Time series prediction using a hybrid model of neural network and FIR filter

Khalaf Ashraf A.M., Nakayama Kenji

IEEE International Conference on Neural Networks - Conference Proceedings

Volume 3

Page range: 1975-1980

Year: 1998-05-01

URL: http://hdl.handle.net/2297/6783
Time Series Prediction Using a Hybrid Model of Neural Network and FIR filter

Ashraf A.M.Khalaf
Graduate School of Nat. Sci. and Tech., Kanazawa Univ. Japan
E-mail : nakayama@t.kanazawa-u.ac.jp

Kenji Nakayama

Abstract—Time series prediction is a very important technology in a wide variety of fields. The actual time series contains both linear and nonlinear properties. The amplitude of the time series to be predicted is usually continuous value. For this reason, we combine nonlinear and linear predictors in a cascade form. In order to estimate the minimum size of the proposed predictor, we propose a nonlinearity analysis for the time series of interest. Computer simulations using the sunspot data have demonstrated the efficiency of the proposed predictor and the nonlinearity analysis.

1. Introduction

It is well known that linear filters are insufficient to deal with nonlinear time series processing. On the other hand neural networks are useful for nonlinear adaptive signal processing. They have many important properties such as nonlinearity built into their structures, input-output mapping capability, and adaptivity. So, neural networks have been applied successfully in a variety of signal and information processing fields. One of these fields is the nonlinear time series prediction [1],[2],[3],[4], and others. Neural networks were first applied to time series prediction by Lapedes and Farber (1987) [1].

In practice, many of the time series include both nonlinear and linear properties. Furthermore, the amplitude of the time series is usually continuous. Therefore, it is useful to use a combined structure of linear and nonlinear models to deal with such signals. A combined structure was proposed in [2] and [6] for different tasks.

In this paper, we propose a cascade form predictor, which consists of the following sub-predictors [8],[9],[10]: (1) A nonlinear sub-predictor (NSP), which consists of a multi-layer (ML) neural network with a nonlinear hidden layer and a linear output neuron. (2) A linear sub-predictor (LSP), which is a conventional finite-impulse-response (FIR) filter.

A nonlinearity analysis method for the time series is proposed in order to estimate the minimum effective combination of the input samples and the hidden neurons. Relation between the network size and the learning performance will be discussed. Computer simulation using the sunspot data will be demonstrated.

2. A Cascade Structure Model

2.1 Proposed network structure

The actual time series contains both linear and nonlinear properties and its amplitude is usually continuous value. For this reason, we combine nonlinear and linear predictors in a cascade form. Figure 1 (a) shows the proposed predictor structure. This predictor model is based on a one-step prediction. However, it can be extended to more general prediction.

The nonlinear prediction problem is reduced to a pattern classification using the NSP and linear compensation using LSP. A set of the past samples $x(n-1),...,x(n-N)$ is transformed into the output, which is the prediction of the next coming sample $x(n)$. So, as a first stage of the predictor, we employ a multi-layer neural network which is good for this kind of pattern mapping. It is called a Nonlinear Sub-Predictor (NSP) in this paper. It consists of a sigmoidal hidden layer and a single linear output neuron. The NSP is trained by the supervised learning algorithm using the sample $x(n)$ to be predicted as the target. This means the NSP itself is trained as a single predictor.

However, it is rather difficult to generate the continuous amplitude and to predict linear property. So, we employ a linear predictor after the NSP in order to compensate for the linear relation between the input samples and the target. A finite impulse response (FIR) filter is used for this purpose, which will be called a Linear Sub-Predictor (LSP). The LSP is trained by using $x(n)$ as a target. Thus, the same target is used for both the NSP and the LSP. Figure 1 (b) shows how the LSP works. One of the LSP coefficients ($WO = 1$) passes the NSP output to the overall output of the predictor, and the other coefficients compensate for the remaining (linear) part of the input time series.

In order to confirm the efficiency of the proposed structure, the modified models, described in Sec.4, are used for comparison in computer simulation.

2.2 Network operation and learning algorithm

A set of past N samples of the input signal, $x(n-
1), \(x(n-2), \ldots x(n-N)\) are applied to the NSP and the current sample, \(x(n)\) is used as the desired response for both the NSP and the LSP. \(N\) is the estimated input dimension. The reason why we use \(x(n)\) as a target for the NSP is explained as follows: First, it is difficult to obtain the target only for the nonlinear prediction. It may require separation of nonlinear and linear properties of the time series. Second, since the NSP has a linear output unit, the linear prediction is also possible to some extent. Thus, the NSP output can approach the final target \(x(n)\).

The LSP is an FIR filter of \(K\)-number of taps. The weights of both sub-predictors are adjusted on a pattern-by-pattern basis. The NSP trained by the conventional Back-Propagation algorithm, and the LSP is trained by the LMS algorithm.

### 2.3 System equations of NSP

The output of the \(j\)th hidden neuron, \(y_j(n)\) at the \(n\)th time can be expressed by

\[
\begin{align*}
U_j(n) &= \sum_{i=1}^{N} w_{ji} x(n-i) + \theta_j(n) \\
y_j(n) &= f_h(U_j(n)), \quad j = 1, 2, \ldots, L.
\end{align*}
\]

where \(w_{ji}\) is the connection weight from the \(i\)th input neuron to the \(j\)th hidden neuron and \(\theta_j(n)\) is its bias. The activation function, \(f_h\) used in the hidden layer is a sigmoid function of the form:

\[
f_h(x) = \frac{1}{1 + e^{-x}}
\]

The output layer contains only one linear neuron. Its output value at the \(n\)th time can be expressed by:

\[
u(n) = \sum_i w_j y_j(n) + \theta(n),
\]

\[y_1(n) = f_h(u(n)) = u(n)
\]

\(w_j\) is the connection weight from the \(j\)th hidden neuron to the output neuron. The error of the output unit at the \(n\)th time is

\[
e_{NSP}(n) = d(n) - y_1(n)
\]

where \(d(n)\) is the desired response at the \(n\)th time. The instantaneous squared error of the network is

\[
\xi(n) = \frac{1}{2} e^2_{NSP}(n)
\]

The cost function which has been used as the performance measure is the sum of the squared error over an epoch. It can be written as follows:

\[
\hat{E}_{NSP} = \sum_{n=1}^{M} \xi(n),
\]

where \(M\) is the total number of samples in one epoch.

### 3. Nonlinearity Analysis of Time Series

In order to estimate the minimum size of the proposed predictor, we analyze nonlinearity of the time series of interest. The prediction is equal to mapping a set of the past samples to the next sample to be predicted. The multi-layer neural network is good for this kind of pattern mapping. Still, difficult mapping can exist, which includes the following: Several sets of very similar patterns are mapped into very different samples. The degree of the difficulty of the mapping is closely related to the nonlinearity. The necessary number of the past samples used for prediction, that is the number of the inputs of the NSP, is determined by this nonlinearity analysis. The difficult mapping requires a large number of the past samples. Furthermore, the number of taps of the LSP is determined by the linearity remained at the NSP output.

In this section, we introduce a measure to obtain the...
effective minimum combination of the input samples and
the hidden neurons which enables the network to achieve
its convergence faster than the other networks.

3.1 Input-output mapping

**case 1: Impossible mapping**

A set of the \( N \) samples \( X_n \) is mapped onto the next coming sample \( x(n) \) as

\[
X_n \Rightarrow x(n), \quad n = 1, 2, \ldots, M
\]  

(9)

where \( M \) is the total number of mappings in one epoch, and

\[
X_n = [x(n-1), x(n-2), \ldots, x(n-N)]
\]  

(10)

We consider two different mappings as

\[
X_i \Rightarrow x(i)
\]  

(11)

\[
X_j \Rightarrow x(j)
\]  

(12)

If the above two different mappings satisfy the following relation:

\[
X_i = X_j \quad x(i) \neq x(j),
\]  

(13)

then, they can not be realized by the multi-layer neural network at the same time. If such mappings are exist, the network will fail to converge at all. This problem can be overcome by increasing the number of the input samples \( N \).

**case 2: Difficult mapping**

In this case the two patterns are similar to some extent, and their targets are different from each other. It can be expressed as:

\[
X_i \approx X_j, \quad x(i) \neq x(j)
\]  

(14)

Although this mapping is basically possible, it is still difficult mapping. Although the convergence may be possible, it may often take a very long time. The key question is how to evaluate the degree of this difficulty. We introduce a nonlinearity analysis method for this purpose.

In order to measure the similarity among the sets of the past samples, we employ the Euclidean distance among them as:

\[
d_{ij} = \|X_i - X_j\|, \quad i \neq j
\]  

(15)

Similar sets are selected based on \( d_{ij} \) using some threshold \( I \). If the Euclidean distance between \( X_i \) and \( X_j \) satisfies

\[
d_{ij} \leq I
\]  

(16)

then they are selected as a similar pair. Threshold value, \( I \) is determined by

\[
I = \alpha A_x
\]  

(17)

\[
A_x = \frac{1}{M} \sum_{n=1}^{M} |x(n)|
\]  

(18)

A process of selecting sets of \( X_i \) is as follows: Let the number of \( X_i \) sets to be \( M \), that is, \( \{X_1, X_2, \ldots, X_M\} \). One of these sets, \( X_k \) is selected and find the other \( X_i, i \neq k \) which satisfies

\[
d_{ki} \leq I
\]  

(19)

\( X_i \) is selected as the similar member of \( X_k \). A set of these members is denoted by \( \Omega_k \). Thus,

\[
x_i \in \Omega_k, \quad d_{ki} \leq I
\]  

(20)

\[
X_i \notin \Omega_k, \quad d_{ki} > I
\]  

(21)

\( 1 \leq i \leq M \) and \( i \neq k \)

\( \Omega_k \) is obtained for all data \( X_1 \sim X_M \).

Next, the difference between \( x(i) \) and \( x(j) \), that is, \( \|x(i) - x(j)\| \), is investigated, where both \( X_i \) and \( X_j \) are included in the same \( \Omega_k \). Let \( z_k(i) \) be the corresponding output for the input sample set \( X_i \in \Omega_k \). The variance of \( z_k(i) \) is used to estimate the difference among \( z_k(i) \)

\[
\mu_k = \frac{1}{Q_k} \sum_{i} z_k(i), \quad X_i \in \Omega_k
\]  

(22)

\[
\sigma_k^2 = \frac{1}{Q_k} \sum_{i} (z_k(i) - \mu)^2, \quad X_i \in \Omega_k
\]  

(23)

where \( Q_k \) is the number of elements of \( \Omega_k \). Furthermore, an average of \( \sigma_k^2 \) over all \( \Omega_k \) is used to estimate the difficulty of mapping, that is, the degree of nonlinearity of the entire time series.

\[
\overline{\sigma^2} = \frac{1}{M} \sum_{k=1}^{M} \sigma_k^2
\]  

(24)

For convenience, \( \overline{\sigma^2} \) will be normalized by the signal power.

3.2 Estimation of input dimension of NSP

A large \( \overline{\sigma^2} \) means the similar \( X_i \) is mapped onto the different \( x(i) \), the mapping of this time series is difficult, in other words nonlinearity is high. On the other hand, if \( \overline{\sigma^2} \) is small, the similar \( X_i \) are mapped onto the similar \( x(i) \), then the mapping is easy, and the nonlinearity is low.

Although \( \overline{\sigma^2} \) is large for some number of the past samples \( N \), used in prediction, \( \overline{\sigma^2} \) can be decreased by increasing \( N \). Thus, the necessary number of the past samples, that is the input samples of the NSP is determined by \( \overline{\sigma^2} \). The threshold \( I \) should be appropriately determined.

There is another nonlinearity. \( X_i \) and \( X_j \), whose distance \( \|X_i - X_j\| \) is large, are mapped onto the similar samples \( x(i) \) and \( x(j) \), that is, \( \|x(i) - x(j)\| \) is small. This problem belongs to pattern classification, which is easy problem.
for the multi-layer neural networks under the condition of a small $\sigma^2$.

4. Modified Models for Comparison

In Sec. 2, we have proposed the cascade form predictor structure. Some questions may arise about the order of the combination of the linear and nonlinear processings. Therefore, some modifications are considered here.

In Fig. 2, the LSP part is divided into two parts and the NSP is sandwiched between them. The same number of free parameters as in Fig. 1 are used. It will be called a sandwich model. We also consider that the first two parts of sandwich model represent another separate model in which the LSP and NSP are arranged in reverse order compared with the proposed predictor in Fig. 1. We call this model as a reverse order model. The necessity of using this structure is to answer the question of which is better to use LSP or NSP as the first stage. Later, in Sec. 5, the results of the sandwich model as well as the reverse order model will be compared to that of the proposed model.

In the proposed model, we do not use the LSP in front of the NSP, because the LSP does not work well for nonlinear time series. This point will be investigated through computer simulation.

Figure 3 shows a structure of a multi-layer neural network with direct linear connections from the input layer to the output [3, p.28]. Nonlinear hidden neurons and a linear output neuron are used. It has been stated that: “...this architecture can extract the linearly predictable part early in the learning process and free up the nonlinear resources to be employed where they are really needed”[3].

We have chosen this architecture to compare its simulation results with our proposed structure. Because, this network also try to predict both nonlinear and linear properties using the different structure, by mixing the linear and nonlinear processings in the same network. The network size is chosen to have a very close number of free parameters as that of Figs. 1 and 2.

3. Computer Simulation

5.1 Nonlinear time series

Computer simulations have been done for a one-step ahead prediction task for sunspot time series.

The yearly sunspot time series is used as a benchmark for many years by many researchers. We have used the record of sunspot data from 1700 to 1920 for learning process and the data from 1921 to 1979 for testing process. The same data was used in [1] and [5].

5.2 Nonlinearity analysis

Nonlinearity of the time series are analyzed based on the average variance $\overline{\sigma^2}$ using $I=0.5A_x$, $0.8A_x$ and $0.5A_x$. $I$, $A_x$, and $\overline{\sigma^2}$ are defined by Eqs. (17), (18), and (24) respectively. The values of $I$ are determined by experience. At the present, we do not have a general rule how to determine $I$. However, another important point is the universality of the value of $I$. That is, is it possible to use the same threshold for any nonlinear time series?

Table 1 shows the relations among the average variance $\overline{\sigma^2}$, the threshold $I$ and the number of the past samples N, that is the input samples of the NSP. By increasing the number of the input samples, $\overline{\sigma^2}$ can be decreased. In the last column, $\text{a}\cdot\text{0}$ means that all $\Omega_k$ are empty or $\{x_{ki}\mid x_{ki}^{\text{a}\cdot\text{0}}\}$ take the same value.

5.3 Network size estimation

Network size will be estimated based on the nonlinearity analysis shown in Table 1[8,9,10]. For this purpose, we must know relations among a pair of $I$ and $\overline{\sigma^2}$, the convergence speed and the prediction error. However, these relations are complicated. So, we first analyze their relations, and then estimate the appropriate threshold $I$ and the variance $\overline{\sigma^2}$ for both the convergence speed and the prediction error.

In Table 1, if we select $I=0.5A_x$, then the number of the

<table>
<thead>
<tr>
<th>$N$</th>
<th>$\bar{I}$</th>
<th>$\sigma^2$</th>
<th>$\sigma^2$</th>
<th>$\sigma^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>0.00019</td>
<td>0.00029</td>
<td>0.00009</td>
<td>0.0</td>
</tr>
<tr>
<td>10</td>
<td>0.01356</td>
<td>0.0469</td>
<td>0.00062</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0.02291</td>
<td>0.00469</td>
<td>0.00062</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1. Average Variance for Sunspot Example. $\overline{\sigma^2}$ is normalized by the signal power, 0.1032
Table 2. Average variance for sunspot example:

<table>
<thead>
<tr>
<th>N</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I = 0.5A_x$</td>
<td>0.00044</td>
<td>0.00026</td>
<td>0.00016</td>
<td>0.00009</td>
<td>0</td>
</tr>
<tr>
<td>(T.D.), $\sigma^2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I = 0.5A_x$</td>
<td>0.00052</td>
<td>0.00032</td>
<td>0.00023</td>
<td>0.00006</td>
<td>0</td>
</tr>
<tr>
<td>($y_1$), $\delta$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I = 0.5A_x$</td>
<td>0.00150</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>($e_{N_{SP}}$), $\sigma^2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I = 0.8A_x$</td>
<td>0.00069</td>
<td>0.00042</td>
<td>0.00033</td>
<td>0.00022</td>
<td>0</td>
</tr>
<tr>
<td>(T.D.), $\sigma^2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I = 0.8A_x$</td>
<td>0.00073</td>
<td>0.00049</td>
<td>0.00036</td>
<td>0.00023</td>
<td>0</td>
</tr>
<tr>
<td>($y_1$), $\sigma^2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I = 0.8A_x$</td>
<td>0.0023</td>
<td>0.0002</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>($e_{N_{SP}}$), $\sigma^2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I = A_x$</td>
<td>0.00082</td>
<td>0.00056</td>
<td>0.00041</td>
<td>0.00031</td>
<td>0</td>
</tr>
<tr>
<td>(T.D.), $\sigma^2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I = A_x$</td>
<td>0.00086</td>
<td>0.0006</td>
<td>0.00047</td>
<td>0.00032</td>
<td>0</td>
</tr>
<tr>
<td>($e_{N_{SP}}$), $\sigma^2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This means the nonlinearity of the input signal can be predicted, and the remainder is mainly linear.

Figure 4. Coefficients of LSP.

\[ \text{Figure 5 shows the output waveforms of the different models in the testing phase where the other part of the sunspot data from 1921 to 1979 are used.} \]

\[ \text{The results of different models in both training and testing phases, with the specified size are listed in Table 4. The network size of the proposed and reverse order models are chosen to give the best performance in generalization. The size of the sandwich model is taken to be equivalent to that of the proposed model. The ML-WDC size is slightly larger than the other models. The prediction error is measured at the output of each model.} \]

\[ \text{From these simulation results, in the ML-WDC model, the sandwich model, and the reverse order model, the error is large compared with that of the proposed model.} \]

\[ \text{1979} \]
Table 3  The normalized root mean squared errors (NRMSE) for Proposed model at different LSP taps.
(The * points to better results at specified LSP tap).

<table>
<thead>
<tr>
<th>LSP taps</th>
<th>Learning Phase</th>
<th>Testing Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>0.1402</td>
<td>0.1731</td>
</tr>
<tr>
<td>12</td>
<td>0.1482</td>
<td>0.1696</td>
</tr>
<tr>
<td>10 *</td>
<td>0.1493</td>
<td>0.1684</td>
</tr>
<tr>
<td>8</td>
<td>0.1503</td>
<td>0.1722</td>
</tr>
<tr>
<td>6</td>
<td>0.1503</td>
<td>0.1724</td>
</tr>
<tr>
<td>0</td>
<td>0.1623</td>
<td>0.2013</td>
</tr>
</tbody>
</table>

Table 4 Comparison of NRMSE values among different models.

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Learning Phase</th>
<th>Testing Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>MGWDC model (12-8-1)</td>
<td>0.1747</td>
<td>0.2617</td>
</tr>
<tr>
<td>Sandwich model LSP(5)+NSP(12-8-1) +LSP(5)</td>
<td>0.1854</td>
<td>0.1980</td>
</tr>
<tr>
<td>The reverse order model LSP(6)+NSP(12-8-1)</td>
<td>0.1589</td>
<td>0.2014</td>
</tr>
<tr>
<td>Proposed model NSP(12-8-1)+LSP(10)</td>
<td>0.1493</td>
<td>0.1684</td>
</tr>
</tbody>
</table>

6. Conclusions

A nonlinear predictor connecting the multi-layer neural network (NSP) and the FIR filter (LSP) in a cascade form has been proposed. A nonlinearity analysis method for the time series has been also proposed in order to achieve the fast convergence and the small residual error with the minimum network size. The proposed model has demonstrated its superiority over the other compared models in both learning and testing phases. It has been also confirmed that the number of taps in the LSP is sensitive to generalization of the nonlinear prediction.

References