On-line Identification and Prediction of Lorenz's Chaotic System Using Chebyshev Neural Networks

A. Ashraf-Modarres
Tarbiat Modarres University
a_modarres@modares.ac.ir

V. Johari-Majd
Tarbiat Modarres University
majd@modares.ac.ir

Abstract: In this paper, a single layer functional link ANN based on Chebyshev polynomials is used for identification and prediction of chaotic systems. This model is linear in their parameters and nonlinear in the inputs. Therefore, on-line system identification is achievable by the use of recursive least squares method with forgetting factor. The remarkable prediction performance was gained using proposed Chebyshev neural network (CNN) model and recursive nonlinear predictor method on noise free chaotic Lorenz series.

Keywords: Identification, Prediction, Chaotic Systems, Chebyshev Neural Networks.

1 Introduction

A large set of methods has been developed recently which apply artificial neural networks (ANN) to the tasks of identification and prediction of nonlinear dynamic systems. Different neural network structures for nonlinear dynamic system modeling and identification is presented in [1]. The identification and prediction of nonlinear chaotic systems, which show "sensitive dependence on initial conditions" behavior, occupy an important place among of researcher interests.

At present, most of the works on identification and prediction of nonlinear chaotic system using neural networks are based on multilayer perceptron (MLP) with backpropogation learning [2-4].

Identification and prediction of chaotic system using radial basis functions neural networks (RBFNN) are dealt in [5-7]. A comparative study on the effects of different basis functions used in the RBF neural networks for time series prediction purpose has been reported in [8].

Recently, there has been a considerable interest in the functional link artificial neural network (FLANN) [9-12]. In this paper we use a type of FLANN based on Chebyshev polynomials to show that functional network architectures provide simpler and more efficient techniques to predict nonlinear time series. Because of single layer structure of CNN, its computational complexity is intensively less than MLP and can be used easily for on-line identification purposes [9, 12].

This paper is structured as follows. In Section 2 we give a brief introduction to Chebyshev neural networks. In this section, also we describe how CNN can be used for identification and prediction applications. Section 3 introduces Lorenz's system, while section 4 presents the simulation results of the multi-step ahead prediction on mentioned chaotic system using CNN. Section 5 gives some conclusions.

2 Chebyshev neural network (CNN)

2.1 Structure of CNN

The CNN architecture used in this paper has a single layer structure, which introduced in [9]. CNN is a functional link network (FLN) based on Chebyshev polynomials. Among orthogonal polynomials, the Chebyshev polynomials occupy an important place, since, in the case of a broad class of functions, expansions in Chebyshev
polynomials converge more rapidly than expansions in other set of polynomials.

The Chebyshev polynomials can be generated by the following recursive formula [11]:
\[ T_{i+1}(x) = xT_i(x) - T_{i-1}(x), \quad T_0(x) = 1 \]  
(1)

For example, consider a two dimensional input pattern \( X = [x_1, x_2] \). An enhanced pattern obtained by using Chebyshev functions is given by:
\[ \phi = [1 \ T_1(x_1) \ T_2(x_1) \cdots \ T_1(x_2) \ T_2(x_2) \cdots] \]  
(2)

where \( T_i(x_j) \) is a Chebyshev polynomial, \( i \) the order of polynomials chosen and \( j = 1, 2 \). The different choices of \( T_i(x) \) are \( x, 2x, 2x-1 \) and \( 2x+1 \). In this paper, \( T_1(x) \) is chosen as \( x \).

The architecture of the CNN, as shown in Fig. 1, consists of two parts, namely numerical transformation part and learning part. Numerical transformation deals with the input to the hidden layer by approximate transformable method. The transformation is the functional expansion of the input pattern comprising of a finite set of Chebyshev polynomials. As a result the Chebyshev polynomial basis can be viewed as a new input vector. The learning part is a functional link neural network based on Chebyshev polynomials.

The output of the single layer neural network is given by:
\[ \hat{y} = \hat{W}^T \phi \]  
(3)

where \( \hat{W} \) are the weights of the neural network given by \( \hat{W} = [w_1 \ w_2 \cdots] \).

A general nonlinear function \( f(x) \) can be approximated by CNN as:
\[ f(x) = \hat{W}^T \phi + \varepsilon \]  
(4)

where \( \varepsilon \) is the CNN functional reconstruction error vector.

2.2 Identification by CNN

The identification problem in the time-domain for either linear or nonlinear modeling is to infer relationships between past input-output data and future outputs.

If a finite number of past inputs \( u(n) \) and outputs \( x(n) \) are collected into the vector \( \phi(n) \),
\[ \phi(n) = [x(n), x(n-1), \cdots, x(n-n_x)], \]
\[ u(n), u(n-1), \cdots, u(n-n_u)]^T \]  
(5)

then the problem is to understand the relationship \( f(.) \) between the next output \( x(n+1) \) and \( \phi(n) \). The function \( f(.) \) can be any function and it is indeed the function, which defines the model structure. As already discussed in section 2.1, the problem of identification by CNN is setting up a suitably parameterized model as Eq. (4), and adjusting the parameters \( \hat{W} \) to optimize a performance function based on the error between the plant and identification model outputs.

As can be seen from Fig.1, CNN is linear in the weights and nonlinear in the inputs. Therefore, we can use the recursive least squares method with forgetting factor as the learning algorithm for the purpose of on-line weight updation. The performance function to be minimized is given by:
\[ E = \sum_{i=1}^{k} \lambda^{i-1} |e(i)|^2 \]  
(6)

The algorithm for the discrete time model is given by:
\[ \hat{W}(n) = \hat{W}(n-1) + k(n)e(n) \]
\[ k(n) = \frac{\lambda^{-1} P(n-1) \phi(n)}{1 + \lambda^{-1} \phi^T(n) P(n-1) \phi(n)} \]  
(7)
\[ e(n+1) = x(n+1) - \hat{x}(n+1) \]
\[ P(n) = \lambda^{-1} P(n-1) - \lambda^{-1} k(n) \phi(n)^T P(n-1) \]

where \( \lambda \) is the forgetting factor and \( \phi \) is the basis function formed by the functional expansion of the input and \( P(0) = cI \), \( c \) is a positive constant, \( \| P(t) \| < R_0 \), \( R_0 \) is a constant that serves an upper bound for \( \| P(t) \| \). All matrix and vectors are of compatible dimension for the purpose of computation. When a satisfactory model is found, prediction may readily be computed.
2.3 Prediction of chaotic system by CNN

Consider a vector $X = [x(n), x(n-1), x(n-2), \ldots, x(n-k)]$ as input to CNN. Thus, a functional expansion $\phi(n)$ could be formed as following:

$$\phi(n) = [1, T_1(x(n)), T_2(x(n)), \ldots, T_k(x(n-k)), T_2(x(n-k)), \ldots]$$  \hspace{1cm} (8)

An estimate $\hat{x}(n+1)$ of the next data sample $x(n+1)$ is obtained by constructing a nonlinear model:

$$\hat{x}(n+1) = f(\phi(n)) = \hat{W}^T \phi(n)$$  \hspace{1cm} (9)

Eq. (9) is for 1-step ahead prediction, but this could be generalized for $N$-step ahead prediction, i.e.

$$\hat{x}(n+N) = f_N(\phi(n))$$  \hspace{1cm} (10)

where $f_N(\phi(n))$ would, in general, represent a different predictor function for each value of $N$.

Haykin and Li [13] presented using recursive prediction to test the generalization properties of their nonlinear predictors. Recursive prediction is performed by first of all training a predictor to obtain the mapping in Eq. (9). Then the trained predictor is given one input vector from the available data. From then on, the output of the predictor is fed back to its input, and the system becomes autonomous.

Haykin and Principe [14] suggested using recursive prediction as a pragmatic approach for testing how well a 1-step ahead predictor had managed to model the underlying dynamics of a chaotic signal. If the predictor is successful at modeling the underlying dynamics of the chaotic signal, then the predictor’s output, in recursive prediction mode, should satisfy the two conditions listed below.

(1) Short-term behavior: Once initialization is completed, the reconstructed time series $\{\hat{x}(n)\}$ should closely follow the original time series $\{x(n)\}$, for a period of time approximately equal to the prediction horizon of the signal.

(2) Long-term behavior: The dynamic invariants (such as the correlation dimension), computed from the reconstructed time series $\{\hat{x}(n)\}$ should closely match the corresponding ones from the original time series $\{x(n)\}$. This is because these dynamic invariants measure the global properties of a chaotic signal.

The performance criterion evaluated for the dynamic system to be identified is the multiple correlation coefficient, $R^2$ given by:

$$R^2 = 1 - \frac{\sum_{i=1}^{p} (x_i(n) - \hat{x}_i(n))^2}{\sum_{i=1}^{p} (x_i(n) - \bar{x})^2}$$  \hspace{1cm} (11)

where $p$ is the number of measured samples of the output of dynamic systems [5]. A value of $R^2$ equal to 1.0 indicates an exact fit of the model to the measured data. An $R^2$ value of 0.9–1.0 is considered sufficient for practical applications [15].

3 Lorenz’s chaotic system

In this section, a typical chaotic system – Lorenz’s model – is chosen to demonstrate the CNN’s prediction abilities.

Edward Lorenz was a meteorologist who, in his paper on deterministic non-periodic flow [16], presented an important study on the behavior of a gaseous system. Lorenz equation is shown as the following equations:

$$\dot{x} = -ax + ay$$
$$\dot{y} = -xz + rx - y$$
$$\dot{z} = xy + bz$$  \hspace{1cm} (12)

The constants $a$, $r$, and $b$ determine the system’s behavior. These three equations exhibit chaotic behavior and are very sensitive to initial conditions. The values used here in Lorenz systems are $a = 10$, $r = 28$, and $b = 8/3$. The initial conditions in present work are chosen as $[x_0, y_0, z_0]^T = [7.5, -9.2, 7]^T$, and the sampling time is 0.005 sec. To obtain the time series value at each integer point, the fourth order Runge-Kutta method is used to get the numerical solutions of Eq. (12).

The chaotic behavior of this system is represented in Fig. 2. In this work, the behavior of $x(t)$ is predicted by the CNN.
4 Simulation results

This section presents simulation results for Lorenz’s system prediction using a CNN method with the vector 
\[ X = [x(n), x(n-1), x(n-2), x(n-3)] \] as the input of numerical transformation part, and a ninth order functional expansions corresponding to each element of \( X \). The recursive least squares method with forgetting factor \( \lambda = 0.95 \) has been used to estimate the parameters of model.

In Table 1, the prediction results obtained by CNN are compared with RBFNN method reported in [5].

<table>
<thead>
<tr>
<th>Horizon of prediction</th>
<th>CNN</th>
<th>RBFNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.9998</td>
<td>0.9985</td>
</tr>
<tr>
<td>20</td>
<td>0.9943</td>
<td>0.9954</td>
</tr>
<tr>
<td>30</td>
<td>0.9906</td>
<td>0.9824</td>
</tr>
<tr>
<td>40</td>
<td>0.9826</td>
<td>0.9874</td>
</tr>
<tr>
<td>50</td>
<td>0.9704</td>
<td>0.9937</td>
</tr>
<tr>
<td>60</td>
<td>0.9599</td>
<td>0.9873</td>
</tr>
<tr>
<td>70</td>
<td>0.9413</td>
<td>0.9522</td>
</tr>
<tr>
<td>80</td>
<td>0.9319</td>
<td>0.9331</td>
</tr>
<tr>
<td>90</td>
<td>0.9206</td>
<td>0.9162</td>
</tr>
</tbody>
</table>

Fig. 3-6 demonstrate the true and predicted values of \( x(t) \) for 20, 40, 60 and 90-step ahead prediction.

Fig. 2. Bi-dimensional phase plan of Lorenz system

Table 1. Best \( R^2 \) obtained by CNN and RBFNN for Multi-step ahead prediction of Lorenz system

Fig. 3. True and 20-step ahead predicted values of \( x(t) \) for chaotic Lorenz system

Fig. 4. True and 40-step ahead predicted values of \( x(t) \) for chaotic Lorenz system
As we can find from numerical results and figures, the CNN method provides effective and accurate prediction of chaotic systems.

5 Conclusion

In this paper, it was shown that Chebyshev neural networks (CNNs) can be successfully used to identify and predict the nonlinear chaotic systems. Due to significant properties of CNN including fast convergence and linearity in parameters, the use of recursive least squares method is possible. The advantages of CNN in comparison with widely used MLP and RBFNN methods are to implement easily and obtain more accurate results. The evaluation of method for a chaotic Lorenz system demonstrates good performance of CNN.

References