Stochastic Modelling Hints for Neural Network Based Time Series Predictions[1]

Radu Drossu Zoran Obradović[2]

rdrossu@eecs.wsu.edu zoran@eecs.wsu.edu

School of Electrical Engineering and Computer Science

Washington State University, Pullman, Washington, 99164-2752

Abstract

The objective of this study is to investigate the relationship between stochastic and neural network approaches to time series modelling. Experiments on both a complex real life prediction problem (entertainment video traffic series) as well as on an artificially generated nonlinear time series on the verge of chaotic behavior (Mackey-Glass series) indicate that the initial knowledge obtained through stochastic analysis provides a reasonably good hint for the selection of an appropriate neural network architecture. Although not necessarily the optimal, such a rapidly designed neural network architecture performed comparable or better than more elaborately designed neural networks obtained through expensive trial and error procedures.

Keywords: time series, non-stationary process, ARMA modelling, neural network modelling, prediction horizon.

[1] Correspondence: Z. Obradović, phone: (509) 335-6601, Fax: (509) 335-3818

[2] Research sponsored in part by the NSF research grant NSF-IRI-9308523.

INTRODUCTION

A *time series* x_t can be defined as a random (or nondeterministic) function x of an independent variable t [6]. Its main characteristic is that its future behavior can not be predicted exactly as in the case of a deterministic function of t. However, the behavior of a time series can sometimes be anticipated by describing the series through probabilistic laws. Commonly, time series prediction problems are approached either from a stochastic perspective [1] or, more recently from a neural network perspective [9, 12]. Each of these approaches has advantages and disadvantages: the stochastic methods are usually fast, but of limited applicability since they commonly employ linear models, whereas the neural network methods are powerful enough, but the selection of an appropriate architecture and parameters is a time consuming trial and error procedure. At first glance it might seem that there isn't any direct relationship between time series and neural networks, but there are at least two reasons that might make neural networks very attractive for modelling time series:

1. The theoretical work shows that neural networks are computationally very powerful models. For example, a multilayer perceptron with a single layer of hidden processing units using sigmoidal nonlinearities is powerful enough to uniformly approximate almost any arbitrary continuous function on a compact domain [3]. In addition to the ability of representing complex nonlinear functions, neural networks can effectively construct approximations for those functions by learning from examples. This ability to approximate complex input-output mappings makes them attractive in practical applications where traditional computational structures have performed poorly (e.g. ambiguous data or large contextual influence).

2. There is a direct relationship between the fundamental stochastic models (autoregressive and autoregressive-moving average) for time series and feedforward and recurrent neural network models, as explained in the next section.

A comparative study of stochastic and neural network techniques for time series prediction with respect to number of data samples and prediction horizon is performed in [11]. The combination of stochastic and neural network techniques in a hybrid system for improving prediction accuracy, or the use of some stochastic prior knowledge of the underlying process for configuring the neural network are topics that deserve further consideration.

This paper proposes the use of stochastic modelling both for providing initial *hints* for selecting an appropriate neural network architecture (number of external inputs and number of context inputs) and data sampling rate, as well as a means of comparison to neural network prediction techniques.

The proposed approach is to perform an initial stochastic analysis of the data and to choose an appropriate neural network model accordingly. The motivation for this approach is that the linear stochastic modelling is more cost effective than the selection of a neural network architecture by a trial and error procedure. The objective of this study is not to obtain "the optimal" neural network architecture for a given problem, but to provide rapidly an architecture with close to optimal performance. Since the hint is obtained from a linear model, for more complex problems the neural network might be over-dimensioned (similar performance might be obtained using a smaller machine and less training data). However, the exhaustive trial and error procedure involved for determining such an optimal machine could be costlier than the stochastic hint-based alternative.

The evaluation of the stochastic hints for neural network prediction is performed in the context of different prediction objectives, of "clean" and "noisy" data, as well as of data sets of various complexity. An additional issue addressed in the paper is whether neural networks different than the ones suggested by the stochastic model can lead to improved prediction accuracy as compared to the suggested ones.

STOCHASTIC MODELS AND NEURAL NETWORKS FOR TIME SERIES

A stationary process can be described as a process whose characteristic parameters don't change over time. The autoregressive moving average model of orders p and q (ARMA(p,q)) of a stationary stochastic process x_t can be described by the following equation:

$$x_t = \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + \ldots + \varphi_p x_{t-p} + a_t + \psi_1 a_{t-1} + \psi_2 a_{t-2} + \ldots + \psi_q a_{t-q}, \qquad (1)$$

where $x_{t-1}, x_{t-2}, \ldots, x_{t-p}$ represent the process values at p previous time steps, $a_t, a_{t-1}, \ldots,$ a_{t-q} are realizations of a random process, usually emanating from a normal (Gaussian) distribution with mean zero and variance σ_a^2, and $\varphi_1 \ldots \varphi_p, \psi_1 \ldots \psi_q$ are the model parameters. Commonly, x_t doesn't represent the actual physical process, but the process with the mean removed. The AR(p) and MA(q) processes are special cases of the ARMA(p,q) model, obtained by setting $q = 0$ and $p = 0$, respectively. The most general stochastic model, ARIMA (autoregressive integrated moving average) assumes that the process exhibits a "homogeneous" non-stationary behavior that can be eliminated through a suitable discrete differentiation pre-processing.

A natural generalization of the linear AR and ARMA models to the nonlinear case leads to the NAR model

$$x_t = h(x_{t-1}, x_{t-2}, \dots, x_{t-p}) + a_t, \tag{2}$$

and the NARMA model

$$x_t = h(x_{t-1}, x_{t-2}, \dots, x_{t-p}, a_{t-1}, \dots, a_{t-q}) + a_t, \tag{3}$$

where h is an unknown smooth function.

The NAR and NARMA models are very complex, thus making them unsuitable for real life applications. Feedforward and recurrent neural networks have been proposed [2, 12] for simulating nonlinear AR and ARMA models respectively (see Fig. 1). A conditional mean predictor [2] for the NAR model can be approximated as:

$$\hat{x}_t = \hat{h}(x_{t-1}, \dots, x_{t-p}) = \sum_{i=1}^{m} W_i f(\sum_{j=1}^{p} w_{ij} x_{t-j} + \theta_i), \tag{4}$$

where f represents a non-linear, smooth, bounded function (usually called *activation* or *transfer* function). This approximation of the NAR model corresponds to the *feedforward* neural network obtained by disconnecting the context inputs $\hat{a}_{t-1} \dots \hat{a}_{t-q}$ from Fig. 1. Similarly, a predictor for an invertible [1] NARMA model can be approximated as:

$$\hat{x}_t = h(x_{t-1}, \dots, x_{t-p}, \hat{a}_{t-1}, \dots, \hat{a}_{t-q}) = \sum_{i=1}^{m} W_i f(\sum_{j=1}^{p} w_{ij} x_{t-j} + \sum_{j=1}^{q} w'_{ij}(x_{t-j} - \hat{x}_{t-j}) + \theta_i), \tag{5}$$

with $\hat{a}_k = x_k - \hat{x}_k,\ \ j = \overline{t-q, t-1}$, corresponding to the *recurrent* neural network from Fig. 1. Most commonly, the parameters W_i, w_{ij} and w'_{ij} (weights) are estimated from examples by a gradient descent error minimization technique known as backpropagation learning [10]. This is also the learning method employed in our experiments.

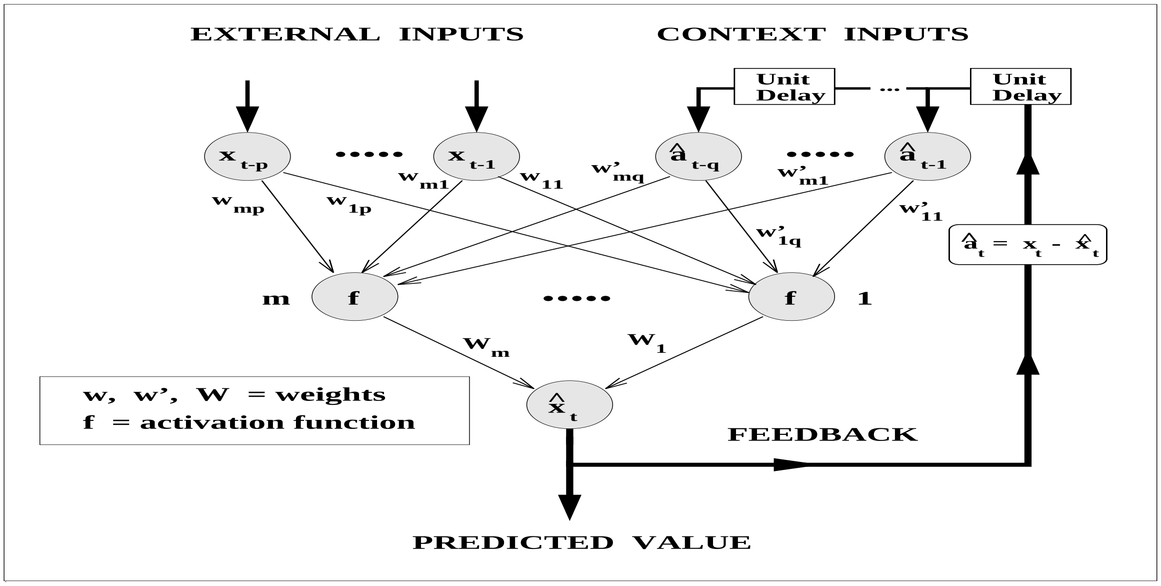

Figure 1: Stochastic Model Approximation

STOCHASTIC AND NEURAL NETWORK PREDICTIONS

Both the stochastic and the neural network predictions can be described in a procedural fashion as follows:

```
perform initial data pre-processing          /* step 1 */
repeat
  perform model identification               /* step 2 */
  perform parameter estimation on data set 1  /* step 3 */
  perform model validation on data set 2      /* step 4 */
until model suitable
perform prediction                           /* step 5 */
```

However, the individual steps are different for stochastic and neural network predictors, as explained below.

A. Stochastic Models.

A1. The data pre-processing step usually comprises a *smoothing* and possibly a *stationarization.* In practice a logarithmic transformation of the original positive valued series is commonly performed for smoothing ($y_t = log(x_t)$) and a first or second order discrete differentiation for stationarization ($w_t = y_t - y_{t-1}$ or $w_t = y_t - 2y_{t-1} + y_{t-2}$ respectively).

A2. The model identification step selects a model type (AR, MA or ARMA), as well as corresponding model orders.

A3. The parameter estimation step uses a first data set for performing most commonly a maximum likelihood estimation of the model parameters. In the case of pure AR models, Burg's method or the Modified Covariance method are also applied [7].

A4. The model validation step performs an adequacy check of the model by using Akaike's final prediction error and information criterion, Bode plots, pole-zero cancellation, as well as a residual analysis of prediction errors on the second data set in both time and frequency domain.

B. Neural Network Models.

B1. The data pre-processing step can be performed as in the case of the stochastic models. Alternatively, it can be completely omitted or, more commonly, it can encompass a linear transformation of the data so as to fit it to a convenient range.

B2. The model identification step selects a neural network architecture (feedforward or recurrent), layer structure (number of layers and number of units per layer), as well as learning parameters (learning rate, momentum and tolerance).

B3. The parameter estimation step encompasses the neural network training on a first data set (training set), in which the network weights are modified according to a given learning technique (backpropagation in our case).

B4. The model validation performs a residual analysis of prediction errors in time and frequency domain on the second data set (test set).

For both stochastic and neural network models the residual analysis of prediction errors should comprise at least the computation of:

- error mean $\mu = \frac{1}{n}\sum_1^n \hat{a}_i$,
- root mean squared error $RMSE = \sqrt{\sum_1^n \hat{a}_i^2/n}$,
- coefficient of determination $r^2 = 1 - RMSE^2/VAR[x]$,
- histogram,
- power spectra,

where the $\hat{a}_i$'s stand for prediction errors and $VAR[x]$ for the variance of the actual data.

For a good predictor, the residuals should be normally distributed (the normality of the distribution can be indicated by a histogram resembling the Normal distribution, as well as by a "flat spectrum") with μ close to 0, small RMSE and r^2 close to 1. It is common practice to test the prediction accuracy on either the same data set used for model validation or on a third data set, also known as *cross-validation set.*

The actual prediction can deal with either predicting a characteristic process parameter for just the next time step (prediction horizon one), or with predicting a parameter several steps ahead (prediction horizon larger than one). Prediction horizons larger than one are useful in numerous real life problems like power consumption predictions, car sales predictions or Internet traffic predictions.

EXPERIMENTAL RESULTS

The hint explored throughout the experiments is whether the order of the most appropriate stochastic model provides an indication of the appropriate number of neural network inputs: feedforward neural network with p or $p+1$ external inputs for an AR(p) process or recurrent neural network with p or $p+1$ external and q context inputs for an ARMA(p,q) process. The choice of p or $p+1$ external inputs from the hint depends on whether the stochastic pre-processing step is done without or with discrete differentiation respectively. Since the stochastic modelling of the data sets considered in the experiments indicated AR(p) models as the most appropriate, all the analyzed neural network architectures were of feedforward type.

The validity of the stochastic modelling hint for selecting an adequate neural network architecture was tested in the context of the following prediction problems:

1. Prediction with horizon one on data un-corrupted by noise.

2. Influence of increased prediction horizon.

3. Influence of noise corruption.

4. Influence of increased complexity of the data set.

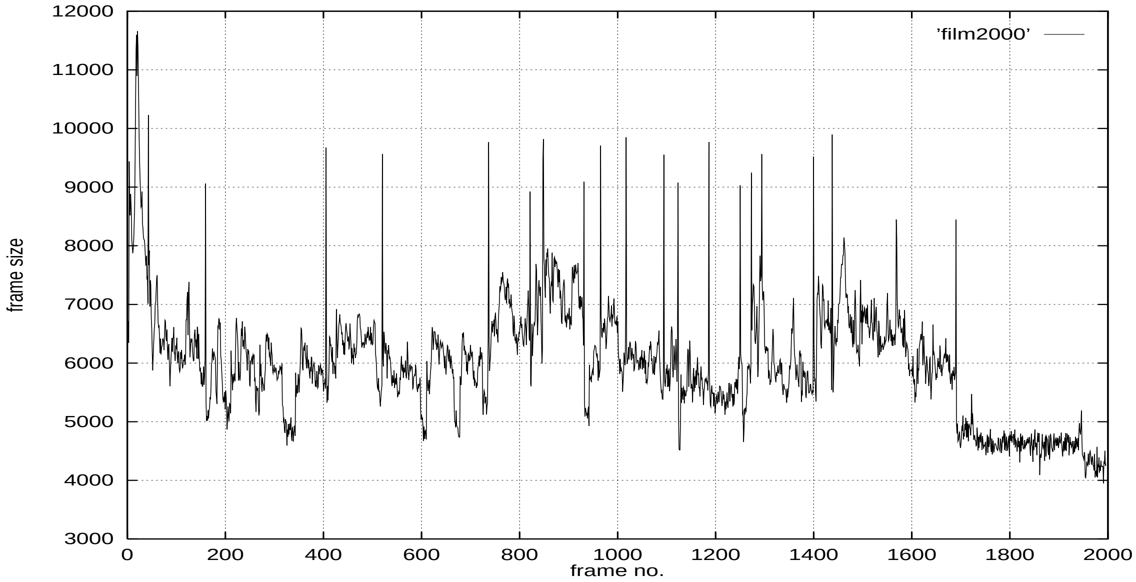

Figure 2: Entertainment Video Traffic Data

The first data set used in the experiments consisted of a real life, compressed, entertainment video traffic data used in an ATM (Asynchronous Transfer Mode) network, in which each sample represents the size of a corresponding compressed video frame [4]. The difficulty associated with this data set is the *non-stationarity* (data distribution changes over time), as well as the existence of "outliers" (values very different from neighboring ones). The data set consisted of 2000 samples (shown in Fig. 2), of which the first 1000 were used for parameter estimation and the last 1000 for model validation. The actual predictions were done using the same 1000 samples used for model validation and not a separate cross-validation set.

The second data set is artificially generated and is obtained by a delay differential equation (also known as Mackey-Glass series):

$$\frac{dx(t)}{dt} = \frac{Ax(t-\tau)}{1+x^{10}(t-\tau)} - Bx(t)$$

Experiments were performed for $A = 0.2, \quad B = 0.1, \quad \tau = 17$, case in which the system

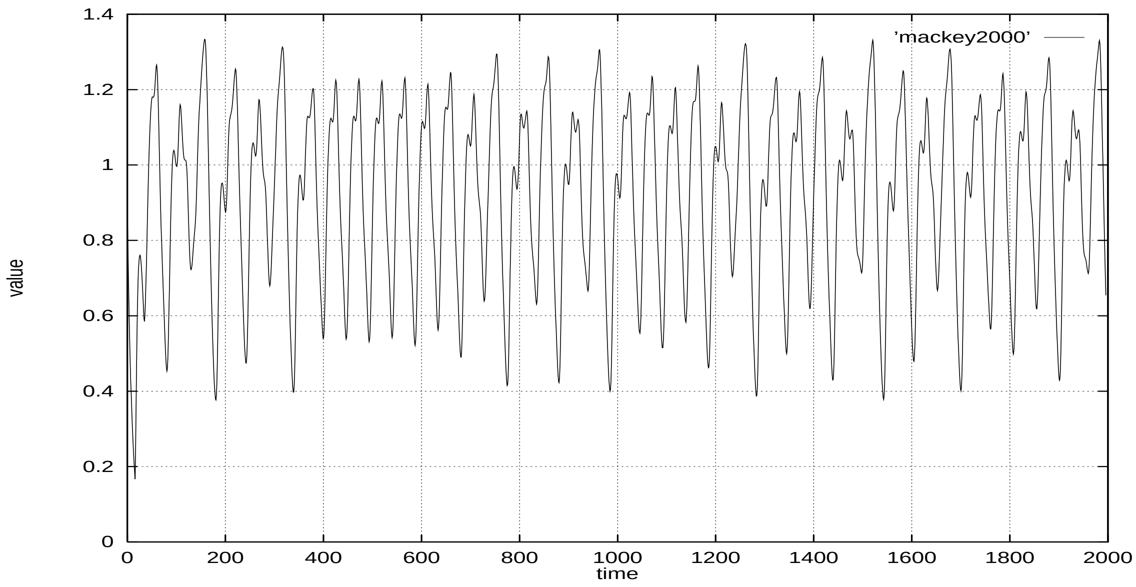

Figure 3: Mackey-Glass Data

exhibits chaotic behavior. The difficulty associated with this data set is the high *nonlinearity*. The data set consisted of 3000 samples (the first 2000 shown in Fig. 3), out of which various experiments used either the first 1000 or the first 2000 values for parameter estimation (training), and the following 1000 for model validation (testing).

Prediction problems 1, 2 and 3 were analyzed in the context of the entertainment video traffic data set, whereas the influence of increased complexity was analyzed in the context of the Mackey-Glass series.

Prediction with Horizon One on Data Un-corrupted by Noise.

The data pre-processing step for the stochastic model encompassed both a logarithmic smoothing and a first order differentiation for stationarization purposes, whereas it included just a linear scaling transformation in the case of the neural network, since this yielded far better results than the previously mentioned transformations. The residual analysis results

Model	μ	RMSE	r^2
AR(5)	2.313	462.664	0.749
NN 6-1	-359.283	592.596	0.593
NN 6-1 *	-105.982	482.995	0.730
NN 6-3-1	-365.196	596.864	0.587
NN 6-6-1	-380.949	606.855	0.573
NN 6-6-1 *	-97.991	477.913	0.736
NN 8-8-1	-386.014	611.476	0.569
NN 4-4-1	-406.859	628.954	0.539
NN 6-6-6-1	-417.531	637.428	0.528

Table 1: Stochastic and NN Performance for Prediction Horizon 1
("*" stands for bias removal)

obtained for the most appropriate stochastic model, as well as for the most significant neural networks are presented in Table 1.

The residual analysis indicated an AR(5) model as the most appropriate, thus suggesting the use of a feedforward neural network with 5 inputs. The first order discrete differentiation has been accounted for by allowing an additional neural network input, hence leading to a feedforward neural network with 6 inputs.

All the neural network results were obtained using a learning rate $\eta = 0.05$, a momentum term $\alpha = 0.7$, a tolerance $t = 0.02$ and training for 4000 epochs. The only transformation performed on the data was a linear transformation for mapping the data range to the $(0.4, 0.6)$ range. All the results presented for the neural networks were averaged over 10 runs with different initial random weights. The notation NN x-y_1-y_2-z stands for a neural network with x external inputs, y_1 and y_2 units in the first and second hidden layer respectively and z output units.

Varying the hidden layer size suggested that a number of hidden units equal to the number of inputs is an appropriate choice. Although the 6-1 architecture has the smallest RMSE and the largest r^2 of all neural network models, spectral analysis and histogram of residuals indicated the 6-6-1 architecture to be more appropriate. They indicated also that the 6-6-1 neural network is more adequate than the AR(5) model, despite the corresponding RMSE and r^2 values from Table 1.

By comparing the r^2 value for the AR(5) model to that of the 6-y_1-1 and the 6-y_1-y_2-1 neural networks (lines 1, 5 and 9 in Table 1), it can be seen that the AR(5) model still performs better, although its histogram and spectrum of the residuals didn't indicate a normal distribution. The neural network predictor appears to be biased (mean of residuals far from zero), thus suggesting the need for the following bias removal post-processing procedure:

(1) train the neural network as before; **(2)** perform a cross-validation test; **(3)** compute the mean of the residuals (prediction errors) for the cross-validation set; **(4)** subtract the computed mean from each actual prediction.

For the 6-6-1 architecture this procedure yielded μ=-97.991, RMSE=477.913, r^2=0.736 using the training data as a cross-validation set. These results, as well as the histogram and the spectrum, indicated a significant improvement without deteriorating the "normality" of the residuals. They also confirmed that the 6-6-1 is indeed a better choice than the 6-1 architecture (compare lines 3 and 6 from Table 1).

The conclusions drawn from these experiments are:

1. The hint provided by the stochastic analysis of using 6 input units for neural network modelling of the underlying process seems to be appropriate. Increasing the number of inputs to 8 did not improve prediction accuracy, while decreasing the number of inputs

to 4 deteriorated the prediction accuracy considerably.

2. Even when applying the bias removal post-processing, the performance of the most appropriate AR model is still better than that of the corresponding neural network (although not significantly better).

Influence of Increased Prediction Horizon.

In the previous experiment, the performance of the hint-based neural network was still not better than that of the corresponding stochastic model. Thus, different related problems of increased complexity were experimented with, as the computationally more powerful neural network model is expected to yield better performance in those cases [11]. One of these more difficult problems that is quite important in practice is prediction for an increased horizon.

For both stochastic and neural network models, the data pre-processing step was performed similarly as in the case of the prediction horizon one. For a larger prediction horizon different sampling rates can be employed, which makes the trial and error neural network architecture selection even more impractical. Consequently, in this experiment the choice of an appropriate sampling rate based on the stochastic modelling hint is analyzed.

In addition, similar to the prediction horizon one hint, it is explored whether an appropriate AR(p) model indicates the use of a feedforward neural network with $p + 1$ external inputs.

The same entertainment video traffic data was used for experimentation, but now with prediction horizon ten (the tenth step ahead process value is predicted). To predict the process, $\hat{x}$, at time step $t + 10$ using k process values up to time t, different sampling rates (divisors of the prediction horizon) are considered:

- sampling rate 1, where the k previous process values are $x(t), x(t-1), x(t-2), \ldots, x(t-k+1)$;
- sampling rate 2, where the k previous process values are $x(t), x(t-2), x(t-4), \ldots, x(t-2*(k-1))$;
- sampling rate 5, where the k previous process values are $x(t), x(t-5), x(t-10), \ldots, x(t-5*(k-1))$;
- sampling rate 10, where the k previous process values are $x(t), x(t-10), x(t-20), \ldots, x(t-10*(k-1))$.

For horizon h larger than one, the prediction can be done either in a *direct* or in an *incremental* fashion. In the direct approach, the neural network is trained to predict directly the h-th step ahead without predicting any of the intermediate $1, \ldots, h-1$ steps. In the incremental approach, the neural network predicts all the intermediate values up to h steps ahead by using the previously predicted values as inputs for predicting the next value. Since the incremental approach lead to an undesirable accumulation of error for our data set, the presented results are obtained by using the direct approach. All neural network results are, as for the previous set of experiments, averaged over 10 runs with different initial random weights. All results were obtained using a learning rate $\eta = 0.01$, a momentum term $\alpha = 0.7$, a tolerance $t = 0.02$ and training for 6000-10000 epochs.

The most appropriate AR models obtained for different sampling rates, as well as the corresponding neural network models are presented in Table 2. The stochastic models indicate a sampling rate 1 as the most appropriate. The neural network results confirm the

Sampling	Model	μ	RMSE	r^2
1	AR(5)	-1.635	534.640	0.667
1	NN 6-6-1	-191.596	623.589	0.556
2	AR(4)	4.042	544.573	0.442
2	NN 5-5-1	-177.545	557.727	0.406
5	AR(5)	1.759	928.670	0.131
5	NN 6-6-1	24.472	880.784	0.222
10	AR(4)	31.118	855.588	0.361
10	NN 5-5-1	10.223	1004.144	0.333

Table 2: Stochastic and NN Performance for Different Sampling Rates

hint drawn from the stochastic analysis, according to which a sampling rate 1 is the most appropriate. Except for the case of the 5-5-1 neural network applied for sampling rate 10, all the neural networks employed the bias removal post-processing.

Table 3 summarizes the results obtained for the best stochastic model, as well as for different representative neural networks for a sampling rate 1. The neural networks employed a similar bias removal post-processing as in the case of prediction horizon one. The table indicates that the neural network having 6 inputs yielded the best prediction, this being consistent with the hint provided by the stochastic modelling (allowing again an additional external input as compared to the most appropriate AR(5) model to account for the first order differentiation). Neural network architectures much larger than the ones indicated were also experimented with, but their performance was poor (the coefficient of determination, r^2 was 0.160 for a 20-20-1 architecture and -0.025 for a 30-30-1 architecture respectively).

The conclusions that could be drawn from these experiments are:

1. The data sampling rate indicated by the stochastic models seems to be appropriate also for the neural network models.

Model	μ	RMSE	r^2
AR(5)	-1.635	534.640	0.667
NN 3-3-1	-272.557	691.800	0.446
NN 4-4-1	-192.799	625.247	0.552
NN 5-5-1	-232.390	664.002	0.498
NN 6-6-1	-191.596	623.589	0.556
NN 7-7-1	-201.559	636.116	0.540

Table 3: Stochastic and NN Performance for Prediction Horizon 10
with Sampling Rate 1

2. The hint provided by the stochastic analysis regarding the number of external inputs is effective also for larger horizons.

3. The performance of the AR models is still better, this indicating that the complexity of the data sets might still be too low.

Influence of Noise Corruption.

Another practical problem is constituted by the prediction in a noisy environment. For such an experiment, an additive Gaussian noise is introduced to the entertainment video traffic data and predictions with horizon one are performed. The first experiment used un-corrupted (noise-free) data for parameter estimation and data with 50% noise for model validation. The noise level is computed as a ratio of the standard deviation of the additive noise and the standard deviation of the un-corrupted data.

For both stochastic and neural network models the data pre-processing step was as in the previous noise-free experiments. All neural network results were obtained using a learning rate $\eta = 0.01$, a momentum term $\alpha = 0.7$, a tolerance $t = 0.02$, training for 10000-20000 epochs and averaged over 10 runs with different initial random weights. In addition, the

Noise Level	Model	r^2
50%	AR(5)	0.359
50%	NN 3-3-1	0.382
50%	NN 4-4-1	0.413
50%	NN 5-5-1	0.421
50%	NN 6-6-1	0.430
50%	NN 7-7-1	0.427
50%	NN 8-8-1	0.429
50%	RWALK	0.0655
80%	AR(5)	0.095
80%	NN 6-6-1	0.232
80%	RWALK	-0.293

Table 4: Stochastic and NN Performance on Noisy Data

previously discussed bias removal post-processing was applied.

The results for the most appropriate stochastic model, as well as for some of the representative neural network models are presented in Table 4. It can be observed that the stochastic model is outperformed by the corresponding 6-6-1 neural network. Table 4 also includes experimental results for an 80% noise level in the model validation data. Again, the neural network outperforms the corresponding stochastic model, but this time more significantly. For comparison purposes Table 4 includes also the results obtained for a random walk (RWALK) predictor on 50% and 80% noise corrupted data. A random walk predictor is a trivial predictor in which the next predicted value is identical to the last observed process value. The very low values for the coefficient of determination obtained for the random walk predictor as compared to both stochastic and neural network models show clearly that both models are capable of extracting useful information even in the conditions of such a high noise level.

The conclusions drawn from this experiment are:

1. The neural network corresponding to the most appropriate stochastic model has a better performance than the other tested neural networks.

2. In this problem the performance of the neural network is better than that of the corresponding stochastic model.

Influence of Increased Complexity of Data Set.

The final experiment uses a well known benchmark problem, the Mackey-Glass time series. In accordance to previously published results [8], a sampling rate six is used for predicting six steps ahead.

The pre-processing step for the stochastic model included either both logarithmic smoothing and first order differentiation (yielding a most appropriate AR model of order 24) or just the logarithmic smoothing (leading to a most appropriate AR model of order 29). Experiments without first order differentiation were performed in this case since the data was apparently "stationary".

In the neural network models, the pre-processing was similar to the previous experiments. The neural network results were obtained with a learning rate $\eta = 0.01$, a momentum term $\alpha = 0.7$, a tolerance $t = 0.02$ and training for 40000 epochs. In contrast to the previous experiments, the neural network results were obtained as an average over three runs with different initial random weights, since training was computationally too expensive for ten runs.

In addition to neural network learning performed on the training set used for stochastic modelling (1000 examples), additional experiments were performed using a twice larger training set (2000 examples). The motivation for these additional experiments was the con-

Model	Training Set Size	r^2
AR(24)	1000	0.751
NN 25-25-1	1000	0.914
NN 25-25-1	2000	0.925
AR(29) *	1000	0.767
NN 29-29-1	1000	0.917
NN 29-29-1	2000	0.929
NN 4-10-10-1	1000	0.912
NN 4-10-10-1	2000	0.936

Table 5: Stochastic and NN Performance on Mackey-Glass Data
("*" stands for no differentiation)

cern that the original 1000 training examples might not be enough to fit the parameters (weights) of the neural networks corresponding to the AR(24) and AR(29) stochastic models. Instead of comparing the stochastic hint-based neural networks to the neural networks of somewhat different architectures obtained through a trial and error process as previously, here the results are compared versus an earlier reported "optimal" neural network topology with 4 inputs and two hidden layers of 10 units each [8].

The conclusions drawn from this experiment (reported in Table 5) are:

1. A differentiation pre-processing step in the stochastic modelling for this time series is not needed.

2. The performance of the neural networks is much better as compared to the most appropriate stochastic model.

3. The stochastic hint-based neural networks performed similar to the "optimal" neural network architecture, further supporting the hint-based design approach.

4. Although the hint-based neural network might appear to be highly over-dimensioned

as compared to “the optimal” network, training and prediction in an actual hardware implementation would be faster for the hint-based architecture since it contains a single layer of hidden units as compared to two such layers in the “optimal” architecture.

FINAL REMARKS AND FURTHER RESEARCH

This study tested whether a stochastic analysis can provide any initial knowledge for a neural network time series prediction. This issue was analyzed in the context of fairly difficult time series prediction problems (entertainment video traffic and Mackey-Glass).

Although neural networks are computationally more powerful models than the linear stochastic models, there are important real life problems in which a simple stochastic model can outperform neural networks (see first two sections of Experimental Results). Anyhow, there are many problems in which the computational power of neural networks is beneficial (see last two sections of Experimental Results). Consequently, when predicting time series, both methodologies should be considered before deciding upon the most appropriate prediction model.

Experiments suggested that a neural network architecture selected according to the hint provided by the stochastic analysis performs comparable or better than neural network architectures determined through a trial and error procedure. It is important to emphasize that the goal of the proposed hint-based approach is not to find “the optimal” neural network architecture for a given problem but to provide rapidly (after a fast stochastic analysis) a neural network architecture with close to optimal performance. Further research is needed to explore the validity of these hints to other time series prediction problems as well as to extend the study from AR to ARMA modelling hints (that would indicate the choice of a recurrent neural network).

We strongly believe that potentially better results are achievable by integrating prior knowledge and neural network learning. For example, a successful integration of expert system rules and neural network classifiers is presented in [5]. The approach proposed in this study is a way of incorporating prior knowledge into neural network systems for time series prediction. As a further level of integration, our current research considers the use of stochastic modelling hints with additional sources of prior knowledge (e.g. embedding or chaotic dimension) for neural network based time series predictions.

References

[1] G. Box and G. Jenkins, "Time Series Analysis. Forecasting and Control," Prentice Hall, 1976.

[2] J. T. Connor et al., "Recurrent Neural Networks and Robust Time Series Prediction," *IEEE Transactions on Neural Networks*, Vol. 5, No. 2, March 1994, pp. 240-254.

[3] G. Cybenko, "Approximation by Superpositions of a Sigmoidal Function," *Mathematics of Control, Signal, and Systems*, Vol. 2, 1989, pp. 303-314.

[4] R. Drossu, T. V. Lakshman, Z. Obradović and C. Raghavendra, "Single and Multiple Frame Video Traffic Prediction Using Neural Network Models," *Proc. IEEE Networks 94*, Madras, India, 1994.

[5] J. Fletcher and Z. Obradovic, "Combining Prior Symbolic Knowledge and Constructive Neural Networks," *Connection Science: Journal of Neural Computing, Artificial Intelligence and Cognitive Research*, Vol. 5, Nos. 3-4, 1993, pp. 365-375.

[6] G. Jenkins and D. Watts, "Spectral Analysis and Its Applications," Holden-Day, 1968.

[7] S. M. Kay, "Modern Spectral Estimation. Theory and Application," Prentice Hall, 1988.

[8] A. Lapedes and R. Farber, "Nonlinear Signal Processing Using Neural Networks: Prediction and System Modelling," *Technical Report*, LA-UR87-2662, Los Alamos National Laboratory, Los Alamos, New Mexico, 1987

[9] K. S. Narendra, "Adaptive Control of Dynamical Systems Using Neural Networks," in *Handbook of Intelligent Control. Neural, Fuzzy, and Adaptive Approaches*, D. A. White and D. A. Sofge eds., Van Nostrand Reinhold, 1992, pp. 141-184.

[10] D. E. Rummelhart et al., "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing*, Vol. 1, MIT Press, 1986, pp. 318-362.

[11] Z. Tang et al., "Time Series Forecasting Using Neural Networks vs. Box-Jenkins Methodology," *Artificial Neural Networks: Forecasting Time Series*, V. R. Vemuri and R. D. Rogers eds., IEEE Computer Society Press, 1994, pp. 20-27.

[12] P. Werbos, "Neural Networks, System Identification and Control in the Chemical Process Industries," in *Handbook of Intelligent Control. Neural, Fuzzy, and Adaptive Approaches*, D. A. White and D. A. Sofge eds., Van Nostrand Reinhold, 1992, pp. 283-356.

Radu Drossu (rdrossu@eecs.wsu.edu) received his M.S. degree in Electrical Engineering from the Polytechnical Institute of Bucharest - Romania in 1990. He worked as a system programmer and hardware designer at the Research Institute for Automation (IPA), Bucharest from 1990 to 1993. He is currently a Ph.D. candidate in Computer Science at Washington State University, doing his research in Artificial Neural Networks under the supervision of Dr. Zoran Obradovic.

Zoran Obradovic (zoran@eecs.wsu.edu) received the B.S. degree in Applied Mathematics, Information and Computer Sciences in 1985; the M.S. degree in Mathematics and Computer Science in 1987, both from the University of Belgrade; and the Ph.D. degree in Computer Science from the Pennsylvania State University in 1991. He was a systems programmer at the Department for Computer Design at the Vinca Institute, Belgrade, from 1984 to 1986, and has been a research scientist at the Mathematical Institute of the Serbian Academy of Sciences and Arts, Belgrade, since then. At present, he is an Assistant Professor in the School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164-2752, USA. The objective of his current research is to explore applicability of neural networks technology to large scale classification and time series prediction problems in very noisy domains.