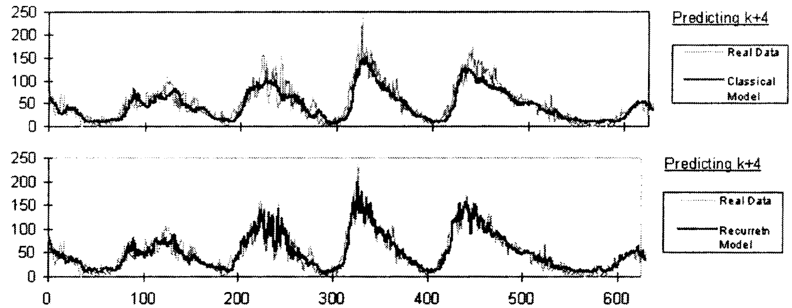

Figure 5. Classical and MSRN models predicting the sunspots data at the horizon $h = 3$.

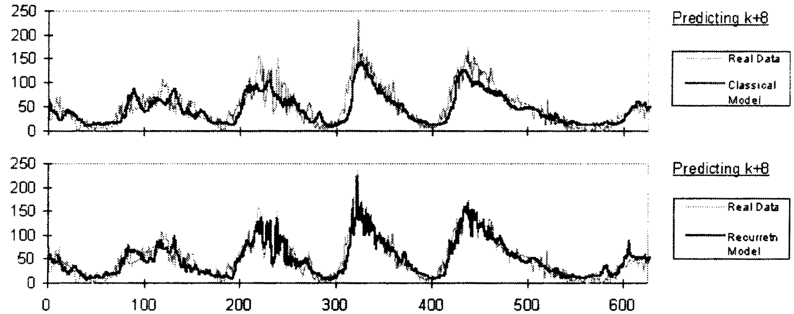

Figure 6. Classical and MSRN models predicting the sunspots data at the horizon $h = 7$.

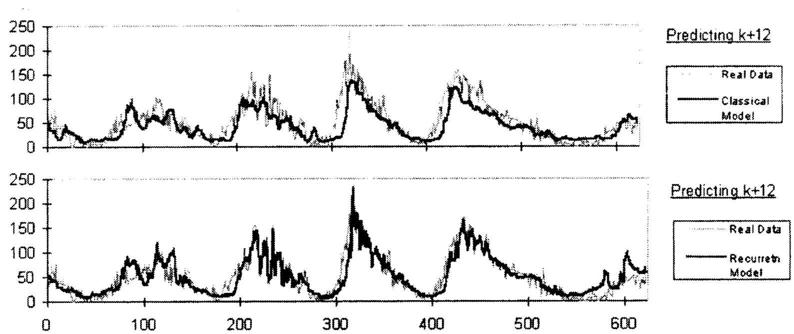

Figure 7. Classical and MSRN models predicting the sunspots data at the horizon $h = 11$.

Before the learning of the neural classical and recurrent models, the first question that arises concerns the choice of the structure of NAR model. In [20, 21] it is shown that pulsation of the laser data more or less follow the theoretical Lorenz model. Taking this assumption into account, it may be possible to explain the

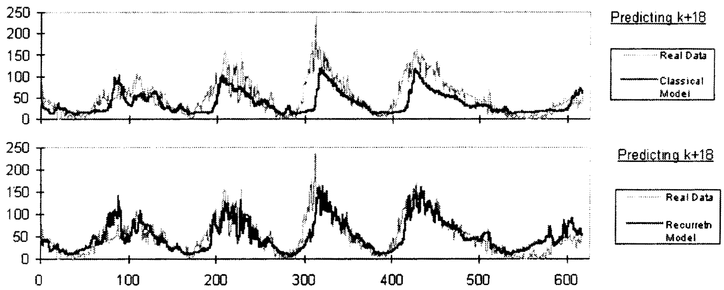

Figure 8. Classical and MSRN models predicting the sunspots data at the horizon $h = 17$.

behaviour of the laser time series using around previous nine values or probably less. In this work, the NAR model adopts the following structure:

$$x(k+1) \quad F(x(k), x(k-1), \ldots, x(k-9)). \tag{17}$$

As in the sunspot application, other structures of NAR models have been also tested. From these simulations it has concluded that model given by Equation (16) might be appropriate when the goal is to predict the laser series values at the future.

As in the previous applications, the functional F has been approximated using a multilayer feedforward neural network and the recurrent neural network. In this case, the multilayer feedforward network had 10 input units, 20 hidden units and one output unit; the learned network has been used to predict the laser value at instants $k + h + 1$, for $h \quad 0, 4, 9, 14$ and 19. In other hand, MSRN models with 20 hidden units have been trained for the purpose of approximating the laser data at the same prediction horizons ($h \quad$ 0, 4, 9, 14 and 19). The prediction errors (Equation (13)) on the training and test data are shown in Tables V and VI, respectively.

As it is possible to observe in these tables, the MSRN model has provided approximations more accuracy than the classical neural model. That can be appreciated as in the training set as in the test set.

For the laser time series prediction, the best performance of the MSRN model is obtained for $h \quad 4, 9$ and 14. When the prediction horizon is fixed to $h \quad 19$, the MSRN model has also provided the best predictions, although it is noticed that the performance decreases. This may be due to the fact that NAR model does not own enough information through the inputs (ten previous sampling times) to forecast a large and ambitious prediction horizon ($h \quad 19$) and the network find it difficult to capture the relationship. In addition, it is necessary to point out that in this case, the MSRN has to memorise the 16 previous predicted values, which makes difficult the learning of the network.

In order to observe the behaviour of the neural models in the prediction task of the laser series, only a part of the test set has been graphically represented. Figures 9 12

Table V. Laser Time Series: Prediction Errors over the Training Data Set

Prediction horizon	Classical Neural Model	MSRN Model
$h = 0$	0.000517	0.000517
$h = 4$	0.003902	0.001509
$h = 9$	0.006098	0.001832
$h = 14$	0.008264	0.003041
$h = 19$	0.011655	0.005560

Table VI. Laser Time Series: Prediction Errors over the Test Data Set

Prediction horizon	Classical Neural Model	MSRN Model
$h = 0$	0.000749	0.000749
$h = 4$	0.006071	0.001891
$h = 9$	0.009019	0.002834
$h = 14$	0.012241	0.005007
$h = 19$	0.016627	0.009762

show the predictions provided by the classical and MSRN models for each prediction horizon.

5. Conclusions

In this work, we have shown that the predictive capability of classical neural models can improve if a special learning phase for the purpose of multi-step prediction is imposed. The classical neural model is trained to solve one-step prediction problems which reduce its performance when a long-term prediction task is formulated.

The MSRN model proposed in this paper can help in improving the prediction accuracy at several sampling times in the future because the parameters of the model have been determined to minimize the (Equation (12)). During the training phase, the parameters of the model capture the relationship between the predictions in the future and the previous predicted values by the network, which seems to be a better approach and contribute to obtain better neural forecasting models.

It is also interesting to point out that the number of the inputs of the models has an important significance on the quality of predictions when a multi-step prediction problem is formulated. If the goal is to predict several instants in the future, the structure of NAR model must have information through the inputs such that the relationship between the patterns can be captured.

We have performed experiments to evaluate the performance of the proposed MSRN model using three time series, the logistic map, sunspot data and the laser time series. The results presented in the experimentation section show the superiority of the MSRN model over the classical one for the three time series used as test cases. The quality of the predictions provided by the MSRN model is different in each case, however, in all of them the improvement of the MSRN model is appreciated

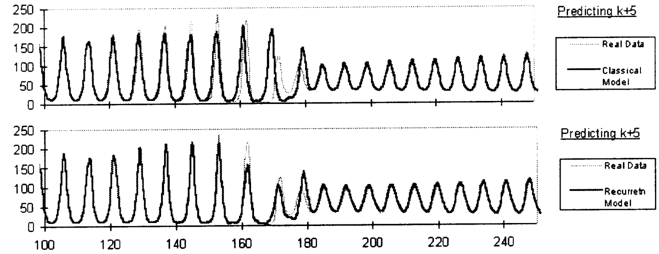

Figure 9. Classical and MSRN models predicting the laser data at the horizon $h = 4$.

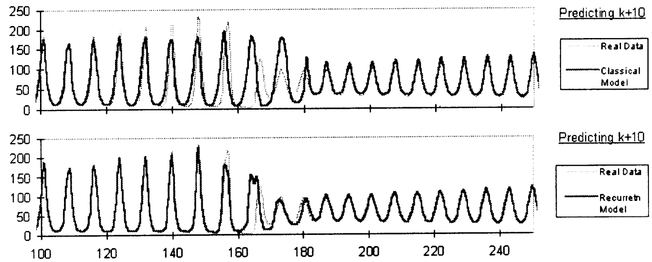

Figure 10. Classical and MSRN models predicting the laser data at the horizon $h = 4$.

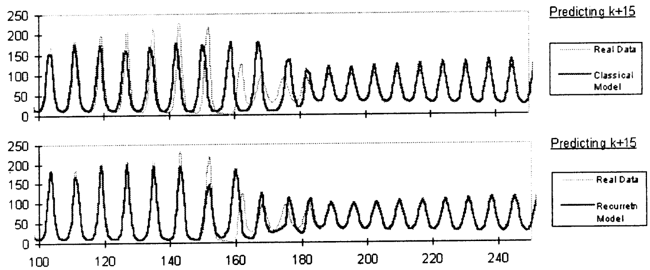

Figure 11. Classical and MSRN models predicting the laser data at the horizon $h = 4$.

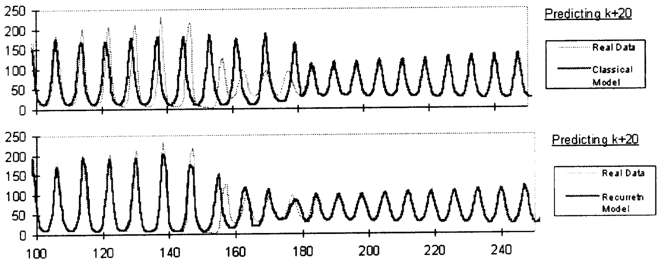

Figure 12. Classical and MSRN models predicting the laser data at the horizon $h = 4$.

References

1. Laepes, A. and Farben, R.: Nonlinear signal processing using neural networks: prediction and system modelling, Technical report, Los Alamos National Laboratory, Los Alamos, NM (1987).
2. Whit, H.: *Artificial Neural Networks: Approximation and Learning Theory*, Oxford, Blackwell (1992).
3. Hoptroff, R.: The principles and practice of times series forecasting and business modelling using neural nets, *Neural Computing and Applications* **1** (1993), 59 66.
4. Wasserman, P.: *Neural Computing: Theory and Practice*, New York, Van Nostrand Reinhold (1989).
5. Kou, C. and Reitsch, A.: Neural networks vs. conventional methods of forecasting, *The Journal of Business Forecasting* (1995), pp. 17 22.
6. Weiggend, A., Huberman, B. and Rumelhart, D.: Predicting the future: A connectionist approach, Technical report, PARC (1990).
7. Cottrell, M., Girard, B., Girard, Y., Mangeas, M. and Muller, C.: Neural modelling for times series: A statistical stepwise method for weight elimination, *IEEE Transactions on Neural Networks* **6**(6) (995), 1355 1363.
8. Yu, H. Y. and Bang, S. Y.: An improved time series prediction by applying the layer by layer learning method to FIR neural networks, *Neural Networks* **10**(9) (1997), 1717 1729.
9. Deco, G. and Schurmann, M.: Neural learning of chaotic system behaviour, *IEICE Transactions Fundamentals* **E77**(A) (1994), 1840 1845.
10. Principe, J. and Kuo, J.: Dynamic modelling of chaotic time series with neural networks, In: R. Lippman, J. Moody and D. S. Touretzky (eds), *Advances in Neural Information Processing System (NIPS)*, (1995).
11. Stagge, P. and Senhoff, B.: An extended Elman net for modelling times series, In: *International Conference on Artificial Neural Networks* (1997).
12. Rumelhart, D., Hinton, G. and Williams, R.: *Parallel Distributed Processing*, Learning internal representations by error propagation, Cambridge, MIT Press (1986).
13. Galván, I. and Zaldivar, J.: Application of recurrent neural networks in batch reactors. Part I: NARMA modelling of the dynamic behaviour of the heat transfer fluid temperature, *Chemical Engineering and Processing* **36** (1997), 505 518.
14. Navendra, K. and Parthasarathy, K.: Gradient methods for the optimization of dynamical systems containing neural networks, *IEEE Transactions on Neural Networks* **2** (1991), 252 262.

15. Galván, I., Alonso, J. and Isasi, P.: *New Frontiers in Computational Intelligence and its Applications*, Improving multi-step time series prediction with recurrent neural modelling, IOS Press Ohmsha (2000).
16. Yule, G.: On a method of investigating periodicities in disturbed series with special reference to Wolfer's sunspot numbers, *Philosophical transactions of the Royal Society of London*, **Series A**(267), (1927).
17. Priestly, M.: *Spectral Analysis and Time Series*, New York, Academic Press (1981).
18. Tong, F.: *Nonlinear Times Series*, Oxford, Clarendon Press (1990).
19. Aussen, A. and Murtag, F.: Combining neural network forecasts on wavelet-transformed time series, *Connection Science* **9**(1) (1997), 113–121.
20. Huebner, U., Abraham, N. and Weiss, C: Dimensions and entropies of chaotic intensity pulsations in a single-mode far-infrared NH3 laser, *Physical Review* **A**(40) (1989a), 6354–6360.
21. Huebner, U., Klische, W., Abraham, N. and Weiss, C.: *Coherence and Quantum Optics VI*, Comparison of Lorentz-like laser behaviour with the Lorentz Model, New York, Plenum Press (1989b).