A GA-Based Pruning Strategy and Weight Update Algorithm for Efficient Nonlinear System Identification

G. Panda, Babita Majhi, D. Mohanty, and A.K. Sahoo

ECE department, National Institute of Technology Rourkela,
Rourkela - 769 008, Orissa, India

Abstract. Identification of nonlinear static and dynamic systems plays a key role in many engineering applications including communication, control and instrumentation. Various adaptive models have been suggested in the literature using ANN and Fuzzy logic based structures. In this paper we employ an efficient and low complexity Functional Link ANN (FLANN) model for identifying such nonlinear systems using GA based learning of connective weights. In addition, pruning of connecting paths is also simultaneously carried out using GA to reduce the network architecture. Computer simulations on various static and dynamic systems indicate that there is more than 50% reduction in original model FLANN structure with almost equivalent identification performance.

1 Introduction

The area of system identification is one of the most important areas in engineering because of its applicability to a wide range of problems. Recently, Artificial Neural Networks (ANN) has emerged as a powerful learning technique to perform complex tasks in highly nonlinear dynamic environments. At present, most of the work on system identification using neural networks are based on multilayer feedforward neural networks with backpropagation learning or more efficient variations of this algorithm [1,2]. On the other hand the Functional link ANN (FLANN) originally proposed by Pao [3] is a single layer structure with functionally mapped inputs. The performance of FLANN for system identification of nonlinear systems has been reported [4] in the literature. Patra and Kot [5] have used Chebyschev expansions for nonlinear system identification and have shown that the identification performance is better than that offered by the multilayer ANN (MLANN) model. Evolutionary computation has been applied to search optimal values of recursive least-square (RLS) algorithm used in the system identification model [6].

While constructing an artificial neural network the designer is often faced with the problem of choosing a network of the right size for the task to be carried out. The advantage of using a reduced neural network is less costly and faster in operation. However, a much reduced network cannot solve the required problem while a fully ANN may lead to accurate solution. Choosing an appropriate ANN

A. Ghosh, R.K. De, and S.K. Pal (Eds.): PReMI 2007, LNCS 4815, pp. 244–251, 2007.
© Springer-Verlag Berlin Heidelberg 2007

architecture of a learning task is then an important issue in training neural networks. Giles and Omlin [7] have applied the pruning strategy for recurrent networks. In this paper we have considered an adequately expanded FLANN model for the identification of nonlinear plant and then used Genetic Algorithm (GA) to train the filter weights as well to obtain the pruned input paths based on their contributions. Procedure for simultaneous pruning and training of weights have been carried out in subsequent sections to obtain a low complexity reduced structure.

The rest of paper is organized as follows. In Section 2 the basics of adaptive system identification is presented. Section 3 illustrates the proposed GA based pruning and training method using FLANN structure. The performance of the proposed model obtained from computer simulations are presented in Section 4. We present some concluding remarks in Section 5.

2 Adaptive System Identification

The essential and principal property of an adaptive system is its time-varying, self-adjusting performance. System identification concerns with the determination of a system, on the basis of input output data samples. The identification task is to determine a suitable estimate of finite dimensional parameters which completely characterize the plant. Depending upon input-output relation, the identification of systems can have two groups. In static identification the output at any instant depends upon the input at that instant. The system is essentially a memoryless one and mathematically it is represented as $y(n) = f[x(n)]$ where $y(n)$ is the output at the nth instant corresponding to the input $x(n)$. In dynamic identification the output at any instant depends upon the input at that instant as well as the past inputs and outputs. These systems have memory to store past values and mathematically represented as $y(n) = f[x(n), x(n-1), x(n-2) \cdots y(n-1), y(n-2), \cdots]$ where $y(n)$ is the output at the nth instant corresponding to the input $x(n)$. A basic system identification structure is shown in figure 1. The impulse response of the linear segment of the plant is represented by $h(n)$ which is followed by nonlinearity (NL) associated with it. White Gaussian noise $q(n)$ is added with nonlinear output accounts for measurement noise. The desired output $d(n)$ is compared with the estimated output $y(n)$ of the identifier to generate the error $e(n)$ for updating the weights of the model. The training of the filter weights is continued until

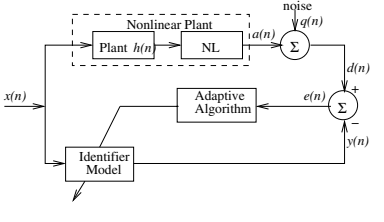


Fig. 1. Block diagram of system identification structure

the error becomes minimum and does not decrease further. At this stage the correlation between input signal and error signal is minimum.

3 Proposed GA Based Pruning and Training Using FLANN Structure

The FLANN based system identification is shown in figure 2. The FLANN is a single layer network in which the hidden nodes are absent. Here each input pattern is functionally expanded to generate a set of linearly independent functions. The functional expansion is achieved using trigonometric, polynomial or exponential functions. An N dimensional input pattern $X = \left[x_1\ x_2 \cdots x_N\right]^T$. Thus the functionally expanded patterns becomes $X^* = \left[1\ x_1\ f_1(x_1) \cdots x_N\ f_1(x_N)\right]$ where all the terms in the square bracket represents enhanced patterns. Then the improved patterns are used for pattern classification purpose.

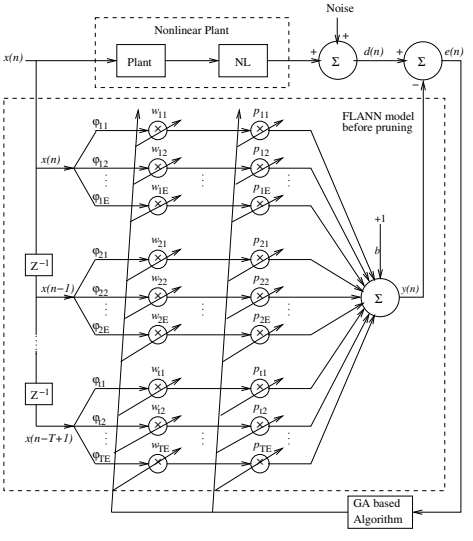


Fig. 2. FLANN based identification of nonlinear plants showing updating of weights and pruning path

In this Section a new algorithm for simultaneous training and pruning of weights using binary coded genetic algorithm (BGA) is proposed. Such a choice has led to effective pruning of branch and updating of weights. The pruning strategy is based on the idea of successive elimination of less productive paths (functional expansions) and elimination of weights from the FLANN architecture. As a result, many branches (functional expansions) are pruned and the overall architecture of the FLANN based model is reduced which in turn reduces the corresponding computational cost associated with the proposed model without sacrificing the performance. Various steps involved in this algorithm are dealt in this section.

Step 1-Initialization in GA: A population of M chromosomes is selected in GA in which each chromosome constitutes $TE(L+1)$ number of random binary bits where the first TE number of bits are called Pruning bits (P) and the remaining bits represent the weights associated with various branches (functional expansions) of the FLANN model. Again $(T-1)$ represents the order the filter and E represents the number of expansions specified for each input to the filter.

A pruning bit (p) from the set P indicates the presence or absence of expansion branch which ultimately signifies the usefulness of a feature extracted from the time series. In other words a binary 1 will indicate that the corresponding branch contributes and thus establishes a physical connection where as a 0-bit indicates that the effect of that path is insignificant and hence can be neglected. The remaining TEL bits represent the TE weight values of the model each containing L bits.

Step 2-Generation of Input Training Data: $K \geq (500)$ number of signal samples is generated. In the present case two different types of signals are generated to identify the static and feed forward dynamic plants.

i. To identify a feed forward dynamic plant, a zero mean signal which is uniformly distributed between ± 0.5 is generated.
ii. To identify a static system, a uniformly distributed signal is generated within ± 1.

Each of the input samples are passed through the unknown plant (static and feed forward dynamic plant) and K such outputs are obtained. The plant output is then added with the measurement noise (white uniform noise) of known strength, there by producing k number of desired signals. Thus the training data are produced to train the network.

Step 3-Decoding: Each chromosome in GA constitutes random binary bits. So these chromosomes need to be converted to decimal values lying between some ranges to compute the fitness function. The equation that converts the binary coded chromosome in to real numbers is given by

$$RV = R_{min} + \left\{(R_{max} - R_{min})/(2^L - 1)\right\} \times DV \tag{1}$$

Where R_{min}, R_{max}, RV and DV represent the minimum range, maximum range, decimal and decoded value of an L bit coding scheme representation. The first L number of bits is not decoded since they represent pruning bits.

Step 4-To Compute the Estimated Output: At nth instant the estimated output of the neuron can be computed as

$$y(n) = \sum_{i=1}^{T} \sum_{j=1}^{E} \varphi_{ij}(n) \times W_{ij}^{m}(n) \times P_{ij}^{m}(n) + b^{m}(n) \tag{2}$$

where $\varphi_{ij}(n)$ represents j^{th} expansion of the i^{th} signal sample at the n^{th} instant. $W_{ij}^{m}(n)$ and $P_{ij}^{m}(n)$ represent the j^{th} expansion weight and j^{th} pruning weight

of the i^{th} signal sample for m^{th} chromosome at k^{th} instant. Again $b^m(n)$ corresponds to the bias value fed to the neuron for m^{th} chromosome at nth instant.

Step 5-Calculation of Cost Function: Each of the desired output is compared with corresponding estimated output and K errors are produced. The Mean-square-error (MSE) corresponding to m^{th} chromosome is determined by using the relation (3). This is repeated for M times (i.e. for all the possible solutions).

$$MSE(m) = \sum_{k=1}^{K} e_k^2/K \tag{3}$$

Step 6-Operations of GA: Here the GA is used to minimize the MSE. The crossover, mutation and selection operators are carried out sequentially to select the best M individuals which will be treated as parents in the next generation.

Step 7-Stopping Criteria: The training procedure will be ceased when the MSE settles to a desirable level. At this moment all the chromosomes attain the same genes. Then each gene in the chromosome represents an estimated weight.

4 Simulation Study

Extensive simulation studies are carried out with several examples from static as well as feed forward dynamic systems. The performance of the proposed Pruned FLANN model is compared with that of basic FLANN structure. For this the algorithm used in Sections 3 is used in the simulation.

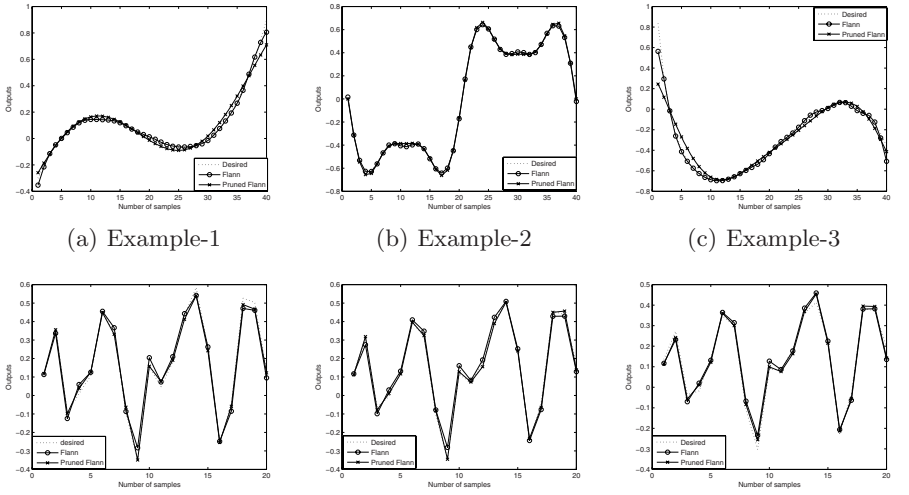
4.1 Static Systems

Here different nonlinear static systems are chosen to examine the approximation capabilities of the basic FLANN and proposed Pruned FLANN models. In all the simulation studies reported in this section a single layer FLANN structure having one input node and one neuron is considered. Each input pattern is expanded using trigonometric polynomials i.e. by using $\cos(n\pi u)$ and $\sin(n\pi u)$, for $n = 0, 1, 2, \cdots, 6$. In addition a bias is also fed to the output. In the simulation work the data used are $K = 500$, $M = 40$, $N = 15$, $L = 30$, probability of crossover $= 0.7$ and probability of mutation $= 0.1$. Besides that the R_{max} and R_{min} values are judiciously chosen to attain satisfactory results. Three nonlinear static plants considered for this study are as follows:

Example-1: $f_1(x) = x^3 + 0.3x^2 - 0.4x$

Example-2: $f_2(x) = 0.6\sin(\pi x) + 0.3\sin(3\pi x) + 0.1sin(5\pi x)$

Example-3: $f_3(x) = (4x^3 - 1.2x^2 - 3x + 1.2)/(0.4x^5 + 0.8x^4 - 1.2x^3 + 0.2x^2 - 3)$



(a) Example-1 (b) Example-2 (c) Example-3

(d) Example-1 (e) Example-2 (f) Example-3

Fig. 3. Comparisons of output response; (a)–(c) for static systems of Example 1–3 and (d)–(f) for dynamic systems of Examples 1–3

At any n^{th} instant, the output of the ANN model $y(n)$ and the output of the system $d(n)$ is compared to produce error $e(n)$ which is then utilized to update the weights of the model. The LMS algorithm is used to adapt the weights of basic FLANN model where as a proposed GA based algorithm is employed for simultaneous adaptation of weights and pruning of the branches. The basic FLANN model is trained for 20000 iterations where as the proposed FLANN model is trained for only 60 generations. Finally the weights of the ANN are stored for testing purpose. The responses of both the networks are compared during testing operation and shown in figures 3(a), 3(b), and 3(c).

The results of identification of $f_1(\cdot)$, $f_2(\cdot)$ and $f_3(\cdot)$ are shown figures 3(a)–3(c). In the figures the actual system output, basic FLANN output and pruned FLANN output are marked as "Desired", "FLANN" and "Pruned FLANN" respectively. From these figures, it may be observed that the identification performance of the FLANN model with all the examples is quite satisfactory. For

Table 1. Comparison of computational complexities between a basic FLANN and a Pruned FLANN model for Static Systems

Ex. No.	Number of operations				Number of weights	
	Additions		Multiplications			
	FLANN	Prunned FLANN	FLANN	Prunned FLANN	FLANN	Prunned FLANN
Ex-1	14	3	14	3	15	4
Ex-2	14	2	14	3	15	3
Ex-3	14	5	14	5	15	6

the Pruned FLANN structure, quite close agreement between the system output and the model output is observed. In fact, the modeling error of the pruned FLANN structure is found to be comparable with that of the basic FLANN structure for all the three nonlinear static structures considered. Table 1 illustrates the total computational complexity involved in both the architectures to identify the same system.

4.2 Dynamic Systems

In the following the simulation studies of nonlinear dynamic feed forward systems has been carried out with the help of several examples. In each example, one particular model of the unknown system is considered. In this simulation a single layer FLANN structure having one input node and one neuron is considered. Each input pattern is expanded using the direct input as well as the trigonometric polynomials i.e. by using u, $\cos(n\pi u)$ and $\sin(n\pi u)$, for $n = 1$. In this case the bias is removed. In the simulation work we have considered $K = 500$, $M = 40$, $N = 9$, $L = 20$, probability of crossover = 0.7 and probability of mutation = 0.03. Besides that the R_{max} and R_{min} values are judiciously chosen to attain satisfactory results. The three nonlinear dynamic feed forward plants considered for this study are as follows:

Example-1:

i. Parameter of the linear system of the plant $\begin{bmatrix}0.2600 & 0.9300 & 0.2600\end{bmatrix}$
ii. Nonlinearity associated with the plant $y_n(k) = y_k + 0.2y_k^2 - 0.1y_k^3$

Example-2:

i. Parameter of the linear system of the plant $\begin{bmatrix}0.3040 & 0.9029 & 0.3410\end{bmatrix}$
ii. Nonlinearity associated with the plant $y_n(k) = \tanh(y_k)$

Example-3:

i. Parameter of the linear system of the plant $\begin{bmatrix}0.3410 & 0.8760 & 0.3410\end{bmatrix}$
ii. Nonlinearity associated with the plant $y_n(k) = y_k - 0.9y_k^3 + 0.2y_k^2 - 0.1y_k^3$

The basic FLANN model is trained for 2000 iterations where as the proposed FLANN is trained for only 60 generations. While training, a white uniform noise of strength $-30dB$ is added to actual system response to assess the performance of two different models under noisy condition. Then the weights of the ANN are stored for testing. Finally the testing of the networks model is undertaken by presenting a zero mean white random signal to the identified model. Performance comparison between the FLANN and pruned FLANN structure in terms of estimated output of the unknown plant has been carried out. The responses of both the networks are compared during testing operation and shown in figures 3(d), 3(e), and 3(f).

The results of identification of all the examples are shown in figures 3(d)–3(f). In the figures the actual system output, basic FLANN output and pruned FLANN output are marked as "Desired", "FLANN" and "Pruned FLANN" respectively. From the simulation results, it may be seen that the model output

Table 2. Comparison of computational complexities between a basic FLANN and a pruned FLANN model for Dynamic Systems

Ex. No.	Number of operations				Number of weights	
	Additions		Multiplications			
	FLANN	Prunned FLANN	FLANN	Prunned FLANN	FLANN	Prunned FLANN
Ex-1	8	3	9	4	9	4
Ex-2	8	2	9	3	9	3
Ex-3	8	2	9	3	9	3

responses closely agree with those of plant output for both the FLANN and the pruned FLANN based structures. Comparison of computational complexities between the conventional FLANN and the pruned FLANN is provided in Table 2. From Tables 1 and 2 it is evident that for all the identification performance cases studied, the computational load on the pruned FLANN is much lower than that of FLANN model.

5 Conclusions

The present paper has proposed simultaneous weight updating and pruning of FLANN identification models using GA. Computer simulation studies on static and dynamic nonlinear plants demonstrate that there is more than 50% active paths are pruned keeping response matching identical with those obtaining from conventional FLANN identification models.

References

1. Jagannathan, S., Lewis, F.L.: Identification of a class of nonlinear dynamical systems using multilayered neural networks. In: IEEE International Symposium on Intelligent Control, Columbus, Ohio, USA, pp. 345–351 (1994)
2. Narendra, K.S., Parthasarathy, K.: Identification and control of dynamical system using neural networks. IEEE Trans. Neural Networks 1(1), 4–26 (1990)
3. Pao, Y.H.: Adaptive Pattern Recognition and Neural Network. Addison Wesley, Reading, MA (1989)
4. Patra, J.C., Pal, R.N., Chatterji, B.N., Panda, G.: Identification of nonlinear dynamic systems using functional link artificial neural networks. IEEE Trans.on Systems, Man and Cybernetics-Part B 29(2), 254–262 (1999)
5. Patra, J.C., Kot, A.C.: Nonlinear dynamic system identification using chebyshev functional link artificial neural networks. IEEE Trans. on Systems Man and Cybernetics-Part B 32(4), 505–511 (2002)
6. Juang, J.G., Lin, B.S.: Nonlinear system identification by evolutionary computation and recursive estimation method. In: Proceedings of American Control Conference, pp. 5073–5078 (2005)
7. Giles, C.L., Omlin, C.W.: Pruning recurrent neural networks for improved generalization performance. IEEE Trans. on Neural Networks 5(5), 848–851 (1994)