International Symposium on Nonlinear Theory and its Applications
Xi'an, PRC, October 7–11, 2002

Structural Learning of Neural Network for Chaotic Time Series Prediction

Basabi Chakraborty

Faculty of Software and Information Science
Iwate Prefectural University
152-52 Aza Sugo, Takizawamura, Iwate 020-0193, Japan
Phone: +81-19-194-2584, Fax: +81-19-694-2573
email: basabi@soft.iwate-pu.ac.jp

Abstract

Among several nonlinear models for prediction of chaotic time series, artificial neural network(ANN) models have gained major attention in recent days. The widely recognized problem with the use of neural network is to design an efficient structure of the network in order to achieve sufficient prediction performance. Various approaches to reduce or to grow the network to optimum size from an initially larger or smaller network respectively via special learning algorithm can be found in literature as possible solutions. In this work, considering the close tie of chaotic time series with fractal structure, a neural network with fractal connection structure has been proposed as the initial network instead of a full connection network to reduce the training time. It is then trained by a variation of structural learning algorithm with forgetting to optimize the connections. The simulation has been done with a simple chaotic time series generated by a logistic map.

1. Introduction

Prediction of future values of a chaotic time series is quite a challanging task. There are several well-known algorithms available for the analysis and prediction of chaotic time series with the tools from classical physics [1], [2] and statistics [3]. Recently connectionist models are found to achieve better results than conventional methods. Several researchers have developed varieties of artificial neural network models for processing and prediction of chaotic time series. Lapdes-Ferber network [4] or Farmar-Sidorowitch network [5] used backpropagation algorithm whereas Moody-Darken [6], Broomhead and Lowe [7], and many others [8] used radial basis function networks for chaotic time series prediction. Wan [9] used time-delay perceptron model for the prediction task. Connectionist models seem to have better performance in prediction accuracy than traditional mathematical models.

However the design of an efficient architecture and choice of the parameters of a neural network predictor, suitable for a particular problem, is quite tricky. An approach to handle this problem is to use genetic algorithms (GA) or evolutionary computation for finding out the most suitable architecture from a large pool of network architectures. But these methods require extensive computation time and proper encoding of the problem at various stages. Another technique is network pruning. The objective of pruning is to find the smallest possible network that fits the data well and produces good forecasts. The main idea is to start from a larger network and to remove unnecessary connections and nodes through learning. There are varieties of techniques for getting pruned network, a good survey is in [10].

In the present work, considering the close tie of chaotic time series with fractal structure, a neural network with fractal connection structure proposed earlier by author [11] has been used as the initial network instead of a full connection network to reduce the training time. It is then trained with a structural learning algorithm to fix the proper structure and parameters of the neural network. The trained network is used for the prediction of future values of the time series. The simulation has been done with logistic map which generates a simple chaotic time series.

2. Neural Network Model

2.1. Fractal Neural Network

The proposed fractal neural network(FNN) model is a modified feed forward multilayer network in which upper layer neurons are connected to the lower layer neurons with a probability following an inverse power law which generates a sparse network with statistically self similar fractal connection structure instead of full connection. However the final hidden layer is fully connected to the output layer. Each layer is an array of neurons in one or two dimension depending on the type of input to be processed. The probability that ith processing element in the kth layer receives connection from the jth processing element of the previous layer, defined by CP_{ijk} follows the law

$$\begin{aligned} CP_{ijk} &= Ar_{ijk}^{D_k - d} \\ i &= 1, 2 \dots n_k \\ j &= 1, 2 \dots n_{k-1} \\ & 0 \leq D_k \leq d \end{aligned} \tag{1}$$

where r_{ijk} is the Euclidean distance between ith processing element in the kth layer (considering one dimensional layers) and jth processing element of the previous layer defined as

$$r_{ijk} = ||Q_{ik} - Q_{j(k-1)}||, \; r_{ijk} \geq 1 \tag{2}$$

d denotes dimension of the array of neurons in kth layer. A represents a constant, D_k represents the fractal dimension of the connection distribution of k thx layer. Q_{ik}, and $Q_{j(k-1)}$ denotes the spatial position of the ith processing element in the kth layer and jth processing element of the previous layer defined by

$$Q_{ik} = [\lceil n_{k-1}(2i - 1)/2n_k \rceil, k] \text{ for } i = 1, 2, \dots, n_k \tag{3}$$

where n_{k-1} and n_k represents the number of neurons in the $(k-1)$th and kth layers respectively.

To implement such a sparse NN, for each i, j, k, a uniform random number ρ on the interval [0,1] has to be generated and the connectivity C_{ijk} of the link from the ith processing element in the kth layer to the jth processing element of the previous layer is to be assigned as

$$\begin{aligned} C_{ijk} &= 1, \text{ if } CP_{ijk} \geq \rho \\ &= 0, \text{ Otherwise} \end{aligned} \tag{4}$$

2.2. Proposed Structural Learning

Structural learning with forgetting [12], a modified backpropagation algorithm with forgetting unwanted connections is a special type of pruning technique in which the final trained network retains only the essentials connections. The optimized network obtained by structural learning shows better generalization for a particular problem than an arbitrary feedforward network with standard back propagation.

The basic idea according to [12] to update the connection weight as follows.

$$\Delta w_{ij} = \Delta w'_{ij} - \gamma sgn(w_{ij}) \tag{5}$$

$\Delta w'_{ij}$ is the change of ijth weight according to standard back propagation , γ is the forgetting rate and $sgn(x)$ denotes signum function. The corresponding criterion function is given by,

$$J = \sum_i (y_i - y_i^*)^2 + \gamma' \sum_{i,j} |w_{i,j}| \tag{6}$$

Here y_i and y_i^* are the actual and target values of network outputs, respectively. $\gamma' = \lambda\gamma$, where λ is the learning rate. The first term of Eq 6 represents the standard sum of squared errors (SSE) and the second term represents the sum of absolute value of weights. At the beginning of the training SSE drops quickly while the sum of the weights changes only slightly as learning advances SSE changes less intensively. The delayed convergence is a disadvantage of structural learning but it can be compensated by the reduction in the final size of the network. The proper choice of γ is crucial for better generalization and optimum structure. Various strategies exist for choice of γ. In this work a variation of the above algorithm has been proposed and is described as follows.

1. The initial network is set up with fractal connection structure, fractal dimension being matched to the fractal dimension of the time series. The connection weights of some of the links are zero representing no connection and others are initialized to random values within a predefined limit.
2. The network is then trained for all the training patterns with Eq 6 as the cost function incorporating following rule for weight change.
   - For the nonzero initial connections change the weight as usual.
   - For the connections with initial zero connection weight if $\Delta w_{ij} \geq \delta$ then change the weight. δ is a predefined limit to be set after some experimentation.
3. Examine the network and make the connections zero if it is smaller than δ.
4. Go to step 2 for the next cycle of training.
5. Repeat step 2 to 4 until one of the following stopping criteria is met.
   - A predefined limit of maximum number of cycles is reached.
   - The average sum of squared error reaches a minimum and is not changing appreciably.

The above algorithm decreases the total weight of the network and in this process unwanted connections fade away. Starting with an intial sparse network reduces the training time considerably than structural learning algorithm with initial full connection network. The initial sparse network also helps in keeping the total weight of the network to a comparatively lower value, making the mean square error comparable to standard BP algorithm.

3. Chaotic Time series Prediction

3.1. Data Generation

In this paper the experiment has been done with the chaotic time series generated by the logistic map characterized by a control parameter C whose variation generates different time series of different correlation value of the consecutive terms in the series. The chaotic time series generated by the logistic map the dynamics of which is governed by the following difference equation:

$$X_n = CX_{n-1}(1 - X_{n-1}) \tag{7}$$

where C denotes the control parameter and $0 < C \leq 4$ to make the time series data lie in the interval $[0, 1]$. The initial value of the series x_0 has been taken as 0.1 for all the generated series. The values of C used in our experiment correspond to 3.56, 3.68 and 4.00 to have the series in the chaotic range.

3.2. Neural Network Model Selection

A multilayer (one dimensional layers) feed forward network with one hidden layer and fractal connection structure described earlier is used for our experiment. The number of neurons in input layer, hidden layer and output layer is 12, 20 and 1 respectively. The connection structure is set up according to Eq 1 and Eq 4 with value of constant A as 1 with proper choice of value for fractal dimension. For comparison a full connected multilayer perceptron with single hidden layer is also used with number of neurons in the input, hidden and output layers as 12, 20 and 1 respectively.

3.3. Training and Test sample generation

The training patterns are generated by segmenting the time series data by a window of width 12 to have 12 consecutive values of the time series as input and 13th value as the output to the network. The successive samples are generated by sliding the window over the series. A total of 50 samples have been generated for training. For testing purpose another portion of the time series, well past the portion which is used for training, is used and 20 test samples are generated in the same way as training samples.

Number of cycles	Prediction Error for		
	Standard BP learning	Structural learning	
		with MLP	with FNN
10000	0.06147	0.04645	0.05932
30000	0.05329	0.01423	0.01401
50000	0.05198	0.01275	0.01217

Table 1: Prediction Error for different networks for the time series for C = 3.68

4. Simulation and Results

The neural network model is trained by proposed structural learning algorithm for a training cycle of 10,000 to 50,000 with the training samples generated from logistic map. The learning rate and the forgetting rate are taken as 0.1 and 0.0001. The value of δ is taken as 0.001 . The fractal dimension of the fractal connection structure is varied from 0.8 to 0.95 and different network has been tried. The fractal dimension of value 0.9 came out to be the best. Table 1 represents the prediction error of the time series generated by logistic map with control parameter $C = 3.68$. The table shows the comparison of the full connection multilayer feedforward model(MLP) with standard backpropagation, standard structural learning with forgetting with full connection network and structural learning with forgetting with initial fractal neural network model(FNN) (fractal dimension 0.9) in terms of prediction performance. Prediction error (PE) is defined as the average root mean square error over the test samples as follows.

$$PE = \frac{1}{n}\sqrt{\sum_{t=1}^{n}(a_t - y_t)^2} \tag{8}$$

where y_t and a_t represents the predicted and actual value of the time series for tth test sample and n represents the number of test samples.

Table 2 represents the comparison of prediction values with the same networks for the time series with $C = 4.00$. Both the tables support that the prediction accuracy for the proposed algorithm is better than the other two algorithms. For the time series for $C = 4.00$ where the correlation between the successive samples are less than the time series of $C = 3.68$, the prediction error has larger values for all the algorithms. Now the training time and the prediction time for the proposed algorithm has found to be considerably lower than that of structural algo-

Number of cycles	Prediction Error for		
	Standard BP learning	Structural learning	
		with MLP	with FNN
10000	0.07254	0.07561	.07964
30000	0.06378	0.05782	.03841
50000	0.05432	0.04156	.02147

Table 2: Prediction Error for different networks for the time series for C = 4.00

Actual	0 .985	0 .057	0.214	0.674	0.879
Predicted	0 .948	0.059	0.183	0.725	0.976

Table 3: The best prediction output obtained with structural learning with FNN

rithm with initial full connection network. The number of cycles needed to have a stable prediction is larger for structural learning algorithms compared to standard BP algorithm but prediction accuracy is far better. Table 3 shows the best prediction output for the proposed algorithm for time series of $C = 4.00$.

5. Conclusion

In this work, chaotic time series prediction by multilayer feedforward neural network model with different learning algorithm has been studied. Structural learning algorithm with forgetting is useful for optimization of the structure of the network. Optimized network structure is essential for good prediction accuracy and quicker prediction. By standard backpropagation learning one has to do trial and error procedure for finding out the most suitable architecture. But structural learning, though good for finding out the optimal structure automatically through learning, takes longer time for training if started from full connection network. The proposed fractal neural network used as the initial network reduces the training time without deteriorating the final prediction accuracy. Though generalization behaviour of the final network has not studied in this work but intuitively it should not be worse as final network in both the structural algorithms is almost the same. The effeciency of the procedure should be tested with other complicated chaotic time series as well as real world chaotic data. The simulation with chaotic time series generated by simple logistic map shows encouraging result.

References

[1] M. Sano and Y. Sawada, 'Measurement of the Lyapunov spectrum from chaotic time series', Phys. Rev. Lett. Vol 55, pp. 1082, 1985.

[2] A. Wolf et al., 'Determining Lyapunov exponents from a time series', Physica D, Vol 16, pp. 285, 1985.

[3] G. E. P. Box, G. M. Jenkins and G. C. Reinsel, 'Time Series Analysis: Forecasting and Control', Prentice-Hall, 1994.

[4] A. Lapdes and R. Farber, 'Nonlinear signal processing using neural networks: Prediction and system modelling', Tech. Rep. LA-VR-87-2662 Los Alamos National Laboratory, Los Alamos, NM 1987.

[5] J. D. Farmar and J. Sidorowitch, 'Predicting chaotic time series', Phys. Rev. Lett. Vol 59, pp. 845-848, 1987.

[6] J. E. Moody and C. J. Darken, 'Fast learning in networks of locally tuned processing units', Neural Computation Vol 1, pp. 281–294, 1989.

[7] D. S. Broomhead and D. Lowe, 'Multivariable functional interpolation and adaptive networks', Complex Systems, Vol 2, pp.321–325, 198 9.

[8] G. Chakraborty, B. Chakraborty, M. Murakami, and N.Shiratori, 'A Study on Chaotic Time Series Prediction Using RBF-Connectionist Network', International Conference cum Workshop on Current trends in Data Acquisition and Control, Calcutta, India, pp.110–115, December 1994.

[9] E. A. Wan, 'Time series prediction by using a connectionist network with internal delay lines', in Time series Prediction : Forecasting the Future and Understanding the Past', pp. 195–217, Addison-Wesley, 1994.

[10] R. Reed, 'Pruning Algorithms – A Survey', IEEE Trans. on NN, Vol . 4, No. 5, pp 740–747, September 1993.

[11] Basabi Chakraborty and Goutam Chakraborty, ' Chaotic Time Series Prediction by Fractal Neural Network' in Proceedings of KBCS-98 (International Conference on knowledge Based Computing Systems), PP.323–332, December 1998.

[12] M. Ishikawa,'Structural Learning with forgetting', Neural Networks, Vol. 9, PP. 509–521, 1996.