PREDICTION OF CHAOTIC TIME SERIES USING RECURRENT NEURAL NETWORKS

Jyh-Ming Kuo, Jose C. Principe, and Bert de Vries*

Department of Electrical Engineering
University of Florida
Gainesville, FL 32611

*Sarnoff Research Center
Princeton, N.J. 0857

Abstract - In this paper, we propose to train and use a recurrent artificial neural network (ANN) to predict a chaotic time series. Instead of training the network with the next sample in the time series as is normally done, a sequence of samples that follows the present sample will be utilized. Dynamical parameters extracted from the time series provide the information to set the length of these training sequences. The proposed method has been applied to predict both periodic and chaotic time series, and is superior over the conventional ANN approach.

INTRODUCTION

Chaotic time series are the output of a deterministic system with positive Lyapunov exponents. Therefore, unless one can specify the initial condition with infinite precision, the long term future behavior of these time series becomes unpredictable. However, the short term behavior may be captured with accuracy. This characteristic makes the prediction of chaotic time series very challenging.

In order to build a predictive model for a chaotic time series, the most common approach is to adjust the parameters of a pre-selected model to minimize the mean squared error in one-step prediction [1],[2],[3],[4]. Since this predictive model is developed with the interest of predicting more than one sample ahead, the resultant predictor must be utilized recursively to produce new samples from predicted samples, i.e. multi-step prediction. It is worth mentioning that the way the resultant predictor is used differs from the way it is trained, i.e. the model system is trained as a feedforward network (Figure 1a), but it is used as a recurrent system for multi-step prediction (Figure 1b). The feedforward model is trained one sample at a time in a static framework, without information regarding the time evolution of the signal. Therefore, it is no surprise that the performance in multi-step prediction which involves dynamics (iteration of an autonomous system) may not be very robust [9].

0-7803-0557-4/92$03.00 © 1992

In this paper we propose to change the training paradigm for the predictor from static to dynamic, i.e. we will be using trajectory learning. In trajectory learning one needs an initial condition and a sequence of desired points, which for our case reduces to the time series samples. In implementing this learning based on gradient-descent procedure, one needs either to propogate back errors (backpropagation through time) or to update the recursive formulae for the computation of derivatives (real time recurrent learning). We propose a recurrent configuration to bring forward the results of the past predictions and mimic the way the model system performs multi-step prediction. The issue of finding the appropriate segment of the trajectory will be addressed. Comparisons of the conventional topology and learning schemes with the proposed ones will be given for a periodic ($\sin^{30}(wt)$)and a chaotic time series (Mackey-Glass with D=30).

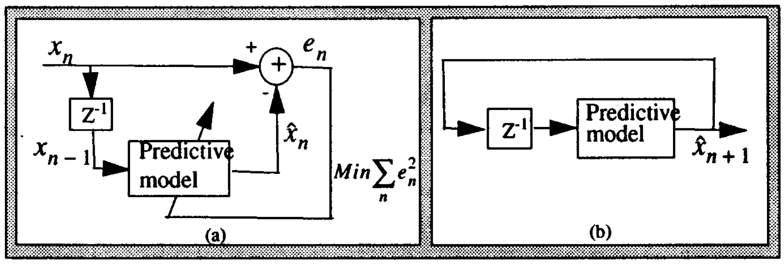

Figure 1. Schematic diagrams for (a) the training mode, and (b) the operating mode in a conventional approach.

NEURAL NETWORK ARCHITECTURE AND TRAINING

A smooth map is guaranteed to exist in state space [4], when Takens' embedding (delay) method [5] is used to reconstructed a trajectory from a time series produced by a nonlinear autonomous dynamical system. Prediction can be equated to the identification of this smooth map, and should be framed as a dynamical operation. Previous researchers have utilized a feedforward system (Figure 1a) to approximate this map. Basically, the input is delayed by one sample and the system is trained to predict the next sample, with weight updates after each sample. Moreover, the time series are always fed to the input of the neural network [1],[2],[3], which amounts to a static learning paradigm (no information from the previous prediction is used). In multi-step prediction, after the map has been established, the previously predicted sample is utilized as the input to the network to generate the next prediction. This iterative use of the predicted values is not accounted for during the training of the network.

According to dynamical theory the predictive model when iterated as in Figure 1b should be able to produce an output that displayed the same dynamical invariants as the original system. Unfortunately, our experience shows that this is not always the case [9]. In order to improve the accuracy of the predictive model we propose here two modifications to the accepted scheme: First, we propose to use a dynamic learning paradigm such as trajectory learning to adapt the predictor. We chose to compute the error at each step but update the coefficients only after the selected number of multi-step predictions. The gradient information is computed at each step following the method of real time recurrent learning [6], but the network weights are adapted only at the end of a training segment. The error criterion utilized for learning is

$$E = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=0}^{l} e_n^2(k)$$

where N is the number of the training patterns, l is the length of each training sequence, and $e_n(k)$ is the recursive prediction error at kth iterates for nth training sequence.

The second aspect deals with the topology of the network, which was chosen to be recurrent, basically, a time-delay neural network (TDNN) with a feedback loop (Figure 2). This architecture implements naturally trajectory learning. This means that the input of the ANN is initially the original time series, but it is progressively replaced by the predicted samples.

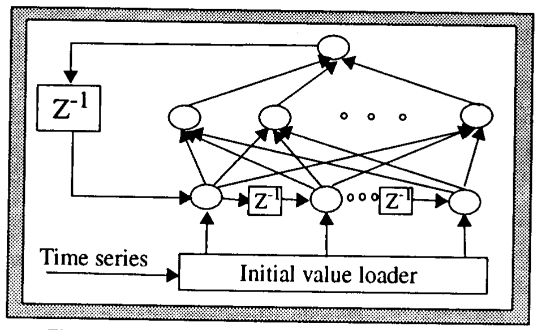

Figure 2. Proposed recurrent network architecture

When trajectory learning is selected as the training paradigm for chaotic time series prediction, the important aspect is to define what is the best segment length due to the existence of at least a positive Lyapunov exponent. A positive Lyapunov exponent means that even small modeling errors will be magnified for large segments, which create unnecessary corrections in weight values. The other problem is that if too large a segment is used, then conflicting requirements from trajectory to trajectory will be imposed on the network (in the attractor no two trajectories are the same so the coefficients will learn an average behavior). These two points will be explained in the following section.

DECIDING THE LENGTH OF TRAINING DATA SEGMENTS

Since a chaotic time series possesses positive Lyapunov exponents, the prediction error will be amplified and accumulated at each iterative prediction. In trajectory learning, the predicted samples, which are subject to modeling error, are fed back as inputs to the ANN. From the point of view of reconstructed dynamics, the desired signal and the predicted sequence represent two different trajectories in the same attractor. The problem is then one of providing the wrong desired signal to the network, because the desired signal is in one trajectory (the original time series) and the network input belongs to another trajectory. The positive Lyapunov exponents is the measure of the divergence of nearby trajectories, therefore the length of the training segment should be determined from the dynamical properties of the time series.

Another problem is the conflict produced by the similarity at small time scales, but divergence at larger time scales. If a small portion of a chaotic signal is selected one can find through the segment several portions that fit it pretty well. However, if the portion of the segment is too large, then nowhere we can match it. This creates problem in extrapolating because if the trajectory for learning is selected too long then conflicting requirements will exist. Suppose that during training the network produced two prediction sequences starting from two slightly different initial conditions. When we convert the sequences into state-space trajectories, the trajectories will diverge from the corresponding segments of the original signal's trajectory. Let us assume after a number of training epochs the model error becomes small, and the divergence of these reconstructed trajectories is mainly caused by the "error-amplifying" effect of the positive Lyapunov exponents. Casdagli suggested that the prediction error will grow under iteration at a rate equal to the largest Lyapunov exponent [4]. If these prediction sequences are long enough, their trajectories will fall into overlapping "uncertain" regions around the corresponding segments of the original signal's trajectory. An example is given in Figure 3 to explain this situation. When this happens, confusion may occur in the training. Consequently, an instability may be observed at the end of training.

To avoid this convergence problem, we propose an upper bound (l_{max}) for the length selection of training data segments,based on Casdagli's conjectured scaling law, given by the following inequality

$$(dist_{min})/2 > \sigma e^{\lambda_{max} l_{max} \tau}$$

where $dist_{min}$ is the minimal distance between two points on the reconstructed trajectory, σ is the mean-square error for one-step prediction, λ_{max} is the largest positive Lyapunov exponent, τ is the sampling period. There are a couple of algorithms available to compute the largest Lyapunov exponent for experimental data (e.g. Wolf's algorithm [7]). σ can be estimated from a one-step-predictor training, which takes far less computation time.

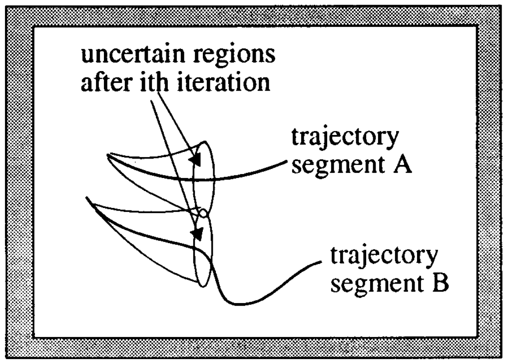

Figure 3. An example of overlaped uncertain regions after ith iteration

RESULTS

The proposed method has been used to predict a periodic, but highly nonlinear time series (generated by sampling $\sin^{30}\omega t$) and a chaotic time series(generated by Mackey-Glass equation with delay equal to 30 [8]). In the experiment with the periodic signal, the proposed recurrent ANN with 9-18-1 architecture (9 input units, 18 hidden units, and 1 output unit) was trained to produce the next 40 samples for a given initial condition. There are 20 training patterns which were prepared every 5 data samples. To compare the performance of the proposed and the conventional methods, a feedforward TDNN with the same size was trained as a one-step predictor using backpropagation algorithm. We stopped the training of both networks when their one-step prediction learning curves reached the same performance.

After the training, 2000 iterations were computed using both networks for the same initial condition. In Figure 4, the results show that the waveform can be preserved in all of these iterations using our proposed method, while it is severely distorted in the later iterates using the TDNN. This implies that the resultant mapping obtained by trajectory learning forms the same attractor as the underlying dynamics of the original signal.

In the experiment with the chaotic time series, an 8-14-1 recurrent ANN was trained to generate 8-sample-long sequences.The computed upper bound is 13 samples. However, we found that the performance was not good using this upper bound. We conjecture that this may be because the measurement of the largest Lyapunov exponent is an average process, and there exists a deviation for this measure. For some training data segments, the confusion in the training, as explained in Figure 3, may occur. For comparison, we also trained a same sized TDNN to be a one-step predictor. The learning curves of one-step prediction error for both ANNs are given in Figure 5. We notice that the error of the TDNN is smaller than that of the recurrent ANN when the training was stopped after 1000 epochs. Since the signal is a chaotic time series, the performance of the resultant models can not be compared by simply observing the waveform [9]. In the literature [1][4], the mean square error curve in multistep prediction has been used as a performance index. In Figure 6, we show the curves of multistep prediction error for both ANNs. The Casdagli's conjectured scaling law due to the largest Lyapunov exponent is also given. Based on the result, the proposed ANN works much better than the TDNN, and its performance is closer to the conjectured curve. To test the robustness of our method, three different sets of initial random weights were used to train the recurrent ANN. The error curves, given in Figure 6, show that the performances of the resulting models with different initial weights are consistent.

CONCLUSIONS

In this paper, we propose that a predictor should be trained to predict a segment of signal instead of just the next sample when multi-step predictions are of interest. This is equivalent to imposing constraints on the iterative map of the predictor during the training. The number of constraints (or the length of training data segments) can be estimated from the dynamics reconstructed from the signal. According to this training scheme, a neural network with the least number of recurrent loop was implemented, and a recursive algorithm was used in the weight adjustments. The experimental results show that the performance of the long-term prediction can be improved significantly when a model was trained with our proposed method.

Although an upper bound for the length of the training segments can be computed using our proposed formula, we found in our experiments that this estimate is usually too large. We conjecture that this is because the measure of the largest

Lyapunov exponent is a statistic. To estimate this upper bound more precisely, a "guard distance" between two uncertain regions in figure 3 has to be taken into account. This distance can be decided from the statistics in the measurement of the largest Lyapunov exponent.

In our experiment with the chaotic time series, we conclude that a smaller one-step prediction error does not guarantee a better performance in multi-step prediction. This finding is consistent with the results of our previous work of comparing the performances of linear and nonlinear predictors[9].

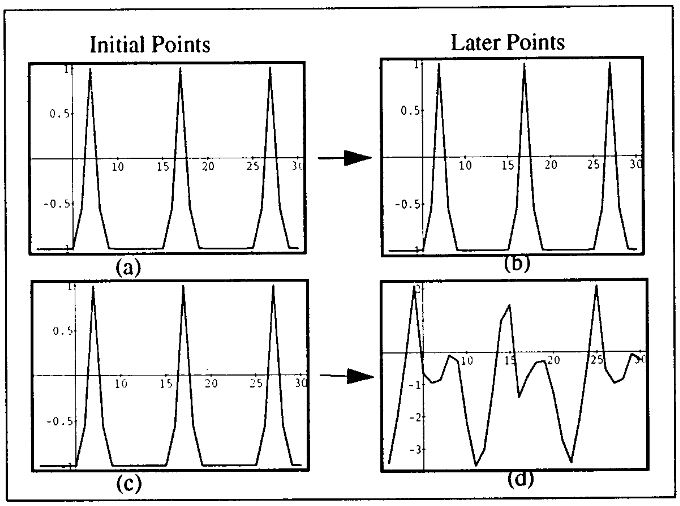

Figure 3. (a)first 30 iterates and (b)last 30 iterates in multistep prediction using the recurrent ANN, and (c) &(d) are results using the TDNN.

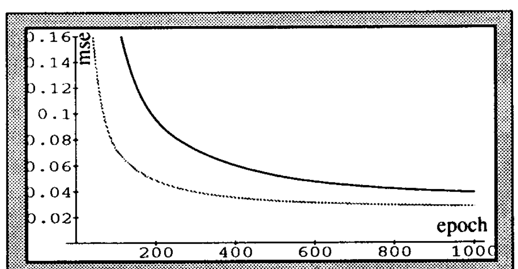

Figure 4. Learning curves of one-step prediction error
dark line: recurrent ANN, gray line: TDNN

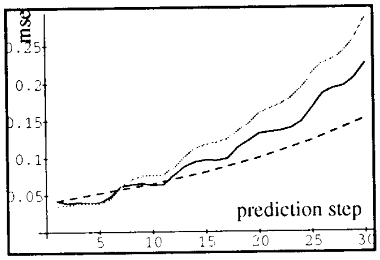

Figure 5. Performance comparison
dark line: recurrent ANN
gray line: TDNN
dashed line : conjectured curve

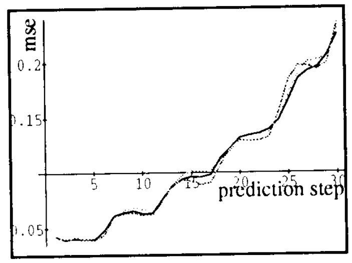

Figure 6. error curves for different initial weights

References

[1] A. Lapedes and R. Farber, "Nonlinear signal processing using neural networks: prediction and system modelling", technical report LA-UR-87-2662, Los Alamos National Laboratory, Los Alamos, NM. 1987

[2] W.C. Mead, et al. "Prediction of chaotic time series using CNLS-NET-- Example: the Mackey-Glass equation", technical report LA-UR-91-720, Los Alamos National Laboratory, Los Alamos, NM. 1991.

[3] A.S. Weigend, B. A. Huberman, and D. E. Rumelhart, "Predicting the future: a connectionist approach", Int'l Journal of Neural Systems, 1:193, 1990.

[4] M. Casdagli, "Nonlinear prediction of chaotic time series", Physica 35D, 1989, pp. 335-356.

[5] F. Takens, "Detecting strange attractors", In: Dynamical systems and turbulence, edited by D.A. Rand and L.S. Yang, Lecture Notes in Mathematics, vol. 898, Berlin: Springer, 1981.

[6] R.J. Williams, and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks", Neural Computation, 1, 1989, pp.270-280.

[7] A. Wolf, et al., "Determining Lyapunov exponents from a time series", Physica 16D, 1985, pp. 285-317.

[8] M.C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems", Science ,197, 1977.

[9] J.C. Principe, A. Rathie, and J.M. Kuo, "Prediction of chaotic time series with neural networks and the issue of dynamic modeling", accepted in Int'l Journal of Bifurcation and Chaos, 1992.