Nonlinear Adaptive Filter Based on Pipelined Bilinear Function Link Neural Networks Architecture

Dinh Cong Le(✉), Van Minh Le, Thai Son Dang, The Anh Mai, and Manh Cuong Nguyen

School of Engineering and Technology, Vinh University, Vinh, Vietnam
ldcong@vinhuni.edu.vn

Abstract. In order to further enhance the computational efficiency and application scope of the bilinear functional links neural networks (BFLNN) filter, a pipelined BFLNN (PBFLNN) filter has been developed in this paper. The idea of the method is to divide the complex BFLNN structure into multiple simple BFLNN modules (with a smaller memory-length) and cascade connection in a pipelined fashion. Thanks to the simultaneous processing and the nested nonlinearity of the modules, the PBFLNN achieves a significant improvement in computation without degrading its performance. The simulation results have demonstrated the effectiveness of the proposed method and the potentials of the PBFLNN filter in many different applications.

Keywords: Pipelined · Generalized FLNN · Nonlinear adaptive filtering

1 Introduction

Many practical systems (such as system identification, signal prediction, channel equalization, and echo and noise cancelation,…) may contain nonlinearity. The linear adaptation technique cannot model well enough because of the nonlinear nature of these systems [1]. To overcome this problem, many new classes of nonlinear filters are based on neural networks (NNs) and truncated Volterra series (VFs) has been developed [1, 2]. However, they also reveal many disadvantages such as complex architecture and heavy computing burden in their implementation [3, 4].

It is well known that the FLNN has been proposed to replace the multilayer artificial neural network (MLANN) in some simple nonlinear applications because it has a single layer structure, low computational complexity, and the simple learning rule [4]. It has been successfully applied in other areas of nonlinear filtering including nonlinear dynamic systems identification, channel equalization, active noise control, nonlinear acoustic echo cancellation [4–8]. However, the performance of the FLNN-based model may be significantly impaired when faced with systems containing strong nonlinear distortion. As pointed out in [9], the main reason may be that the basic functions of FLNN lack the cross-terms (for example $x(n) * x(n-1)$, $x(n-1) * x(n-2), \ldots$. To mitigate this disadvantage, some studies have added appropriate cross-terms into the conventional FLNN structure [9, 10]. Research results

D.-T. Tran et al. (Eds.): ICISN 2021, LNNS 243, pp. 256–264, 2021.
https://doi.org/10.1007/978-981-16-2094-2_32

in [9, 10] indicate that these new models outperform the Volterra-based model in the noise control application.

On the other hand, in order to increase the computational efficiency for recurrent neural networks (RNN), a pipelined RNN (RNN) structure was developed in a speech predictor [11]. The benefits of pipelined architecture make the nonlinear predictor significantly reducing the total computational of recurrent neural networks (RNN). Following this study, many computational efficiency systems using pipeline architecture have been developed and successfully applied for speech signal prediction [12], channel equalization [13].

In order to be able to model certain nonlinear systems well enough, the BFLNN filter needs to be designed with a sufficiently large memory length. However, this also leads to the computational complexity of BFLNN becoming quite heavy. Inspired by efficient pipelined architecture, a pipelined BFLNN (PBFLNN) filter is proposed in this paper. In this method, a complex BFLNN structure (contains many cross-terms) is divided into several simple small-scale BFLNN modules (smaller memory-length, contains less than cross-terms) and cascaded in a pipelined parallel fashion. Thanks to the parallel processing of small-scale modules, its total computational efficiency is significantly improved.

2 Nonlinear Adaptive PBFLNN Filter

The PBFLNN Structure: The proposed PBFLNN structure consists of simple small-scale BFLNN modules connected in a pipelined fashion. In addition, to ensure the overall output is a global estimate, the outputs of each module are filtered through a conventional transversal filter. The design of the PBFLNN is illustrated in Fig. 1.

The simple BFLNN modules are identical in design (i.e. the parameters selected for the external signal input and the cross-terms are the same). Thus, the modules have the same synaptic weight matrix. As designed, it is easy to see that the input to each module consists of two types of signals: one of them is the external signal and the other is the output signal of the previous module.

Suppose we define N external input signals of the ith module as

$$X_{Ei}(\xi) = [x(\xi - i), x(\xi - i - 1), \ldots, x(\xi - i - N + 1)]^T \tag{1}$$

Hence, the input signal of the ith module is

$$X_i(\xi) = \left[X_{Ei}^T(\xi), U_i(\xi)\right]^T = [x(\xi - i), x(\xi - i - 1), \ldots, x(\xi - i - N + 1), U_i(\xi)]^T, \qquad i = 1, \ldots, M \tag{2}$$

where $U_i(\xi) = y_{i+1}(\xi)$ when the module differs from the Mth module; $U_i(\xi) = y_M(\xi - 1)$, when the module is the Mth module.

Since each module is a BFLNN structure, the input signal is expanded to

$$Xf_i(\xi) = \left[Xf1_i^T(\xi),\ Xf2_i^T(\xi), \cdots,\ XfV_i^T(\xi)\right]^T \tag{3}$$

where $Vi = (6 + 4k)$ is the number of channels of the ith module, and k is the cross-term selection parameter.

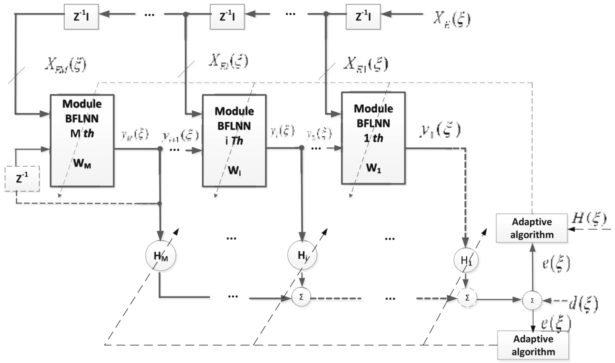

Fig. 1. The proposed nonlinear adaptive PBFLNN filter

Based on the ideal of the BFLNN filter (refer to [10]), the signals for each channel $(Xf1_i(\xi), Xf2_i(\xi), \ldots, XfV_i(\xi))$ of the ith module can be expressed as follows

$$Xf1_i(\xi) = [x(\xi - i), x(\xi - i - 1), \ldots x(\xi - i - N + 1), U_i(\xi)]^T \tag{4}$$

$$Xf2_i(\xi) = [cos(\pi x(\xi - i)), \ldots cos(\pi x(\xi - i - N + 1)), cos(\pi U_i(\xi))]^T \tag{5}$$

$$Xf3_i(\xi) = [sin(\pi x(\xi - i)), \ldots sin(\pi x(\xi - i - N + 1)), sin(\pi U_i(\xi))]^T \tag{6}$$

$$Xf4_i(\xi) = \left[y(\xi - i - 1), y(\xi - i - 2), \ldots y\left(\xi - i - N_{py}\right)\right]^T \tag{7}$$

$$\begin{aligned} Xf5_i(n) = [&y(\xi - i - 1)cos(\pi x(\xi - i)), y(\xi - i - 2)cos(\pi x(\xi - i - \\ &1)), \ldots, y(\xi - i - k)cos(\pi x(\xi - k + 1))]^T \end{aligned} \tag{8}$$

$$\begin{aligned} Xf6_i(\xi) = [&y(\xi - i - 1)sin(\pi x(\xi - i)), y(\xi - i - 2)sin(\pi x(\xi - i - \\ &1)), \ldots, y(\xi - i - k)sin(\pi x(\xi - k + 1))]^T \end{aligned} \tag{9}$$

$$\vdots \qquad = \qquad \vdots$$

$$\begin{aligned} Xf(2k+5)_i(n) = [&x(\xi - i - 1)cos(\pi x(\xi - i)), \ldots, x(\xi - i - k)cos(\pi x(\xi - \\ &i - k + 1)), U_i(\xi)\cos(\pi x(\xi - i - k))]^T \end{aligned} \tag{10}$$

$$\begin{aligned} Xf(2k+6)_i(\xi) = [&x(\xi - i - 1)sin(\pi x(\xi - i)), \ldots, x(\xi - i - k)sin(\pi x(\xi - \\ &i - k + 1)), U_i(\xi)sin(\pi x(\xi - i - k))]^T \end{aligned} \tag{11}$$

$$\begin{array}{ccc} \vdots & = & \vdots \\ & XfV-1_i(\xi) = [U_i(\xi)cos(\pi x(\xi-i))]^T & \end{array} \tag{12}$$

$$XfV_i(\xi) = [U_i(\xi)sin(\pi x(\xi-i))]^T \tag{13}$$

As analyzed above, the synaptic weight vector of each module is designed similarly. We, therefore, define the weight vectors for all modules as

$$W(\xi) = \left[w_1(\xi), w_2(\xi), ..., w_{Lf}(\xi)\right]^T \tag{14}$$

where the length L_f of expanded input signal $Xf_i(\xi)$ is defined by $L_f = 3(N+1) + N_{py} + 2(k+1)(k+2)$. and N_{py} is the feedback coefficient selection parameter.

Therefore, the output of the ith modules is a linear combination of the weights and the extended signals of the ith module as follows

$$y_i(\xi) = W^T(\xi)Xf_i(\xi) \tag{15}$$

The outputs of each module are then filtered through an adaptive finite impulse response (FIR) filter to obtain a global estimate. It is easy to see that the output of this FIR filter is also the output of the PBFLNN nonlinear filter and is defined as follows.

$$\hat{y}(\xi) = H^T(\xi)Y(\xi) \tag{16}$$

where $H(\xi) = [h_1(\xi), h_2(\xi), ..., h_M(\xi)]^T$ is the weight vector of the FIR filter, and $Y(\xi) = [y_1(\xi), y_2(\xi), \ldots, y_M(\xi)]^T$ is the input vector of the FIR filter.

Adaptive Algorithm: In this section, the weight vectors of the modules $W(\xi)$ and the FIR filter $H(\xi)$ are updated to minimize $J(\xi)$ (instantaneous square error). In this way, we define the cost function as follows

$$J(\xi) = e^2(\xi) \tag{17}$$

where the $e(\xi) = d(\xi) - \hat{y}(\xi) = d(\xi) - H^T(\xi)Y(\xi)$ is the instantaneous output error at time ξ.

Thus, weight vectors of the modules $W(\xi)$ and the FIR filter $H(\xi)$ are updated in accordance with the rule as follows

$$H(\xi+1) = H(\xi) - \frac{1}{2}\mu\nabla_{H(\xi)}J(\xi) \tag{18}$$

$$W(\xi+1) = W(\xi) - \frac{1}{2}\eta\nabla_{W(\xi)}J(\xi) \tag{19}$$

where $\nabla_{W(\xi)}J(\xi)$ and $\nabla_{H(\xi)}J(\xi)$ are the gradient of cost function $J(\xi)$ with respect to the $W(\xi)$ and the $H_i(\xi)$, respectively; μ and η are the learning rate. The gradients can be calculated as follows

$$\nabla_{W(\xi)}J(\xi) = \frac{\partial J(\xi)}{\partial W(\xi)} = 2e(\xi)H^T(\xi)\begin{bmatrix} \frac{\partial(y_1(\xi))}{\partial W(\xi)} \\ \vdots \\ \frac{\partial(y_M(\xi))}{\partial W(\xi)} \end{bmatrix} = -2e(\xi)\sum\nolimits_{i=1}^{M} h_i(\xi)Xf_i(\xi) \quad (20)$$

$$\nabla_{H(\xi)}J(\xi) = \frac{\partial J(\xi)}{\partial H(\xi)} = 2e(\xi)\frac{\partial d(\xi) - H^T(\xi)Y(\xi)}{\partial H(\xi)} = -2e(\xi)Y(\xi) \quad (21)$$

Substituting (21) in (18) we yield the update equation of the weigh vector $H(\xi)$ as

$$H(\xi+1) = H(\xi) + \eta e(\xi)Y(\xi) \quad (22)$$

Similarly, substituting (20) in (19) we obtain the update equation of the weigh vector $W(\xi)$ as

$$W(\xi+1) = W(\xi) + \mu e(\xi)\sum\nolimits_{i=1}^{M} h_i(\xi)Xf_i(\xi) \quad (23)$$

Table 1. Computational complexity of FLNN, BFLNN and PBFLNN filters

Type of filter	Multiplications	Additions
FLNN	$2(2B + 1)N_f + 1$	$2(2B + 1)N_f 1$
BFLNN	$2[3N_b + N_y + N_{db}(N_{db} + 1)] + 2N_{db} + 2$	$2[3N_b + N_y + N_{db}(N_{db} + 1)] - 1$
PBFLNN	$2[3(N + 1) + N_{py} + 2(k + 1)(k + 2)] + 2k + 2M + 4$	$2[3(N + 1) + N_{py} + 2(k + 1)(k + 2)] + 2M$

Computational Complexity: To evaluate the effectiveness of the proposed method, a comparison of the computational complexity of the 3 filters (FLNN, BFLNN and proposed PBFLNN) is summarized in Table 1. Assuming N_f, N_b, and N, is external input signals of FLNN, BFLNN and proposed PBFLNN, respectively. N_{db} and N_y, are the cross-term and feedback selection parameter of BFLNN; M is the number of modules.

3 Simulation

To demonstrate the effectiveness of the proposed method, several experiments were conducted to compare the PBFLNN filter with the FLNN and BFLNN filters in terms of performance and computational complexity. The parameters of the FLNN, BFLNN filters and the PBFLNN were selected the same for all experiments. Specifically, the parameter of FLNN is $Nf = 10$; $B = 3$, that of BFLNN is $N_b = 10$; $N_y = 10$; $N_{db} = 9$ and that of PBFLNN is $N = 4$; $N_{py} = 4$; $k = 3$. The function expansion is first-order type for PBFLNN and BFLNN. The experimental results are all taken by averages on 100 independent runs.

Experiment 1: In this experiment, we conducted the identification of a nonlinear dynamic model as described below

$$d(n) = \frac{d(n-1)}{1 + d^2(n-1)} + x^2(n)x(n-1) \tag{24}$$

where $d(n)$ and $x(n)$ are the observed signal and the input signal of the system. The performance of the adaptive filters is evaluated based on the mean square error $MSE = 10\log 10(e^2(n))$. Assuming that the $x(n)$ is the random sequence, and its range is chosen as (0,1).

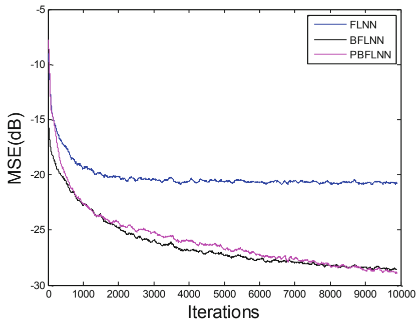

Fig. 2. Comparison of MSE for random input signal.

The step-size of the synaptic weigh vector $W(n)$ (the expanded by BFLNN function) includes linear part $\mu_1 = 0.03$; sin (.) cos (.) part $\mu_2 = 0.05$; feedback part $\mu_3 = 0.03$ and cross-terms part $\mu_4 = 0.04$. The step-size of the FIR filter of the PBFLNN is $\eta = 0.04$. Figure 2 shows the averaged MSE performance curves for the random input signal. It is clear that the performance of the proposed PBFLNN filter is equivalent to that of BFLNN.

In addition, the computational requirements of the filters are summarized in Table 2. It is obvious that the computational requirements of the PBFLNN are about 51% less than that of the BFLNN.

Table. 2 Computational complexity of FLNN, BFLNN and PBFLNN filters

Type of filter	Parameter	Multiplications	Additions
FLNN	$(N_f = 10,\ B = 3)$	141	139
BFLNN	$(B = 1,\ N_{db} = 9,\ N_b = 10,\ N_y = 10)$	280	259
PBFLNN	$N = 4,\ k = 3;\ N_{py} = 4;\ B = 1,\ M = 5$	138	128

Experiment 2: In this experiment, we carried out to compare identification of a nonlinear dynamic system as described in [15].

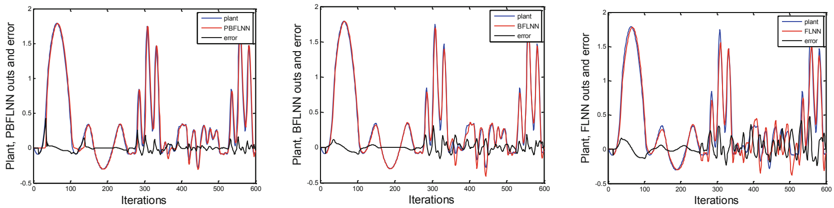

Fig. 3. The identification results of the nonlinear dynamic system are based on the PBFLNN, BFLNN, and FLNN filters

The step-size of the PBFLNN filter are set to $\eta = 0.87$, $\mu_1 = 0.79$, $\mu_2 = 0.68$, and $\mu_3 = 0.83$. The Fig. 3 show the identification results with the corresponding BFLNN, FLNN and PBFLNN filters. It is clear that the PBFLNN achieves equivalent performance to the BFLNN with a lower computational complexity.

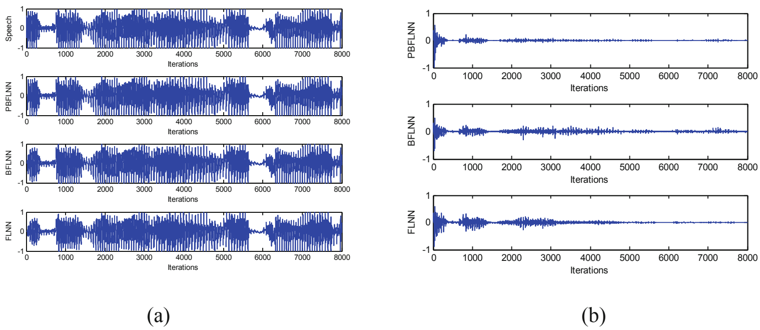

Fig. 4. (a) Original signal and the corresponding prediction signals, (b) corresponding prediction error

Experiment 3: To demonstrate speech signal predictive performance of the proposed PBFLNN, we use the experiment as described in [14]. The one-step forward prediction is employed to measure the predicting capability and defined as in [14].

Figure 4a illustrates the results of speech prediction using the PBFLNN, BFLNN, FLNN filters respectively, and the original speech signal. Figure 4b depicts the correspoding predicting errors. The one-step prediction gain of the PBFLNN, BFLNN, and FLNN are 18.912 dB, 19.324 dB and 17.160dB, respectively. From Fig. 4 and the value of the one-step prediction, we find that the speech signal prediction ability of the PBFLNN is equivalent to that of the BFLNN.

4 Conclusion

This paper has proposed a PBFLNN filter, aiming to reduce computation cost and extend the application scope for the BFLNN. The architecture of the proposed filter is simpler with a shorter memory length. Computational analysis and simulation results have shown that the PBFLNN filter significantly reduces computation cost without degrading performance compared to BFLNN. Furthermore, the simulations have also demonstrated the potential of the PBFLNN filter for nonlinear dynamic identification and speech signal prediction.

Acknowledgements. This work was supported by Ministry of Education and Training, Vietnam fund (project title: Research to reduce computational complexity and impulsive noise impact for nonlinear active noise control (ANC) system, grant B2021-TDV-03).

References

1. Ogunfunmi, T.: Adaptive Nonlinear System Identification. Springer, New York (2007)
2. Diniz, P.: Adaptive Filtering Algorithms and Practical Implementation, 3rd edn. Springer, New York (2008)
3. Li Tan, Jiang, J.: Adaptive Volterra filters for active control of nonlinear noise processes. IEEE Trans. Sig. Process. **49**(8), 1667–1676 (2001)
4. Patra, J.C., Pal, R.N.: A functional link artificial neural network for adaptive channel equalization. Sig. Process. **43**(2), 181–195 (1995)
5. Comminiello, D., Scarpiniti, M., Azpicueta-Ruiz, L.A., Arenas-García, J., Uncini, A.: Functional link adaptive filters for nonlinear acoustic echo cancellation. IEEE Trans. Audio Speech Lang. Process **21**(7), 1502–1512 (2013)
6. Le, D.C., Zhang, J., Li, D.F., Zhang, S.: A generalized exponential functional link artificial neural networks filter with channel-reduced diagonal structure for nonlinear active noise control. Appl. Acoust. **139**, 174–81 (2018)
7. Chakravorti, T., Satyanarayana, P.: Nonlinear system identification using kernel based eponentially extended random vector functional link network. Appl. Soft Comput. **89**, 1–14 (2020)
8. Le, D.C., Zhang, J., Li, D.: Hierarchical partial update generalized functional link artificial neural network filter for nonlinear active noise control. Digit. Sig. Process **93**, 160–171 (2019)
9. Sicuranza, G.L., Carini, A.: A generalized FLANN filter for nonlinear active noise control. IEEE Trans. Audio Speech Lang. Process **19**(8), 2412–2417 (2011)
10. Le, D.C., Zhang, J., Pang, Y.: A bilinear functional link artificial neural network filter for nonlinear active noise control and its stability condition. Appl. Acoust. **132**, 19–25 (2018)

11. Haykin, S., Li, L.: Nonlinear adaptive prediction of nonstationary signals. IEEE Trans. Sig. Process **43**(2), 526–535 (1995)
12. Baltersee, J., Chambers, J.A.: Nonlinear adaptive prediction of speech with a pipelined recurrent neural network. IEEE Trans. Sig. Process **46**(8), 2207–2216 (1998)
13. Goh, S.L., Mandic, D.P.: Nonlinear adaptive prediction of complex-valued signals by complex-valued PRNN. IEEE Trans. Sig. Process **53**(5), 1827–1836 (2005)
14. Zhang, S., Zhang, J., Pang, Y.: Pipelined set-membership approach to adaptive Volterra filtering. Sig. Process **129**, 195–203 (2016)
15. Patra, J.C., Pal, R.N., Chatterji, B.N., Panda, G.: Identification of nonlinear dynamic systems using functional link artificial neural networks. IEEE Trans. Syst. Man. Cybern. B **29**(2), 254–262 (1999)