IEICE TRANS. FUNDAMENTALS, VOL.E82–A, NO.8 AUGUST 1999

PAPER *Special Section on Digital Signal Processing*

A Hybrid Nonlinear Predictor: Analysis of Learning Process and Predictability for Noisy Time Series

Ashraf A.M. KHALAF[†], *Nonmember and* **Kenji NAKAYAMA**[††], *Member*

SUMMARY A nonlinear time series predictor was proposed, in which a nonlinear sub-predictor (NSP) and a linear sub-predictor (LSP) are combined in a cascade form. This model is called "hybrid predictor" here. The nonlinearity analysis method of the input time series was also proposed to estimate the network size. We have considered the nonlinear prediction problem as a pattern mapping one. A multi-layer neural network, which consists of sigmoidal hidden neurons and a single linear output neuron, has been employed as a nonlinear sub-predictor. Since the NSP includes nonlinear functions, it can predict the nonlinearity of the input time series. However, the prediction is not complete in some cases. Therefore, the NSP prediction error is further compensated for by employing a linear sub-predictor after the NSP. In this paper, the prediction mechanism and a role of the NSP and the LSP are theoretically and experimentally analyzed. The role of the NSP is to predict the nonlinear and some part of the linear property of the time series. The LSP works to predict the NSP prediction error. Furthermore, predictability of the hybrid predictor for noisy time series is investigated. The sigmoidal functions used in the NSP can suppress the noise effects by using their saturation regions. Computer simulations, using several kinds of nonlinear time series and other conventional predictor models, are demonstrated. The theoretical analysis of the predictor mechanism is confirmed through these simulations. Furthermore, predictability is improved by slightly expanding or shifting the input potential of the hidden neurons toward the saturation regions in the learning process.

key words: *prediction, nonlinear, time series, neural network, FIR filter, noise robustness*

1. Introduction

The linear signal processing tools are insufficient to deal with nonlinear time series processing. On the other hand, neural networks are useful for nonlinear adaptive signal processing. They have been applied successfully in a variety of signal and information processing fields [1],[2]. One of these fields is the nonlinear time series prediction [3]–[10].

We have considered the nonlinear prediction problem as a pattern mapping one. A multi-layer neural network, which consists of sigmoidal hidden neurons and a single linear output neuron, has been employed as a nonlinear sub-predictor (NSP). Since the NSP includes nonlinear functions, it can predict the nonlinearity of the input series. However, the prediction is not complete in some cases. So, the prediction error of the NSP is further compensated for by employing a linear sub-predictor (LSP) after the NSP. Also, a nonlinearity analysis method for the time series has been proposed to estimate the predictor size [11]–[13].

In this paper, prediction mechanism, that is, a role of the NSP and the LSP will be theoretically discussed. Actual time series usually includes some noise. So, predictability of noisy nonlinear time series by the hybrid predictor will be investigated. Simulation results, using several kinds of nonlinear time series and other conventional predictors, will be demonstrated in order to confirm validity of the theoretical discussions.

Manuscript received December 7, 1998.
Manuscript revised March 4, 1999.

[†]The author is with the Graduate School of Natural Science and Technology, Kanazawa University, Kanazawa-shi, 920–8667 Japan.
[††]The author is with the Department of Electrical and Computer Engineering, Faculty of Engineering, Kanazawa University, Kanazawa-shi, 920-8667 Japan.

2. Hybrid Nonlinear Predictor

2.1 Network Structure

Figure 1 demonstrates the structure of the hybrid predictor [12]. As a first stage of the predictor, we employ a multi-layer neural network (MLNN), which is good for pattern mapping. It is called a nonlinear sub-predictor (NSP). It consists of sigmoidal hidden neurons 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.

Since the NSP includes nonlinear functions, it can predict the nonlinearity of the input time series. However, the prediction is not complete in some cases. So, the NSP prediction error is further compensated for by employing a linear finite impulse response (FIR) sub-

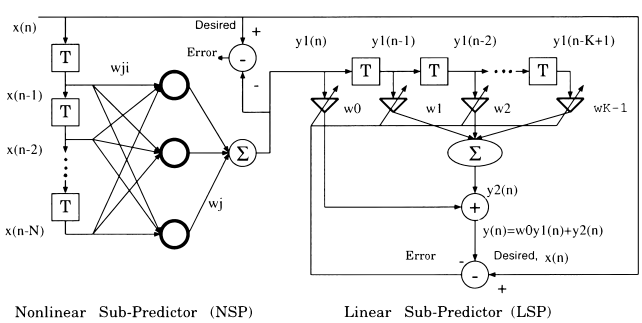

Fig. 1 Structure of the hybrid predictor.

predictor (LSP) after the NSP. The LSP is trained by using $x(n)$ as the target too. Thus, the same target is used for both the NSP and the LSP.

The reason why we use $x(n)$ as the target for the NSP is explained as follows: First, it is difficult to obtain the target only for nonlinear prediction. Second, since the NSP has a linear output neuron, the linear prediction is also possible to some extent. So, nonlinear and some part of linear properties of the input signal can be predicted by the NSP and the remaining part is predicted by the LSP.

2.2 System Equations of NSP

The output of the jth hidden neuron $v_j(n)$ at time n is expressed by

$$u_j(n) = \sum_{i=1}^{N} w_{ji} x(n-i) + \theta_j, \tag{1}$$

$$v_j(n) = f_h(u_j(n)), \qquad j = 1, 2, \ldots L, \tag{2}$$

where w_{ji} is the connection weight from the ith input node to the jth hidden neuron. L is the number of the hidden neurons and θ_j is the bias. The activation function $f_h()$ used in the hidden neurons is a sigmoid function given by

$$f_h(x) = \frac{1}{1 + exp(-x)}. \tag{3}$$

The output layer contains only one linear neuron. Its output $y_1(n)$ is expressed by

$$u(n) = \sum_{j=1}^{L} w_j v_j(n) + \theta, \tag{4}$$

$$y_1(n) = f_o(u(n)) = u(n), \tag{5}$$

w_j is the connection weight from the jth hidden neuron to the output neuron and θ is the bias. The connection weights w_{ji} and w_j, and the biases θ_j and θ in the NSP are adjusted by the back-propagation algorithm.

On the other hand, the LSP coefficients are adjusted by the least mean square (LMS) algorithm. The weights of both sub-predictors are adjusted on a pattern-by-pattern basis.

2.3 Prediction Error Evaluation

The prediction error of the NSP is

$$e_{NSP}(n) = x(n) - y_1(n). \tag{6}$$

The instantaneous squared error of the NSP is

$$\xi_{NSP}(n) = \frac{1}{2} e_{NSP}^2(n). \tag{7}$$

The mean square error MSE over an epoch is

$$MSE_{NSP} = \frac{1}{M} \sum_{n=1}^{M} \xi_{NSP}(n), \tag{8}$$

where M is the number of samples in one epoch. The mean squared error at the LSP output is calculated by the same way. The normalized root-mean-square error ($NRMSE$) will be used to express the prediction error and it is calculated as

$$NRMSE = \sqrt{MSE/P_s}, \tag{9}$$

where MSE indicates the mean squared error at the output of the NSP and the LSP. $P_s = (\mathbf{x^T x})/M$ is the input signal power. $\mathbf{x}$ is the vector contains the input samples. $\mathbf{T}$ is the transposition operator.

3. Prediction Mechanism Analysis

From Eq. (6), $y_1(n)$ can be expressed as

$$y_1(n) = x(n) - e_{NSP}(n). \tag{10}$$

The LSP is the FIR filter with K taps, then its output $y(n)$ can be expressed by

$$\begin{aligned} y(n) &= w_0 y_1(n) + w_1 y_1(n-1) + \ldots + \\ &\quad w_{K-1} y_1(n-K+1). \end{aligned} \tag{11}$$

Substituting Eq. (10) for $w_0 y_1(n)$ in Eq. (11), we have

$$\begin{aligned} y(n) &= w_0(x(n) - e_{NSP}(n)) + w_1 y_1(n-1) + \ldots + \\ &\quad w_{K-1} y_1(n-K+1) \\ &= w_0 x(n) + [-w_0 e_{NSP} + w_1 y_1(n-1) + \ldots + \\ &\quad w_{K-1} y_1(n-K+1)]. \end{aligned} \tag{12}$$

Furthermore, we set

$$\begin{aligned} y_2(n) &= w_1 y_1(n-1) + \ldots + \\ &\quad w_{K-1} y_1(n-K+1)]. \end{aligned} \tag{13}$$

If we can assume $w_0 \approx 1$, then Eq. (12) can be modified as

$$y(n) = x(n) - [e_{NSP}(n) - y_2(n)]. \tag{14}$$

The final prediction error becomes

$$\begin{aligned} e_{final}(n) &= x(n) - y(n) \\ &= e_{NSP}(n) - y_2(n). \end{aligned} \tag{15}$$

This means that the role of the LSP is to predict only the prediction error caused by the NSP. $e_{NSP}(n)$ may include both nonlinearity and linearity. If nonlinearity is dominant, then it cannot be predicted by the LSP. Therefore, it is desirable to predict the nonlinearity by the NSP.

As a result, prediction mechanism can be divided into two stages. In the first stage, nonlinear and some part of linear properties of the input time series are predicted by the NSP. In the second stage, the NSP

prediction error is compensated for by the LSP. This is also another reason why we use the same target for both the NSP and the LSP.

When $w_0 \approx 1$ cannot be held, Eq. (15) will be

$$\begin{aligned} e_{final}(n) &= (1-w_0)x(n) + w_0 e_{NSP}(n) - y2(n) \\ &= [e_{NSP}(n) + (1-w_0)y_1(n)] - y_2(n). \end{aligned} \tag{16}$$

The target of $y_2(n)$ is changed from $e_{NSP}(n)$ to $e_{NSP}(n)+(1-w_0)y_1(n)$. Since $y_1(n)$ is controlled so as to approach to $x(n)$, then $y_1(n)$ contains nonlinearity, which cannot be predicted by the LSP. Therefore, after the learning process, if $w_0 \approx 1$ is not held, the prediction is not optimum by the proposed method. In other words, the hybrid predictor should be optimized so as the condition $w_0 \approx 1$ may be held.

Furthermore, we investigate the contribution of the NSP and the LSP in the overall performance by the following ratio.

$$\beta = P_1/P_2, \tag{17}$$

where, P_1 and P_2 are the power of the NSP output $y_1(n)$ and the one-step linear prediction $y_2(n)$, respectively.

The above discussion concerning the following two items will be investigated through computer simulations.

- The NSP can predict nonlinear property of the time series.
- w_0 is close to unity for good nonlinear prediction.

4. Nonlinearity Analysis of Time Series

Nonlinearity analysis of the time series of interest is very important for estimating size of the NSP and for analyzing the prediction mechanism. A nonlinearity analysis method proposed in [12], will be explained once more because it will be used later in the simulation.

The prediction is equal to mapping a set of the past samples onto the next sample to be predicted. The multi-layer neural network is good for this kind of pattern mapping. 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 input samples of the predictor, is determined by this nonlinearity analysis.

4.1 Input-Output Mapping

4.1.1 Impossible Mapping

We consider N_{T} samples of the time series, that is, $x(1)\cdots x(N_{\mathrm{T}})$. A vector of the past N samples denoted X_n is mapped onto the next coming sample $x(n)$ as

$$X_n = [x(n-1), x(n-2), \ldots, x(n-N)]^T, \tag{18}$$

$$X_n \Rightarrow x(n), \qquad n = N+1, N+2, \ldots, N_{\mathrm{T}}. \tag{19}$$

Letting M be the total number of the above mappings, it is given by $M = N_{\mathrm{T}} - N$.

We consider two different mappings.

$$X_i \Rightarrow x(i), \tag{20}$$

$$X_j \Rightarrow x(j). \tag{21}$$

If the following relation is satisfied

$$X_i = X_j, \qquad x(i) \neq x(j), \tag{22}$$

then, these two different mappings can not be realized by the multi-layer neural network at the same time. If such mappings exist, the network may fail to converge at all. This problem can be overcome by increasing N.

4.1.2 Difficult Mapping

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

$$X_i \approx X_j, \qquad x(i) \neq x(j). \tag{23}$$

Although this mapping is basically possible, it is still difficultmapping. The learning may often take a very long time. The key question is how to evaluate the degree of this difficulty. We have introduced a nonlinearity analysis method for this purpose [12].

In order to measure the similarity among the input patterns, we employ the Euclidean distance among them.

$$d_{ij} = \|X_i - X_j\|, \qquad i \neq j. \tag{24}$$

Similar patterns are selected based on d_{ij} using some threshold$I$. If d_{ij} satisfies

$$d_{ij} \leq I, \tag{25}$$

then X_i and X_j are selected as a similar pair. The threshold I is determined by

$$I = \alpha A_x, \tag{26}$$

where $0 < \alpha \leq 1$ and A_x is expressed by

$$A_x = \frac{1}{M} \sum_{n=N+1}^{N+M} | x(n) | . \tag{27}$$

The process of selecting similar patterns is explained as follows: One pattern X_k of a set $\{X_{N+1}, X_{N+2}, \ldots, X_{N+M}\}$ is selected. Another pattern X_i, $i \neq k$, which satisfies

$$d_{ki} \leq I, \tag{28}$$

is selected as a similar member to X_k. A set of these members is denoted Ω_k. Thus,

$$X_i \in \Omega_k, \qquad d_{ki} \leq I, \tag{29}$$

$$X_i \notin \Omega_k, \qquad d_{ki} > I, \tag{30}$$

where $N+1 \le i \le N+M$, and $i \ne k$. Ω_k is obtained for all $X_{N+1} \cdots X_{N_T}$.

Next, the difference between $x(i)$ and $x(j)$, $\|x(i) - x(j)\|$, is investigated, where both X_i and X_j are included in the same set Ω_k. Let $x_k(i)$ be $x(i)$ for the input pattern $X_i \in \Omega_k$. The variance of $x_k(i)$ denoted σ_k^2 is expressed as

$$\mu_k = \frac{1}{Q_k} \sum_i x_k(i), \qquad X_i \in \Omega_k, \tag{31}$$

$$\sigma_k^2 = \frac{1}{Q_k} \sum_i (x_k(i) - \mu_k)^2, \tag{32}$$

where Q_k is the number of the elements in Ω_k. Furthermore, an average of σ_k^2 overall Ω_k is used to estimate the difficulty of mapping, that is, the degree of nonlinearity of the entire time series.

$$\overline{\sigma_M^2} = \frac{1}{M} \sum_{k=N+1}^{N+M} \sigma_k^2. \tag{33}$$

Furthermore, $\overline{\sigma_M^2}$ is normalized by the variance of the entire time series denoted σ_x^2.

$$\overline{\sigma^2} = \overline{\sigma_M^2}/\sigma_x^2. \tag{34}$$

4.2 Input Dimension Estimation of NSP

4.2.1 Estimation Based on $\overline{\sigma^2}$

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

Although $\overline{\sigma^2}$ is large for a small number of the past N samples used in prediction, $\overline{\sigma^2}$ can be decreased by increasing N. Thus, the necessary number of 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 similar samples $x(i)$ and $x(j)$, whose distance $\|x(i) - x(j)\|$ is small. This problem belongs to pattern classification, which is an easy problem for the multi-layer neural networks.

We can use a larger number of the NSP input samples than that estimated by $\overline{\sigma^2}$. However, it will cause over learning, and good generalization is not guaranteed. Thus, the input samples should be limited under the upper bound estimated by $\overline{\sigma^2}$.

4.2.2 Estimation Based on Nonlinear Model

The nonlinearity of the time series can be estimated by $\overline{\sigma^2}$ introduced in Eq. (34). This means if $\overline{\sigma^2}$ is large, then the linear prediction is difficult. However, in the nonlinear prediction by the NSP, relation between the nonlinear function $f_h()$ given by Eq. (3) and the model which generates the nonlinear time series must be taken into account. Consider an example given by

$$u(n) = a_1 x(n-1) + a_2 x(n-2), \tag{35}$$

$$x(n) = f_h(u(n)). \tag{36}$$

Two input nodes and one output neuron with $f_h()$ are enough for predicting $x(n)$. However, $\overline{\sigma^2}$ of $x(n)$ generated by Eq. (36) is not small.

In natural phenomena, generating models of nonlinear time series are usually very complicated. It is often difficult to express the nonlinearity by a simple equation. Even though the explicit equations of the generating models are not obtained, the prediction can be regarded as "pattern mapping." Therefore, the NSP input dimension will be estimated by $\overline{\sigma^2}$.

Consequently, the lower bound and the upper bound of the NSP input dimension are given by the model equation and $\overline{\sigma^2}$, respectively. We can search for the optimum network size between them from a generalization view point. The generalization means the prediction performance for testing data which are not used in a training phase. How to combine $\overline{\sigma^2}$ and the nonlinear model to estimate the NSP input dimension will be more discussed in another paper.

In Sects.5 and 6, some examples will be shown regarding the above relations.

5. Simulation Results Using Hybrid Model

5.1 Nonlinear Time Series

Computer simulations have been done for a one-step-ahead prediction task using three examples: Sunspot data, Lake data and Chaotic data. Data file of Sunspot time series is downloaded from Santa Fe public home page via [5] statements. Lake data and Chaotic data files are taken from the foppy disc accompanied with [14]. They are shown in Fig. 2

Sunspot data is used as a benchmark for many years by many researchers. We have used the record of the sunspot data from the year 1700 to 1920 (221 samples) in the training phase and the data from 1921 to 1979 (59 samples) in the testing phase. The same data were used in [3] and others. Lake data represent the level of Lake Huron in the years 1875–1972 [14]. The first 50 samples are employed in the training phase and the latter 48 samples are employed in the testing phase. Chaotic data is generated by the following equation.

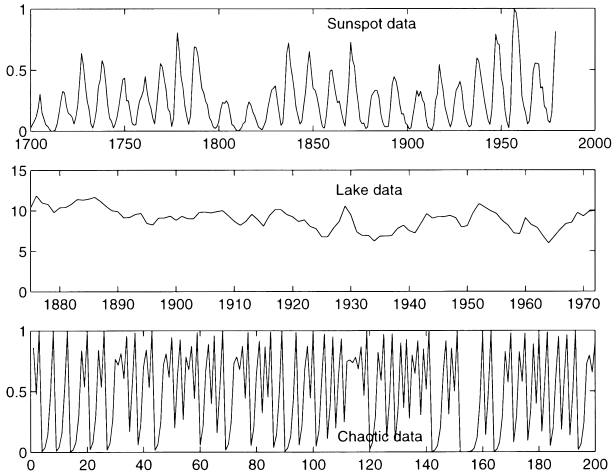

Fig. 2 Time series used in simulation.

$$x(n) = 4x(n-1)(1-x(n-1)), 0 < x(0) < 1. \quad (37)$$

The first 150 samples are used in the training phase and the latter 50 samples are used in the testing phase.

5.2 Network Size Estimation

The NSP input dimension is estimated based on the nonlinearity analysis discussed in Sect.4. The number of hidden neurons and the number of the LSP taps are estimated by try-and-error criterion taking the generalization performance into account.

Table 1 shows the relations among the average variance $\overline{\sigma^2}$ given by Eq. (34), the threshold I and the number of the past N samples. I, II and III represent $I = 0.5A_x$, $0.8A_x$ and A_x, respectively. a, b and c indicate Sunspot, Lake and Chaotic data, respectively. By increasing N, $\overline{\sigma^2}$ can be decreased.

The Chaotic data (c) has large $\overline{\sigma^2}$. Therefore, its nonlinearity is high. This means the chaotic time series is difficult to be predicted by a linear predictor.

On the contrary, the Lake data has small $\overline{\sigma^2}$ except for N=4,6. Its nonlinearity is somewhat low. Thus, it can be predicted by a linear predictor to some extent.

We can estimate the NSP input dimension based on these results. As discussed in Sect.4.2, the nonlinearity analysis by using $\overline{\sigma^2}$ provides the upper bound and the time series generating model should be taken into account to determine the lower bound of the input dimension.

As a result, the following network sizes are determined from a generalization view point. The network size of the hybrid models, NSP(Input nodes-Hidden neurons-Output neurons)+ LSP taps, become (12-8-1)+ 10, (8-8-1)+5 and (4-6-1)+5 for Sunspot, Lake and Chaotic data, respectively.

Sunspot data requires N=12, where $\overline{\sigma^2}$=0 with I=A_x. Similar result was obtained in [15] and used in [3]. For Lake data, N=8 is optimum, because the nonlinearity is lower than that of Sunspot data. Since Chaotic data are generated by the simple equation given by Eq. (37), the NSP input dimension can be reduced to 4 in spite of the highest nonlinearity. Thus, we must find the optimum size under the upper bound determined by $\overline{\sigma^2}$.

Table 1 $\overline{\sigma^2}$ given by Eq. (34). I, II and III represent $I = 0.5A_x$, $0.8A_x$ and A_x, respectively. a, b and c indicate Sunspot, Lake and Chaotic data, respectively.

N		4	6	8	10	12
I	a	0.122	0.044	0.001	0	0
	b	0.2173	0.0061	0	0	0
	c	0.209	0.096	0.040	0.011	0
II	a	0.205	0.110	0.034	0	0
	b	0.3866	0.1640	0.0113	0	0
	c	0.462	0.258	0.130	0.046	0.011
III	a	0.259	0.154	0.056	0.002	0
	b	0.4555	0.2733	0.0405	0.0010	0
	c	0.617	0.405	0.156	0.052	0.011

Table 2 $\overline{\sigma^2}$ given by Eq. (34) for Sunspot data. $I = A_x$.

Input samples, N	2	3	4	5	12
$x(n)$	0.4719	0.3250	0.2375	0.1813	0
$y_1(n)$	0.4156	0.2906	0.2281	0.1531	0
$e_{NSP}(n)$	0.0813	0.0156	0.0006	0	0

5.3 Prediction Mechanism Analysis

Table 2 demonstrates the nonlinearity analysis results for Sunspot data. $\overline{\sigma^2}$ of the input signal $x(n)$, the NSP output $y_1(n)$ in Eq. (5) and the error signal $e_{NSP}(n)$ in Eq. (6) are shown. $\overline{\sigma^2}$ of $y_1(n)$ is close to that of $x(n)$. On the other hand, $\overline{\sigma^2}$ of $e_{NSP}(n)$ is well reduced. This means the nonlinearity of the input signal is well predicted by the NSP.

The LSP coefficients W=$[w_0, w_1, \ldots, w_{K-1}]$ after training are $W_{Sunspot}$=[1.0266, 0.0079, −0.0215, 0.0772, −0.0651, 0.0502, 0.0479, −0.0885, 0.0951, −0.0162], W_{Lake}=[0.9993 −0.2067 0.0763 0.0667 0.0521] and $W_{Chaotic}$=[1.0032, −0.0032, 0.0031, 0.0048, 0.0009]. From these results, it is clear that the first element w_0 is very close to unity.

From the results shown in Table 2 and the LSP coefficients, the prediction mechanism of the hybrid predictor theoretically discussed in Sect.3 are verified. Namely, the NSP can predict the nonlinearity of the input time series. The LSP works to predict the NSP prediction error $e_{NSP}(n)$ as shown in Eq. (15).

The ratio β of the NSP to the LSP output powers defined in Eq. (17) is 110.8, 558.8 and 14561 for Sunspot, Lake and Chaotic data, respectively. β is very large in all cases. This means the NSP can predict the main part of the time series. Nevertheless, the prediction performance is further improved by using the LSP as will be shown in Sect.6.

Thus, the role of the NSP and the LSP are clarified. The main prediction is done by the NSP, and the remaining part is compensated for by the LSP.

6. Comparison with Other Models

In this section, we compare the prediction performance of the hybrid predictor, a linear FIR predictor and a nonlinear MLNN predictor with a linear output neuron. More comparison using other four kinds of predictor was demonstrated in [12]. The results of computer simulation using three kinds of the time series are tabulated in Table 3. The MLNN predictor size is set to be the same as that of the NSP in the hybrid predictor, in order to examine the efficiency of using the LSP.

Compared to the other models, the hybrid predictor has the minimum prediction errors in all cases. However, in the case of Lake data, the difference between the linear predictor and the others is small. Because the nonlinearity is not so high for Lake data as shown in Table 1.

On the other hand, in the case of Chaotic data, since the nonlinearity is high, the linear predictor does not work well. Performance of the MLNN predictor is almost the same as that of the hybrid predictor. Contribution of the LSP is also very small, which can be explained by a very large β=14561.

Figure 3 shows the output waveforms of Chaotic data using different models in the testing phase.

Table 3 Comparison of $NRMSE$ among different models using three kinds of time series. $NRMSE$ is evaluated in testing phase.

Model	Sunspot	Lake	Chaos
FIR	12 taps	8 taps	12 taps
	0.3831	0.0859	0.4400
MLNN	(12-8-1)	(8-8-1)	(4-6-1)
	0.2013	0.0721	0.0177
Hybrid	(12-8-1)+10 taps	(8-8-1)+5 taps	(4-6-1)+5 taps
	0.1684	0.0672	0.0155

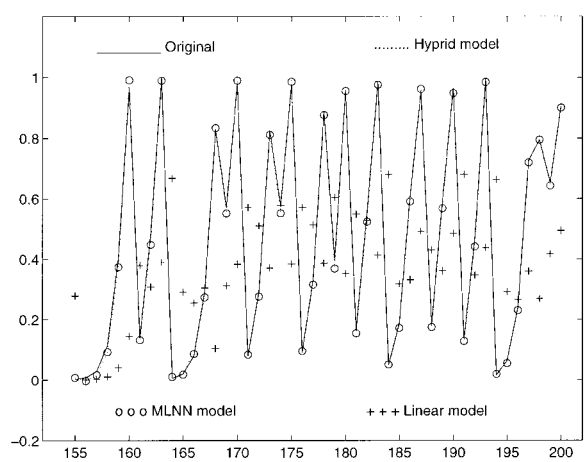

Fig. 3 Predictor output waveforms for Chaotic data using different predictors.

7. Effects of Noise in Nonlinear Prediction

In measuring physical phenomena, data transmission and processing, noise cannot be avoided. Therefore, in real world applications, noise effects must be investigated.

7.1 Training Using Noisy Time Series

Training phase:
It is assumed that we can get noise-free time series and a probability distribution function of noise. The training is carried out using the noisy data as the input and the noise-free data as the target. The noise used here is Gaussian white noise. The training data sets are prepared by adding 10 noise sets to the noise-free time series. So, 10 noisy training data sets are used in one epoch.

Effect of the training using the noisy time series is evaluated. Especially, distribution of the input potential of the hidden layer neurons, that is, $u_j(n)$ given by Eq. (1), is investigated. The sigmoidal functions are used in the hidden neurons. If $u_j(n)$ for all the noisy data can be distributed mainly in the saturation regions, then noise effects can be suppressed.

On the other hand, when spectra of the signal and the noise are separated, noise reduction is also possible by linear filters. However, the above separation is not always guaranteed especially in nonlinear time series.

Testing phase:
After the training, the predictors are tested using other samples of the time series and thenoise, which are not used in the training phase. In this phase, the input signal and the target are the noisy data and the noise-free data, respectively. Furthermore, we employ the following reference in order to evaluate the prediction performance.

$$R = \sqrt{(MSE_{nf} + P_n)/P_s}. \tag{38}$$

MSE_{nf} and P_n are the mean square prediction error for the noise-free time series and the noise power, respectively. The normalization by P_s is the same condition as the previous measure defined in Eq. (9). The meaning of R is the following: Since the noise used here is white noise, which cannot be predicted, the noise will remain just as it is. Therefore, the mean squared prediction error becomes $MSE_{nf} + P_n$. If the prediction error obtained in the simulation is less than R, then it can be said that the noise effect is compensated for through the training, at the same time, the predictor becomes robust against the additive noise.

7.2 An Enhanced Learning Method

As discussed in Sect.7.1, if the input potential $u_j(n)$ of the hidden neurons are located in the saturation regions of the sigmoid function $f_h()$ in Eq. (3), effects of the noise included in the input time series can be suppressed. Since $u_j(n)$ is given by Eq. (1), it can be shifted or expanded toward the saturation regions by enlarging w_{ji} and θ_j. For this reason, the following enhanced learning method is proposed for the NSP.

Stage 1: The NSP is trained by the back-propagation algorithm in an ordinary fashion.

Stage 2: The NSP trained in Stage 1 is further trained through the modified back-propagation algorithm, in which the following enhancement is embedded.

Let $w_{ji}(n)$ and $\theta_j(n)$ be the connection weights and the bias updated at the nth-epoch through the back-propagation algorithm. At the same epoch, they are further enhanced as follows:

$$\begin{aligned}&(1+r^n)w_{ji}(n), \quad 0<r<1,\\ &(1+r^n)\theta_j(n), \quad 0<r<1.\end{aligned} \tag{39}$$

These values are denoted $w_{ji}(n)$ and $\theta_j(n)$, respectively, once more, and are used in the $(n+1)$-th epoch of the back-propagation algorithm. r is determined by experience, resulting in a small value. This means effect of r^n will be vanished within some earlier epochs.

7.3 Simulation Results and Discussions

The nonlinearity, that is, $\overline{\sigma^2}$ of the noisy time series was analyzed in our works. It is almost the same as that of the noise-free time series, then the same network sizes are used. The data are not shown in this paper.

Table 4 shows the $NRMSE$ and R using three kinds of predictors trained by using the noisy time series. In the case of Sunspot data, the hybrid predictor is the best. The MLNN and the hybrid predictors have almost the same $NRMSE$ for Lake and Chaotic data. The linear predictor works better in the case of Lake data than that of Chaotic data, because of the high nonlinearity of Chaotic data. Spectrum of Sunspot and Lake data are somewhat concentrated within some frequency band, however, that of Chaotic data is spread over a whole frequency band.

However, all the $NRMSE$ are larger than R, then nonlinear prediction obtained by training using the noisy time series is affected by the noise.

Table 5 shows the $NRMSE$ for the hybrid predictor trained by the normal back-propagation (NBP) and the enhanced back-propagation (EBP) algorithms using both the noise-free and the noisy training data. Sunspot data are used with the signal to noise ratio S/N=29.5 dB, then R=0.172. The $NRMSE$ obtained by the NBP using the noise-free training data is almost the same as R. This means the white noise cannot be predicted. It remains as it is as assumed in Sect.7.1. When the noisy training data are used, the $NRMSE$ is increased.

On the contrary, the EBP algorithm with r=0.01 in Eq. (39), can reduce the $NRMSE$ under R. This means noise effects can be suppressed by the new training method. In this method, still the noisy training data is not useful.

Figure 4 shows the histogram of the input potential $u_j(n)$ of two hidden neurons. The distribution of $u_j(n)$ obtained by the EBP algorithm are expanded and shifted toward the saturation region of the sigmoidal function compared to those obtained by the NBP algorithm as expected in Sect.7.2.

Table 4 Comparison of $NRMSE$ among different models trained by using noisy time series. S/N=29.5 dB.

Model	Sunspot	Lake	Chaos
R	0.1720	0.0742	0.0370
FIR	12 taps	8 taps	12 taps
	0.2343	0.0941	0.4396
MLNN	(12-8-1)	(8-8-1)	(4-6-1)
	0.2296	0.0870	0.0664
Hybrid	(12-8-1)+10 taps	(8-8-1)+5 taps	(4-6-1)+5 taps
	0.1864	0.0875	0.0655

Table 5 $NRMSE$ for noisy Sunspot data using hybrid predictor trained by normal and enhanced back-propagation algorithms. S/N=29.5 dB.

Learning method	NBP	EBP	R
Noise-free training data	0.170	0.156	0.172
Noisy training data	0.186	0.168	

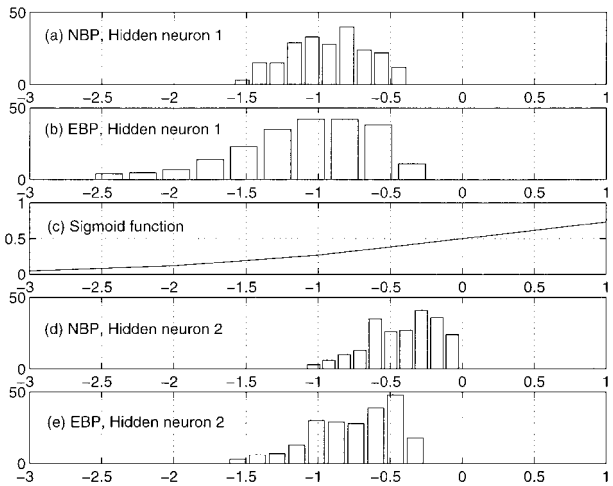

Fig. 4 Histogram of input potential for two hidden neurons of NSP trained by NBP and EBP algorithms. Sunspot data are used.

8. Conclusions

In this paper, the prediction mechanism and a role of the NSP and the LSP of the hybrid predictor have been theoretically and experimentally analyzed and clarified. At the NSP output, the nonlinearity and some part of linearity of the input time series is predicted. A role of the LSP is to predict the NSP prediction error. Predictability for the noisy time series has been investigated. Training using the noisy time series is not useful. In the back-propagation algorithm, slightly enhancing the connection weights and the bias can reduce the noise effects.

Computer simulations have been demonstrated using the linear, the MLNN and the hybrid predictors for Sunspot, Lake and Chaotic data. Properties of these predictors are analyzed taking the nonlinearity of the time series into account.

References

[1] S. Haykin, Neural Networks: A Comprehensive Foundation, Macmillan, New York, 1994.

[2] J. Hertz, A. Krogh, and R.G. Palmer, Introduction to the Theory of Neural Computation, Santa Fe Institute, 1991.

[3] A.S. Weigend and D.E. Rumelhart, "Generalization through minimal networks with application to forecasting," Proc. INTERFACE'91: Computing Science and Statistics, ed. Elaine Keramindas, pp.362–370, Springer Verlag, 1992.

[4] S. Haykin and L. Li, "Nonlinear adaptive prediction of nonstationary signals," IEEE Trans. Signal Processing, vol.43, no.2, pp.526–535, Feb. 1995.

[5] A.S. Weigend and N.A. Gershenfeld, Time series prediction: Forecasting the future and understanding the past, Proc. V. XV, Santa Fe Institute, 1994.

[6] M. Kinouchi and M. Hagiwara, "Learning temporal sequences by complex neurons with local feedback," Proc. ICNN'95, pp.3165–3169, 1995.

[7] T. Matsumoto, H. Hamagishi, and Y. Chonan, "A hierarchial bayes approach to nonlinear time series prediction with neural nets," Proc. ICNN'97, pp.2028–2033, 1997.

[8] T.J. Cholewo and J.M. Zurada, "Sequential network construction for time series prediction," Proc. ICNN'97, pp.2034–2038, 1997.

[9] A. Atia, N. Talaat, and S. Shaheen "An efficient stock market forecasting model using neural networks," Proc. ICNN'97, pp.2112–2115, 1997.

[10] X.M. Gao, X.Z. Gao, J.M.A. Tanskanen, and S.J. Ovaska, "Power prediction in mobile communication systems using an optimal neural-network structure," IEEE Trans. Neural Networks, vol.8, no.6, pp.1446–1455, Nov. 1997.

[11] A.A.M. Khalaf, K. Nakayama, and K. Hara, "A neural-FIR predictor: Minimum size estimation based on nonlinearity analysis of input sequence," Proc. ICANN'97, pp.1047–1052, Lausanne, Switzerland, Oct. 1997.

[12] A.A.M. Khalaf and K. Nakayama, "A cascade form predictor of neural and FIR filters and its minimum size estimation based on nonlinearity analysis of time series," IEICE Trans. Fundamental, vol.E81-A, no.3, pp.364–373, March 1998.

[13] A.A.M. Khalaf and K. Nakayama, "Time series prediction using a hybrid model of neural network and FIR filter," Proc. of IJCNN'98, pp.1975–1980, Anchorage, Alaska, May 1998.

[14] P.J.Brockwell and R.A.Davis, Introduction to Time Series and Forecasting, Springer Verlag, 1996

[15] H. Tong and K.S. Lim, "Threshold autoregression, limit cycles and cyclical data," Journal Royal Statistical Society B, vol.42, pp.245–292, 1980.

Ashraf A.M. Khalaf received the B.Sc. and M.Sc. degrees in Electrical Engineering from Minia University, Minia, Egypt, in 1989 and 1994, respectively. From 1989 to 1995 he was working as a Teaching Assistant in the Department of Electrical Engineering, Faculty of Engineering and Technology, Minia University. Since April, 1996 he has been a Ph.D. candidate with the Graduate School of Natural Science and Technology, Kanazawa University. His current research interests include neural networks. A.A.M. Khalaf is an IEEE student member.

Kenji Nakayama received the B.E. and Dr. degrees in Electronics Engineering from Tokyo Institute of Technology (TIT), Tokyo, Japan, in 1971 and 1983, respectively. From 1971 to 1972 he was engaged in the research on classical network theory in TIT. He was involved in NEC Corporation from 1972 to 1988, where his research subjects were filter design methodology and signal processing algorithms. He joined the Department of Electrical and Computer Engineering at Kanazawa University, in Aug. 1988, where he is currently a Professor. His current research interests include neural networks, adaptive signal processing, and signal theory. Dr. Nakayama is a senior member of IEEE and a member of INNS.