A Neural-FIR Predictor: Minimum Size Estimation Based on Nonlinearity Analysis of Input Sequence

Ashraf A.M.Khalaf[1] , Kenji Nakayama[2] and Kazuyuki Hara[3]

[1] Graduate School of Nat. Sci. and Tech., Kanazawa Univ.
[2] Faculty of Engineering , Kanazawa Univ.
2-40-20 , Kodatsuno , Kanazawa , 920 JAPAN
E-mail : nakayama@t.kanazawa-u.ac.jp
[3] Gunma Polytechnic College

Abstract. In this paper, a hybrid model of multy-layer neural network combined with a finite-impulse-response filter is proposed for a nonlinear time series prediction. We introduce an important analysis of the input sequence to determine the effective minimum combination of the input samples and hidden neurons. Through computer simulations, using both sunspot and computer generated time series, the proposed analysis has shown its effectiveness and the proposed predictor has demonstrated its superiority. It is of a faster convergence and smaller residual error than the conventional nonlinear predictor.

1 Introduction

It is well known that linear filters are still insufficient to deal with a nonlinear prediction task. So, a number of neural network structures have been proposed for this purpose due to their nonlinear properties built into their structures [1], [2], [3], [4], and others. In [1], a multy-layer neural network was used to predict the sunspot time series and its results were encouragable. The network input dimension was used to estimate the minimum network size as discussed in [5]. Although, the network size was optimized, the convergence time was very long. In [2], the convergence speed is appreciated. But, it was on the expense of the network size. It was a large size, complex numbered network and with local feedback in its hidden neurons and its computer simulation made only for a discrete amplitude signal.

Here we propose a cascade structure of two subsections :
(1) A nonlinear subsection (NSS), which consists of a multi-layer (ML) neural network with a single hidden layer.
(2) A linear subsection (LSS), which is a finite-impulse-response (FIR) filter.

The cascade structure was first proposed in [3] using a pipelined recurrent neural network for speech signal prediction trained by real-time recurrent learning algorithm.

2 A Cascade Structure of Neural Network-FIR Predictor

2.1 Network Structure

Figure 1 shows the proposed predictor structure. The NSS performs a nonlinear mapping from the input space into an intermediate space to extract the nonlinear features of the input sequence. The LSS performs a linear mapping or compensation from the intermediate space to the output space. In the intermediate space produced by the NSS a degree of the signal nonlinearity is reduced compared with the original input signal.

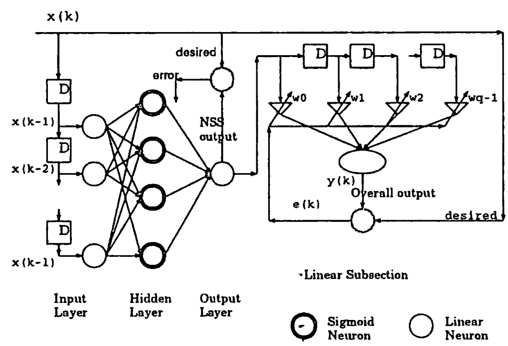

Fig. 1. Structure of the proposed predictor

2.2 Network Operation and Learning Algorithms

The past l samples of the input signal , x(k-1),x(k-2),...x(k-l) are applied to the NSS and the current sample, x(k) is used as the desired response for both the NSS and the LSS. The LSS is a (q-1)th-order FIR filter. The weights of both subsections are simultaneously adjusted. The NSS trained by the real-numbered Back-Propagation algorithm, and the LSS is trained by the normalized LMS algorithm.

3 Input Sequence Analysis

We introduce a theoretical analysis for the input sequence to obtain the effective minimum combination of input samples and hidden neurons which enables the network to achieve its convergence faster than the other networks of the same network size.

3.1 Network Convergence Difficulties

Let us report the difficulties affecting the convergence property of the network. Here we consider two cases :

Case 1: Impossible Mapping

Let the following mapping expression to be:

$$X_k \Rightarrow x(k), \qquad k = 1, 2, ..., i, ..., N \tag{1}$$

where $X_k = (x(k-1), x(k-2), ..., x(k-l))$, represents the kth pattern which will be mapped onto $x(k)$, l is the number of the samples in the input, and N is the total number of patterns in one epoch. Again let:

$$X_i \Rightarrow x(i) \tag{2}$$

If the above two different mappings given by Eqs (1) and (2) satisfy the following relation:

$$X_k = X_i, \qquad x(k) \neq x(i). \tag{3}$$

Then, these two mappings are impossible, and if such mappings are exist, the network will fail to converge at all.

Case 2 : Difficult Mapping

There is another case we will call it a difficult mapping. It can be expressed as:

$$X_k \approx X_i, \qquad x(k) \neq x(i) \tag{4}$$

In this case the two patterns are similar to each other. Such difficult mappings make the learning process so hard to converge. Although the convergence may be possible, but it may often take a very long time.
Let us express the mapping similarity condition as:

$$d = \mid X_k - X_i \mid \leq I \tag{5}$$

where I is a threshold value . Its value will be determined by experience as :

$$0 < I \leq Ag \tag{6}$$

where Ag is the average value of the input time series .

Here, we consider I to take two values $I1$, $I2$, where ($I1 < I2$). Thus the following two degrees of similarity are discussed :

(1) High degree of similarity: X_k differs from X_i by $I1$ and $x(k) \neq x(i)$. We call it as the more difficult mapping case.

(2) Slightly high degree of similarity: X_k differs from X_i, by $I2$. We will call it as a less difficult mapping case. It may represent a true difficulty in some cases depending on the input sequence itself . For long length time series, it is necessary to take a larger threshold values I into account.

3.2 Estimation of the Input Dimension

To determine the effective input dimension based on the above analysis, we can say : The effective number of input samples is that number at which the network does not see the similar mapping problem. To do so, we compute the variance of the output of similar mappings. The number corresponds to the minimum average value of this variance will represent the estimated number of input dimension.
Let us express the following equations :

$$\mu = 1/G * \sum_{g=1}^{G} x(g), \tag{7}$$

$$\overline{\sigma^2} = 1/T[1/G * \sum_{g=1}^{G} (x(g) - \mu)^2], \tag{8}$$

where G is the total number of similar mappings, and $x(g)$ is the sample to be predicted of gth similar mapping. μ is the mean and $\overline{\sigma^2}$ is the average value of the variance of the output of similar mappings respectively.

Table 1, Table 2, and Table 3 demonstrate the calculations of the average values of the variance of the output of similar mappings for the computer simulation examples (See Sec. 4.1). From these tables we can determine the minimum input samples. The effective number of hidden neurons is determined by trial. Therefore, we can obtain the minimum effective combination of the input samples and the hidden neurons.

4 Computer Simulation

4.1 Nonlinear Time Sequences

Computer simulations have been done for both discrete and continous amplitude signals. First, for a discrete amplitude signal used in [2], ($x(k) = (\sum_{n=1}^{M} x(k-n)) \bmod N$). We have taken two examples (easy and difficult examples) at $N = 3, M = 5, and N = 7, M = 3$ respectively. Network input dimension has been determined based on the above analysis for both easy and difficult examples as 3 and 6 respectively (See Table 1, and Table 2). Then the NSS size for both examples are determined as (3-2-1) and (6-7-1) respectively. Also we have used the yearly sunspot data from 1700 through 1920 for training as it has been used in [1]. Our network input dimension is determined based on our analysis (see Table 3). It is the same as that determined in [5] and used by [1].

4.2 Simulation Results

Figures 2, and 3 show that the proposed predictor has achieved comparable results to those obtained in [2], and the networks size have an advantage of a very small size over their ones used in [2].

Figure 4 shows that using the LSS as a second stage after NSS with the optimum size speeds up the convergence and decreases the residual error not only compared with a conventional ML neural network [1] but also for the ML network whith direct connections between each input unit and its output [4].

Table 1. Easy Example Analysis, $I \leq Ag$

No.of Input Samples		2	3
$d \leq I$	σ^2	0.5474	0

Table 2. Difficult Example Analysis, $I1 \leq 50\%Ag$, and $I2 \leq 80\%Ag$

No.of Input Samples		2	3	4	5	6
$d \leq I1$	σ^2	2.9321	0.5127	0	0	0
$d \leq I2$	σ^2	3.4569	1.8187	1.1399	0.0312	0

Table 3. Sunspot Example Analysis, $I1 \leq 50\%Ag$, and $I2 \leq 80\%Ag$

No.of Input Samples		2	3	4	5	6	8	9	10	12
$d \leq I1$	σ^2	0.0108	0.0070	0.0050	0.0030	0.00182	0.00002	0	0	0
$d \leq I2$	σ^2	0.0164	0.0110	0.0084	0.0062	0.00452	0.0014	0.00003	0.00001	0

5 Conclusion

A cascade structure of ML a neural network and an FIR filter has been proposed for nonlinear time series prediction. Some important analysis for the input sequence properties and their relation to the network size and the convergence speed has been investigated. Based on this analysis the minimum number of the input samples has been determined. Our proposed predictor and analysis show their effectiveness through the computer simulations. Faster convergence and smaller residual error have been achieved by using the FIR as a second stage after NSS.

References

1. A. S. Weigend and D. E. Rumrlhart : Generalization through Minimal Networks with Application to Forecasting. Proceeding of INTERFACE'91 : Computing Science and Statistics, edited by Elaine Keramindas. Springer Verlag (1992) 362-370

2. Kinouchi, M. and Hagiwara, M.: Learning Temporal Sequences by Complex Neurons with Local Feedback. ICNN **6** (1995) 3165–3169
3. Haykin and L. Li: Nonlinear Adaptive Prediction of Nonstationary Signals. IEEE Trans. Signal Processing, **43** (1995) 526–535
4. A. S. Weigend and N. A. Gershenfeld, Time Series Prediction: Forecasting the Future and Understanding the Past. Proceedings V. XV, Santa Fe Institute (1994)
5. Tong, Howell and Lim, K. S. : Threshold Autoregression, Limit Cycles and Cyclical Data. Journal Royal Statistical Society B **42**, (1980) 245-292
6. Ashraf A.M.Khalaf, and K.Nakayama : A Hybrid Neural Predictor and Its Convergence Analysis. Proceeding of the 10th Karuizawa Workshop on Circuits and Systems, Japan (1997) 357-362

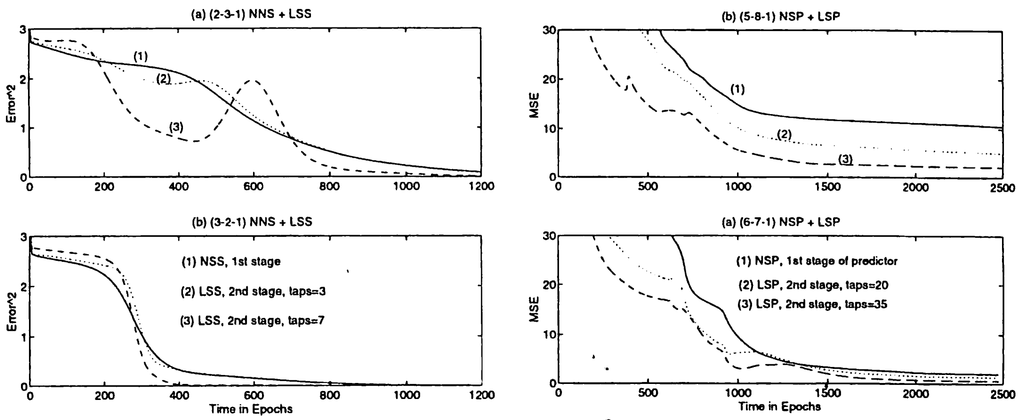

Fig. 2. Easy Example : (a) Few input samples and enough hidden neurons (b) Minimum effective combination of input samples and hidden neurons.

Fig. 3. Difficult Example : (a) Minimum effective combination of input samples and hidden neurons (b) A network of the same but not proper size.

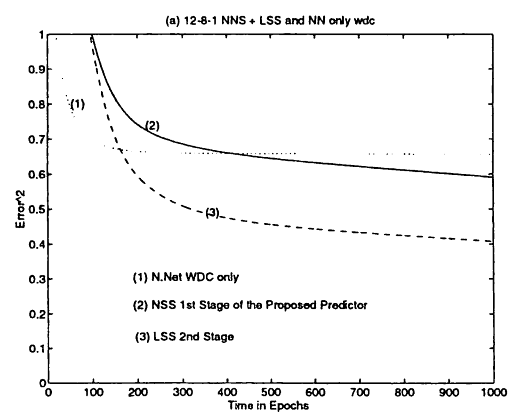

Fig. 4. Sunspot Example : WDC means using only of the ML neural net. with linear direct connections between its inputs and the output.