A Hybrid Nonlinear Predictor and Its Robustness for Noisy Time Series Prediction
混合形非線形予測器とその雑音時系列予測特性

A.A.M. Khalaf　K. Nakayama
カラフ・アシュラフ　中山　謙二
Graduate School of Natural Sceince & Tech., Kanazawa University
金沢大学大学院　自然科学研究科

あらまし
実際の信号は線形性と非線形性を含んでいる。信号の振幅は連続値を取る。このため、非線形予測器と線形予測器を縦続接続した混合形予測器を提案した。本論文では、この予測器における予測のメカニズムを解析する。次に、雑音を含む時系列信号の予測における学習方法と予測能力を評価する。実際に観測されたいくつかの時系列信号を用いて、計算機シミュレーションを行い評価した。

Abstract

Real time series contains both linear and nonlinear properties. Its amplitude is usually continuous value. For these reasons, we combine nonlinear and linear sub-predictors in a cascade form. Here, we are emphasizing on the mechanism analysis of our hybrid model and testing its performance in the noisy environment. Computer simulations using real-world and artificially generated time series have demonstrated the efficiency of the predictor and its robustness for some noisy time series.

1 Introduction

The linear signal processing tools are insufficient to deal with nonlinear time series processing (i.g. predicting, modeling, and characterizations). 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. One of these fields is the nonlinear time series prediction [1], [2], and others. Neural networks were first applied to time series prediction by Lapedes and Farber (1987) [1].

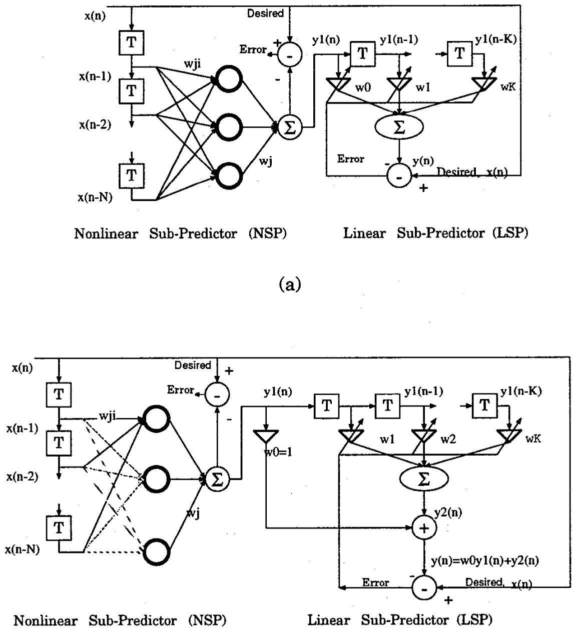
In practice, many of the time series include both nonlinear and linear characteristics. Furthermore, the amplitude of the time series is usually continuous value. Therefore, it is useful to use a combined structure of nonlinear and linear models to deal with such signals. Other hybrid structures were proposed in [2] and [5] for different tasks.

In this paper, we are emphasizing on the analysis of our hybrid prediction model proposed in [6] and its operating mechanism theoretically and through computer simulation of many other time series. Also, we will test the predictor performance under the noisy environment condition using white noise and different signal to noise ratios.

2 Neural and Linear Predictor

Figure 1 (a) shows the hybrid predictor structure. It consistst of two subpredictors, nonlinear-subpredictor (NSP) which is represented by mulilayer neural network and the linear - subpredictor (LSP) which is represented by linear filter.

The nonlinear prediction problem is reduced to a pattern classification problem. A set of N 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 (MLNN) 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 out-



(a)

(b)

Fig. 1.(a) Structure of the proposed hybrid predictor, (b) The same detailed model for analysis.

put 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 using NSP. 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 too. Thus, the same target is used for both the NSP and the LSP. The propposed predictor will be employed for one-step prediction task as an example. However, it can be extended to more general prediction.

2.1 System equations of NSP

The output of the jth hidden neuron, $y_j(n)$ at the nth time can be expressed by

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

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

where w_{ji} is the connection weight from the ith input neuron to the jth 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:

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

The output layer contains only one linear neuron. Its output value at the nth time can be expressed by:

$$u(n) = \sum_j w_j y_j(n) + \theta(n), \tag{4}$$

$$y1(n) = f_o(u(n)) = u(n) \tag{5}$$

w_j is the connection weight from the jth hidden neuron to the output neuron.
The error of the output unit at the nth time is

$$e_{NSP}(n) = d(n) - y1(n) \tag{6}$$

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

$$\xi(n) = \frac{1}{2}e^2_{NSP}(n) \tag{7}$$

The cost function which has been used at NSP output is the mean square error (MSE) value over an epoch (See appendix A for LSP trainig algorithm). It can be written as

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

where M is the total number of samples in one epoch. For conveneince, we normalize the MSE vlaue by the input signal power and take the square root as follows

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

In the case of using noisy time sereis, we define an additional measure reference to compare the value of $NRMSE$ with that reference. This reference can be expressed as

$$R = \sqrt{(MSE_s + P_n)/P_s} \tag{10}$$

where MSE is the prediction mean square error, MSE_s is the prediction mean square error for noise-free signal, P_s, the input power of the signal, and P_n is the noise power.

2.2 Mechanism analysis of hybrid predictor

Figure 1. (b) is the same hybrid model structure in which the LSP is drawn in easy way to understand its mechanism and function inside the overall model.

The output of NSP will be the input signal to LSP. The LSP is an FIR filter of K-number of taps. Theoretically saying, The LSP coefficients will be updated so as to one of these coefficients, ($w_0 \approx 1$) 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. This claim will be verified by simulation results.

Furthermore, we will investigate the contribution of NSP and LSP in the overall output performance of the hybrid predictor by computing the ratio of there output powers. This ratio is

$$\beta = P_1/P_2, \quad (11)$$

where, P_1 is the power of signal $w_0 y1(n)$, passed from NSP to the overall output, and P_2 is the power of signal $y2 = w_1 y1(n-1) + w_2 y1(n-2) + ... + w_K y1(n-K)$ at the LSP output which represents the linear compensation term. By this ratio we can determine the contribution of each sub-predictor in the overall performance of the hybrid model. also we can determine the dominance of linearty or nonlinearity in the input time series.

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 Network size estimation

Nonlinearity of the time series are analyzed based on the average variance $\overline{\sigma^2}$ using a threshold I=A_x, $0.8A_x$ and $0.5A_x$. Where A_x is mean value of the input time series [6]. Input dimension will be taken to correspond to minimum $\overline{\sigma^2}$. (See [6] for more details). The NSP minimum hidden neurons will be determined by try and error criteria. Furthermore, the order of the LSP is determined taking the prediction generalization into account.

3 Simulation Results Using Hybrid Predictor

3.1 Nonlinear time series

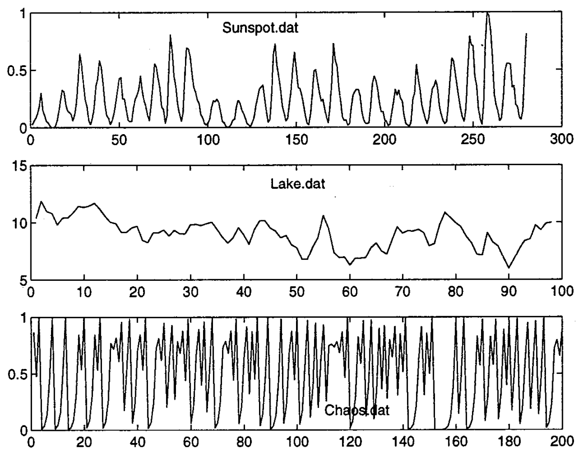


Fig. 2. Time series used in simulation.

Computer simulations have been done for a one-step ahead prediction task for three examples; Sunspot data, Lake data (LAKE.DAT), and Chaotic data (CHAOS.DAT) shown in Figure 2. Data file of Sunspot time series is downloaded from Santa Fe public home page. LAKE.DAT and CHAOS.DAT files are taken from a foppy disck acompanied with [5].

Table 1 demonstrates the values of $\overline{\sigma^2}$ and its relation with the number of input samples N.

Table 1: Average Variance for Sunspot Example

Input Samples, N		8	9	10	12
$I = 0.5A_x$,	$\overline{\sigma^2}$x10^{-4}	0.2	0	0	0
$I = 0.8A_x$,	$\overline{\sigma^2}$x10^{-4}	14	0.3	0.1	0
$I = A_x$,	$\overline{\sigma^2}$x10^{-4}	23	4.84	0.64	0

Table 2 demonstrates the results of analysis of the training data (T.D), the output of NSP, y1(n) in Eq.(5), and the error signal, $e_{NSP}(n)$ in Eq.(6) from the point of view of the above nonlinearity analysis method. All $\overline{\sigma^2}$ values in this table are normalized by their related powers. In this table we see that the nonlinearity of NSP output is close to the nonlinearity of the input signal, training data. On the other hand, the nonlinearity of the difference be-

Table 2: Average variance for sunspot example, $\overline{\sigma^2}$x10^{-4}.

Input Samples, N	2	3	4	5	12
(T.D) $I = 0.5A_x$	4.4	2.6	1.6	0.9	0
(y1) $I = 0.5A_x$	5.2	3.2	2.3	0.6	0
(e_{NSP}) I= 0.5A_x	15	0	0	0	0
(T.D) $I = 0.8A_x$	6.9	4.2	3.3	2.2	0
(y1) $I = 0.8A_x$	7.3	4.9	3.6	2.3	0
(e_{NSP}) I= 0.8A_x	23	2	0	0	0
(T.D) $I = A_x$	8.2	5.6	4.1	3.1	0
(y1) $I = A_x$	8.6	6	4.7	3.2	0
(e_{NSP}) I=A_x	26	5	0	0	0

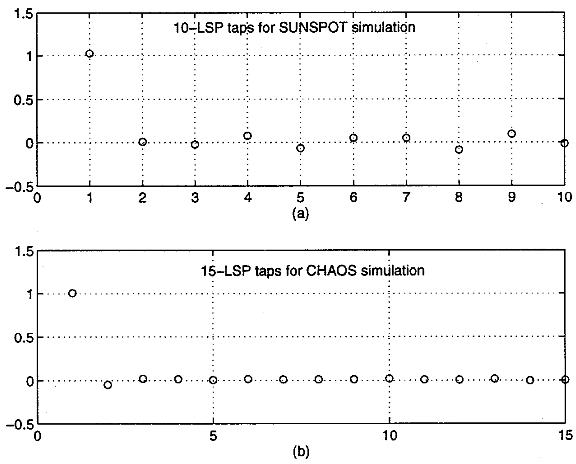


Fig. 3. Coefficients of LSP after training process.

tween them, nonlinearity of $e_{NSP}(n)$ is well reduced. This means the nonlinearity of the input signal can be predicted by NSP, and the remaining part has mainly linear property.

Figure 3. (a) and (b) show the values of the LSP coefficients after training using Sunspot data and Chaos data respectively. The following two vectors, $W1$, $W2$ contain the values of LSP coefficients after training process for Sunspot data and CHAOS data depicted in Figure 3 (a) and (b) respectively, $W1$ =(1.0266, 0.0079, -0.0215, 0.0772, -0.0651, 0.0502, 0.0479, -0.0885, 0.0951, -0.0162), and $W2$ =(1.0065, -0.0481, 0.0197, 0.0140, 0.0056, 0.0171, 0.0123, 0.0117, 0.0109, 0.0197, 0.0076, 0.0066, 0.0164, -0.0019, 0.0034). From this result, it is clear that one of the LSP coefficients (w_0) is updated to reach the value 1, so it can pass the NSP output to the overall output. The other coefficeints compensate for the linear property of the input time series. This result supports the theoretical discussion in Sec. 2.2.

4 Comparison with Other Models

In [6], our hybrid predictor has been compared with other models. Briefly, these models are multilayer neural network with direct linear connections (MLNN-WDC) from its input layer to the output, sandwich model, and reverse order model. The sandwich model is that model in which the LSP part is divided into two parts and the NSP is sandwiched between them. Reverse order model is that model in which the LSP and NSP are arranged in reverse order compared with the proposed predictor[6].

Here, we will compare simulation results of the hybrid model with that of using other models which are linear predictor using FIR model and the conventional multilayer neural network, MLNN. The overall performance measure which will be used here is the $NRMSE$ Eq. (9).

4.1 Simulation for comparison among different models

The results of computer simulation using the three noise-free time series are tabulated in Table 3. In this table, the hybrid model has the smallest $NRMSE$. However, in case of Lake data there is no big difference between linear predicror and the hybrid predictor. Although, the hybrid predictor still has the smallest $NRMSE$ for noise-free Lake data, it may be better to use linear predictor for Lake data according to the simulation of noisy Lake data in the next section.

Figure 4 shows the output waveforms of the different models in the testing phase where the other part of the Chaotic data from sample 163 to 188 are used.

From these simulation results, It is obvious that the proposed model has a superiority over other models. It has a smaller residual error in learning and testing phases and best data fitting for nonlinear time series prediction.

Table 3: Comparison of prediction NRMSE values among Different Models in case of noise-free time series.

Model Name/Signal	Sunspot	Lake	Chaos
Linear predictor(FIR)	0.3831	0.0880	0.4400
MLNN predictor	0.2013	0.0892	0.1232
Proposed model	0.1684	0.0865	0.1024

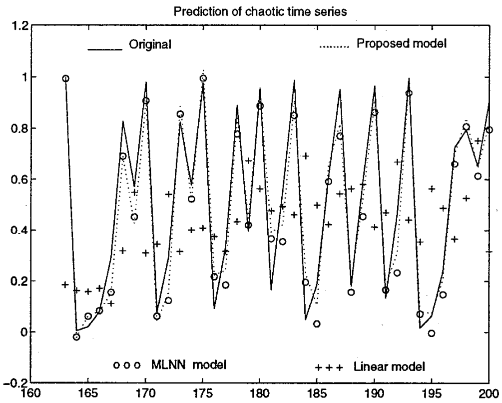


Fig. 4. Chaotic Ex.: Prediction of 38 of CHAOS.DAT from 163 to 200.

5 Robustness of Hybrid Predictor for Noisy Time Series

In this section we will test our model using the noisy data with different signal to noise ratios. In order to have a model robustness, the prediction MSE of the signal without noise plus the noise power must be the upper limit of the prediction MSE of the noisy signal. This condition is verified in our predictor for the chaotic time series example. For high signal to noise ratios, the predictor can work well, however, for lower signal to noise ratios (high noise power), this case represents a common problem for any predictor.

The predictor is trained using a noisy data and noise-free target in the learning phase. In test phase, the input to predictor is noisy and the reference signal is noise-free.

Using noisy data in the learning phase in the hybrid predictor gives a good solution to overcome the noise effect on the prediction performance. This solution is that, in the NSP stage the input potential distribution to hidden neurons will be expanded and shifted toward the saturation regions of the sigmoid function. This will reduce the predictor sensitivity to noise effect and improves its performance. This point will be demonstrated in the next subsection. For noisy time series, we will use an additional measure reference which is R expressed by Eq.(10).

Table 4 demonstrates the values of $NRMSE$ and R of the hybrid predictor for chaotic time series for different signal to noise ratios, S/N. Three cases are taken into account, (1), (2), and (3) corresopond to $S/N = 35.5dB$, $29.5dB$, and $23.5dB$ respectively. White noise is employed in our simulations, 10 noise-data sets are used every one epoch and the avarage is taken over 10 to calculate the average MSE. The

Table 4: $NRMSE$ of the noisy chaotic time series.

Signal to noise ratio S/N	$NRMSE$	R
(1) $S/N = 35.5dB$	0.0684	0.1042
(2) $S/N = 29.5dB$	0.0929	0.1081
(3) $S/N = 23.5dB$	0.1999	0.1226

Table 5: Comparison of prediction NRMSE values among Different Models in case of noisy time series, $S/N = 29.5dB$.

Model Name/Signal	Sunspot	Lake	Chaos
Linear predictor(FIR)	0.2343	0.1339	0.4396
MLNN predictor	0.2296	0.0950	0.0944
Proposed model	0.1864	0.0970	0.0930

first two cases show that the $NRMSE$ of noisy time series is smaller than the measure reference R. This means the performance of the hybrid predictor does not affected by these two noise levels, however, by increasing the noise level in case (3), this will affect the performance.

Figure 5 shows the input potential distribution of two hidden neurons of NSP in the case of using noise-free and noisy chaotic data. Figure 5 (a) and (b) is the input potential distribution of the first hidden neuron using noise-free and noisy data respectively. It is obvious that in case of using noisy time series Figure 5 (b), the input potential has been expanded (compared with Figure 5 (a)) mainly to the left saturation region of the sigmoid function used

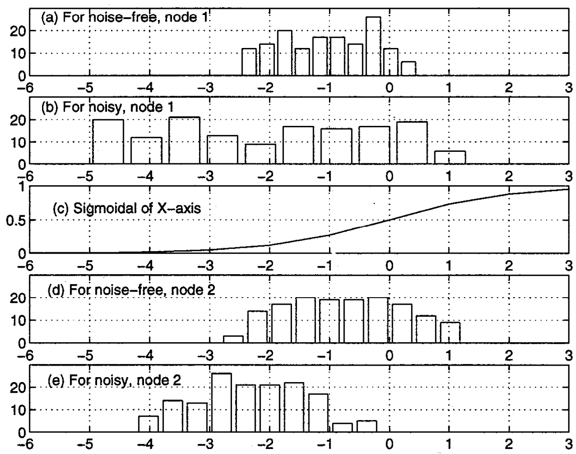


Fig. 5. Histogram of the input potential to two hidden nodes of the NSP for both cases; noisy signal and noise-free one. Input potentials of hidden node 1 are represented by (a), and (b) for noise-free and noisy data respectively.(d), and (e) for hidden node 2. (c) is the sigmoid function plot.

to calculate the neuron output. This make the hidden neuron output does not strongly affected by the noisy data. Other phenomena appears in the second hidden neuron, that is, its input potential distribution in case of noisy data has been shifted to the near of saturation region of the sigmoid function in Figure 5 (e) compared with that distribution in case of noise-free data depicted in Figure 5 (d). Sigmoid function plot is depicted in Figure 5 (c).

To show the performance of the hybrid predictor for other time series, Table 5 shows $NRMSE$ for the above three time series, Sunspot, Lake, and Chaos data. For example, at $S/N = 29.5dB$ and in csae of using Sunspot and Chaos data, the proposed hybrid predictor has a smaller $NRMSE$ than both FIR and MLNN predictors. However, in case of Lake data, FIR has a slightly better performance than MLNN, but the proposed model still the best.

By computing β, the ratio of NSP to LSP output powers Eq.(11). This ratio is calculated for the three time series. Its values are $177.1, 35.8, 196$ for Sunspot, Lake, and Chaos data respectively. In the case of Lake data, we notice that NSP performance is not so good and LSP has a larger contribution than the case of Sunspot and Chaos data. This means that the linear property is dominant in Lake data. Ofcourse, this depends on the time series characteristics.

6 Conclusions

A mechanism analysis of the nonlinear hybrid predictor connecting the multi-layer neural network (NSP) and the FIR filter (LSP) in a cascade form has been demonstrated. The hybrid predictor has demonstrated its efficiency through computer simulations of different kinds of noise-free and noisy time series. Its performance in the noisy data environment does not affected by the low power white noise.

References

[1] A. S. Weigend and D. E. Rumelhart, *Generalization through Minimal Networks with Application to Forecasting*, in Proceeding of INTERFACE'91 : Computing Science and Statistics, edited by E.Keramindas, Springer Verlag, pp. 362-370, 1992.

[2] S.Haykin and L. Li, *Nonlinear Adaptive Prediction of Nonstationary Signals*, IEEE Trans. Signal Processing, vol. 43, No. 2, pp. 526–535, 1995.

[3] W.-T. Lee and J.Pearrson, *A Hybrid Linear/Nonlinear Approach to Channel Equalization Problems*, Advances in Neural Information Processing Systems 5, Editted by S. J. Hanson, J. D. Cowan, and C. Lee Giles, U.S.A, 1993.

[4] S.Haykin and L. Li, *16 kb/s adaptive differential pulse code modulation of speech*, in Proceeding Int. Workshop Applications Neural Networks Telecommun. (Princeton, NJ), pp. 132–138, 1993.

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

[6] A.A.M.Khalaf, 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, 1998.