第10回 回路とシステム
軽井沢ワークショップ
The 10th Karuizawa Workshop on
Circuits and Systems, April 21-22, 1997

A Hybrid Neural Predictor and Its Convergence Analysis

Ashraf A.M.Khalaf† Kenji Nakayama‡
† Graduate School of Nat. Sci. and Tech. ,
‡ Faculty of Eng. , Kanazawa Univ.
2-40-20 , Kodatsuno , Kanazawa , 920 JAPAN
E-mail : nakayama@t.kanazawa-u.ac.jp

1 Abstract

Neural Networks are useful for nonlinear signal processing due to their nonlinear properties. In this paper , some important properties of the input sequence and their relation to the network convergence speed are investigated with the aim of obtaining the effective minimum combination of input samples and hidden neurons . The effective input dimension is determined based on the proposed analysis . After that the effective minimum number of hidden neurons is determined using try-and-error criterion . A cascade structure of multi-layer (ML) neural network combined with a finite-impulse-response (FIR) filter is suggested for one-step prediction of nonlinear time series. Using the FIR as a second stage of the predictor speeds up the convergence of the proposed network for different input samples and hidden neurons combinations.

2 Introduction

Neural Networks are used in many applications of signal processing, specially for nonlinear signals due to their nonlinear properties built into their structure [1] , [2] , [3] , [4] , [5] . A number of networks have been proposed for the nonlinear temporal processing . However , their performance is still insufficient [6] . The convergence speed obtained in [6] is appreciated . But , it is on the expense of the network size (a large size , complex numbered network and with local feedback in its hidden neurons) .

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 [7] using a pipelined recurrent neural network for speech signal prediction trained by real-time recurrent learning algorithm .

3 A Hybrid Neural Predictor

3.1 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 , and the LSS performs a linear mapping (or compensation) from the intermediate space (which will have a less degree of nonlinearity than the original input signal) produced by NSS to the output space .

3.2 Network Operation and Learning Algorithms

As shown in Fig. 1 , the input to the neural network is the past n samples (pattern) of the signal , and the desired output is the current sample. The output of NSS and its $(q-1)$ past values ($(q-1)$ is the number of taps of FIR filter) are inputed to the LSS which has the same desired output as the NSS . Then the filter error is computed and its coefficients are adjusted . This occurs pattern by pattern until the end of an epoch . This procedure is repeated until the squared error of either LSS or that of NSS reaches a reasonable value . The NSS is trained by the real-number Back-Propagation algorithm , and the LSS is trained by the normalized LMS algorithm .

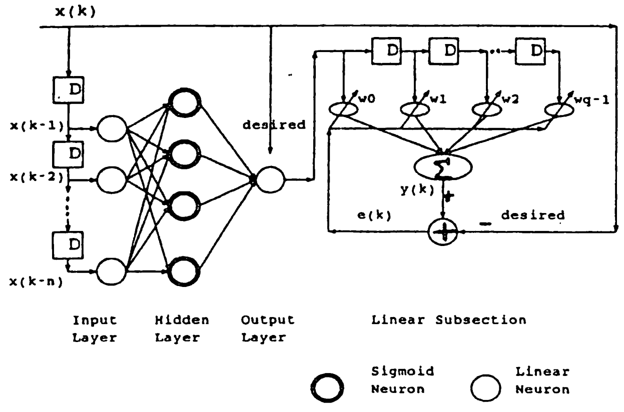

Figure 1: Structure of the proposed predictor

3.3 System Equations of NSS

The output of jth neuron in the nth layer can be expressed by:

$$y_j^{(n)} = f(u_j^{(n)}) \tag{1}$$

$$u_j^{(n)} = \sum_i w_{ji} y_i^{(n-1)} + \theta_j^{(n)} \tag{2}$$

where $u_j^{(n)}$ is the neuron input potential , w_{ji} is the connection weight from the ith neuron in the (n-1)th layer to the jth neuron in the nth layer . $y_i^{(n-1)}$ is the output of ith neuron in the (n-1)th layer , and $\theta_j^{(n)}$ is the bias value at jth neuron of nth layer .The activation function f is a sigmoid function:

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

Note: The output neuron is linear.

4 Input Sequence Analysis

Some important properties of the nonlinear sequence are theoretically analyzed to obtain the effective number of the input samples (input dimension) . The minimum effective number of hidden neurons is determined by try-and-error . Therefore, we can obtain the network of minimum effective combination of input samples and hidden neurons which will be promised to achieve its convergence faster than the other possible combinations except for a network of sufficiently large number of free parameters . In addition , the network size will be reduced to its minimum .
First , let us report the difficulties affecting the convergence property of the network. Here we consider two cases :

4.1 Case 1: Impossible Mapping

Let the following mapping expression to be:

$$X_k \Rightarrow x(k), \qquad k = 1, 2, ..., i, ..., N \tag{4}$$

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

$$X_i \Rightarrow x(i) \tag{5}$$

Table 1: Easy Example Analysis, $I1 = 90\%Ag$

No of Input Samples		2	3	4	5
$d \leq I1$	$\overline{\mu}$	1.0875	1.1250	1.2500	1.1250
	$\overline{\sigma^2}$	0.5474	0	0	0

Table 2: Difficult Example Analysis, $I1 = 50\%Ag$, and $I2 = 80\%Ag$

No of Input Samples		2	3	4	5	6
$d \leq I1$	$\overline{\mu}$	3.2542	3.1944	3.2083	3.2083	3.2083
	$\overline{\sigma^2}$	2.9321	0.5127	0	0	0
$d \leq I2$	$\overline{\mu}$	3.2900	3.1979	3.2066	3.2083	3.2083
	$\overline{\sigma^2}$	3.4569	1.8187	1.1399	0.0312	0

If the above two different mappings given by Eqs (4) and (5) satisfy the following relation:

$$X_k = X_i, \qquad x(k) \neq x(i) \tag{6}$$

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

4.2 Case 2 : Difficult Mapping

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

$$X_k \approx X_i, \qquad x(k) \neq x(i) \tag{7}$$

In this case the two patterns are similar (in some degree) to each other , and increasing in such difficult mappings through the N mappings will increase the difficulties for the network to converge . Although the convergence may be possible , but it may often take a very long time . How can we measure this similarity ? and To what degree this similarity will be allowed for the network to achieve convergence ?

To answer the above questions let us express the mapping similarity condition as:

$$d = \mid X_k - X_i \mid \leq I \tag{8}$$

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

$$0 < I \leq Ag \tag{9}$$

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, the case in which X_k differs from X_i, by $I1$ and $x(k) \neq x(i)$. and we call it as the most difficult mapping case.

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

Note: If $X_k = X_i$ and $x(k) \neq x(i)$ (i.e. $I = 0$) this will be the case called impossible mapping (sec.4.1).

5 Computer Simulation

5.1 Nonlinear Sequence Generation

We use the same nonlinear function used in [1] ,which of the form

$$x(k) = (\sum_{n=1}^{M} x(k-n)) \bmod N. \tag{10}$$

and generate two examples (easy and difficult examples) at $N = 3, M = 5,$ *and* $N = 7, M = 3$ respectively.

Easy Example : $N = 3,\ M = 5$:

$$\ldots, \underbrace{2,0,2,1,1,0,1,2}_{(One-Epoch)}, 2, 0, 2, \ldots$$

Difficult Example : $N = 7,\ M = 3$:

$$\ldots, \underbrace{3,0,4,0,4,1,5,3,2,3,1,6,3,3,5,4,5,0,2,0,2,4,6,5,1,5,4,3,5,5,6,2,6,0,1,0,1,2,3,6,4,6,2,5,6,6,3,1}_{(One-Epoch)}, 3, 0, 4, 0, \ldots$$

5.2 Analysis of Nonlinear Sequences

To determine the effective input dimension of the NSS 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 (at some degree of similarity). To do so , we compute the variance of the output (sample to be predicted) of similar mappings , and take the number of input samples which corresponds to the minimum average value of this variance as the required effective number of input dimension .
Table 1 and Table 2 demonstrate the calculations of the average values of the mean and variance of the output of similar mappings for the easy and difficult example respectively .

By trial the effective number of the hidden neurons is determined. Therefore , we can obtain the minimum effective combination of the input samples and hidden neurons . As it is known , using unrestricted number of free parameters leads to accurate and faster convergence and the LSS may be unimportant. But , in the case of using other combinations of the input samples and hidden neurons the use of FIR filter with sufficiently large number of taps as a second stage of the predictor may result in a faster convergence than the using only NSS .

From Table 1 and 2 , the minimum number of the input samples for both examples has been determined , and by trial we have determined the minimum number of hidden neurons . Therefore , the minimum combination of input samples and hidden neurons for both examples are determined as (3-2) , and (6-7), respectively.

5.3 Convergence Properties

A computer simulation for one-step prediction task for easy and difficult examples have been done . We tried to obtain a compromise between the network size and the network convergence speed. Using the above analysis (sec., 4.2) the effect of changing the input samples and the hidden neurons on the convergence speed has been investigated and demonstrated in the simulation results Fig. 2 , and Fig. 3 . Here the convergence is considered to be achieved when the squared error of either the LSS or NSS reaches 0.01

Simulation results show that a minimum size ML neural network combined with an FIR filter have been achieved its goal with a comparable results to those proposed in[6] , and have an advantage of a very small size over them. Increasing the number of input samples has a larger effect in both NSS and LSS convergence speed than increasing the hidden neurons .

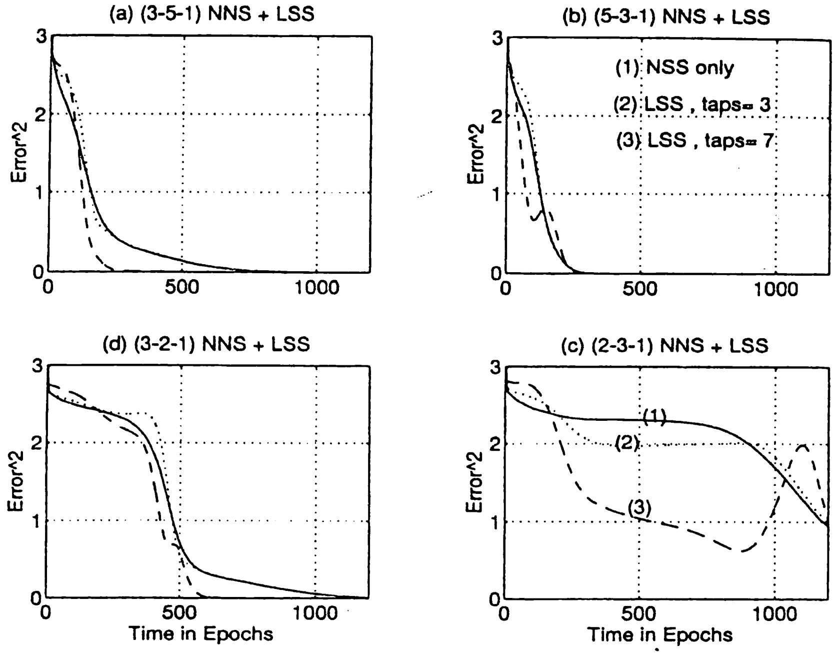

Figure 2: Easy Problem : (a)- Minimum input samples and large hidden neurons (b)- Large input samples and enough hidden neurons (c)- Few input samples and enough hidden neurons (d)- Minimum effective combination of both .

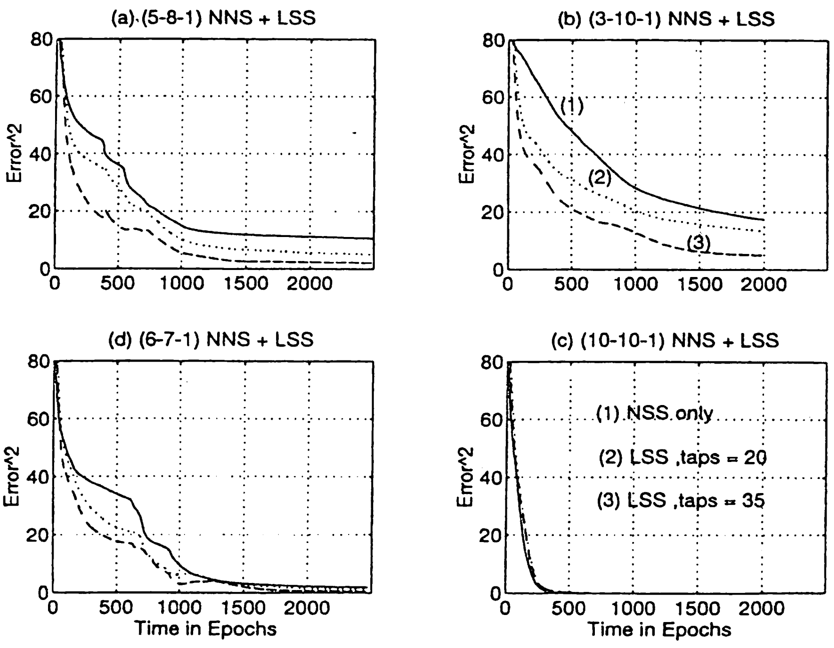

Figure 3: Difficult Problem : (a) , (b) , and (c) are different combinations of input samples and hidden neurons , (d)- Minimum effective combination of both.

6 Conclusion

A hybrid neural predictor has been proposed for nonlinear time series prediction . Some important analyses for the input sequence properties and their relation to the convergence speed have been investigated .Based on this analysis the minimum number of the input samples has been determined . After this , the minimum combination of the input samples and the hidden neurons required for fast convergence has been determined .
Faster convergence has been achieved by using the LSS (FIR filter with a sufficiently large number of taps) as a second stage after NSS . Although the discrete amplitude sequences have been used in the simulation ,the proposed predictor may be applied to the continuous one due to the LSS .

Now we are doing a simulation for the sunspot time series prediction using the proposed structure and its results seem to be promised compared with that obtained in [1] .

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.
[2] A. S. Weigend and N. A. Gershenfeld, Time Series Prediction: Forecasting the Future and Understanding the Past. Proceedings V. XV, Santa Fe Institute, 1994.
[3] Rumelhart, David E., Hinton , Geoffrey E., and Wiliams , Ronald J. : Learning internal representation by error propagation. In Rumelhart, D. E. and McClelland, J. L., editors, Parallel Distributed Processing, Pages 318-362. MIT PRESS, 1986.
[4] S. Haykin, Neural Networks: A Comperhensive Foundation. New York: Macmillan, 1994.
[5] Hertz, J., Krogh, A. and Palmer, R. G.: Introduction to the theory of Neural Computation, Santa Fe Institute, 1991.
[6] Kinouchi, M. and Hagiwara, M.: " Learning Temporal Sequences by Complex Neurons with Local Feedback ", ICNN Vol. 6, pp. 3165-3169 (1995).
[7] S. Haykin and L. Li: " Nonlinear Adaptive Prediction of Nonstationary Signals ", IEEE, Trans. Signal Processing, Vol. 43, No. 2., Feb., 1995.