Comparison of Extreme-ANFIS and ANFIS Networks for Regression Problems

Pushpak Jagtap
Department of Electrical Engineering
Indian Institute of Technology
Roorkee, India
pushpakjagtap@gmail.com

G. N. Pillai
Department of Electrical Engineering
Indian Institute of Technology
Roorkee, India
gopinath.pillai@gmail.com

Abstract— **This paper compares the performance of conventional adaptive network based fuzzy inference system (ANFIS) network and extreme-ANFIS on regression problems. ANFIS networks incorporate the explicit knowledge of the fuzzy systems and learning capabilities of neural networks. The proposed new learning technique overcomes the slow learning speed of the conventional learning techniques like neural networks and support vector machines (SVM) without sacrificing the generalization capability. The structure of extreme-ANFIS network is similar to the conventional ANFIS which combines the fuzzy logic's qualitative approach and neural network's adaptive capability. As in the case of extreme learning machines (ELM), the first layer parameters of the proposed learning machine are not tuned. Performance on two regression problems shows that extreme-ANFIS provides better generalization capability and faster learning speed.**

Keywords—ANFIS; Extreme-ANFIS learning algorithm; Hybrid learning algorithm (HLA)

I. INTRODUCTION

Extreme learning machine (ELM) [1] is a new learning algorithm which works on single-hidden layer feed-forward neural networks (SLFNs). Conventional learning techniques like neural networks (NN) and support vector machines suffer drawbacks like slow learning speed and trivial learning variations for different applications of regression problems and two class and multi-class classification problems. All the parameters of conventional SLFNs need to be tuned and thus there exists the dependency between parameters of hidden layer and output layer. Traditional learning techniques based on gradient descent methods are generally very slow due to improper learning steps and have the problem of converging to local minima. The essence of ELM is that the hidden layer parameters cannot be dependent of training samples, and these parameters need not be tuned. Recent research on ELM ([2], [7]) has shown that ELM is better than support vector machines and neural networks because of faster learning speed and smaller generalization error.

Fuzzy logic systems do not incorporate any learning, while neural networks, a black box approach, do not possess mechanisms for explicit knowledge representation. By judicious integration of neural networks and fuzzy logic, advantages of both these approaches can be incorporated in neuro-fuzzy systems. Adaptive network based fuzzy inference system (ANFIS) is a hybrid intelligent system which combines the fuzzy logic's qualitative approach and neural network's adaptive capabilities towards better performance [3]. ANFIS uses a hybrid learning algorithm to identify parameters of Sugeno-type [4] fuzzy inference systems. Many applications of ANFIS are reported in literature [5].

In spite of the advantages, ELM suffers from the inherent randomness of the results and ANFIS has strong computational complexity restrictions because of hybrid learning algorithm (HLA). A faster learning technique 'Extreme-ANFIS' is proposed in [6]. Extreme-ANFIS reduces the computation complexity of the ANFIS by eliminating the hybrid learning algorithm and avoids the randomness of the ELM networks by incorporating explicit knowledge representation using fuzzy membership functions. This paper shows that the generalization capability of extreme-ANFIS is better than that of conventional ANFIS trained using hybrid learning algorithm. This paper also enumerates the conclusive features of extreme-ANFIS networks.

The remaining sections of the paper are organized as follows. Architecture of the conventional ANFIS networks and hybrid learning algorithm is discussed in section 2. The extreme-ANFIS learning algorithm is described in section 3. Section 4 presents simulations and performance evaluation of extreme-ANFIS networks on two regression problems. Sections 5 presents the discussions and conclusions.

II. ANFIS ARCHITECTURE AND HYBRID LEARNING ALGORITHM

ANFIS maps first order Sugeno fuzzy inference system in multilayer feed-forward adaptive neural network to enhance performance by adding attractive features such as fast and accurate learning and fine tuning of membership function parameters by analysing both linguistic and numerical knowledge. The first order Sugeno fuzzy model, its inference mechanism and defuzzification process is shown in Fig. 1. The typical fuzzy If-Then rule set is used to describe ANFIS architecture. It is expressed as follow:

Rule 1: If x is A_1 and y is B_1, then $f_1 = p_1 x + q_1 y + r_1$
Rule 2: If x is A_2 and y is B_2, then $f_2 = p_1 x + q_2 y + r_2$

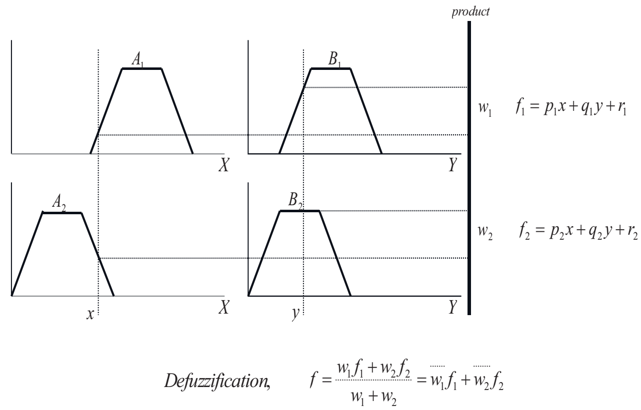

Figure. 1 First order Sugeno Fuzzy inference mechanism

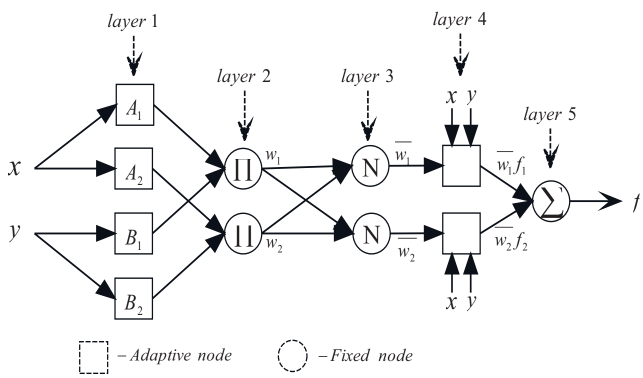

Figure. 2 ANFIS architecture [3]

where x and y are the crisp inputs, A_i, B_i are linguistic variables. The five layer ANFIS architecture explained by Jang [3] is shown in Fig. 2. The function of each layer is described as follow:

1) Layer-1: Every node i in this layer represents fuzzy membership function as node function with an adaptive parameters.

$$O_i^1 = \mu_{A_i}(x), \qquad i = 1,2 \tag{1}$$

where x is input value, O_i^1 is membership value of fuzzy variable A_i. a_i, b_i, c_i are the adaptive parameters commonly referred as premise parameters.

2) Layer-2: Every node in this layer is fixed node which acts like product operation as in Sugeno fuzzy model.

$$O_i^2 = W_i = \mu_{A_i}(x) \ \times \ \mu_{B_i}(y), \qquad i = 1,2 \tag{2}$$

3) Layer-3: layer contains fixed nodes, which calculates normalized firing strength, $\bar{W}_i$ as follows

$$\bar{W}_i = \frac{W_i}{W_1 + W_2}, \qquad i = 1,2 \tag{3}$$

4) Layer-4: Every node i, is an adaptive node with a node function given as

$$O_i^4 = \bar{W}_i f_i = \bar{W}_i(p_i x + q_i + r_i) \tag{4}$$

where $\overline{W_i}$ is the output of layer-3 and $\{p_i, q_i, r_i\}$ are adaptive consequent parameters.

5) Layer-5: It is fixed single node that computes overall output as summation of all incoming signals from layer-4.

$$O_i^5 = \sum_i \bar{W}_i f_i = \frac{\sum_i W_i f_i}{\sum_i W_i} \tag{5}$$

The premise and consequent parameters are updated to minimise error by various learning algorithms. The most commonly used algorithm is hybrid learning algorithm [3]. This algorithm combines the gradient based method (i.e. back propagation) and least square error (LSE) to identify parameters. The hybrid learning algorithm (HLA) consists of two passes explained as follows:

a) Forward pass: In the forward pass of the hybrid learning algorithm, node outputs pass forward till layer-4 by assuming some premise parameters and the consequent parameters are identified by the least-square error method. As the values of the premise parameters are fixed, the overall output can be expressed as a linear combination of the consequent parameters

$$\begin{aligned} f &= \frac{W_1}{W_1 + W_2} f_1 + \frac{W_1}{W_1 + W_2} f_2 \\ &= \overline{W_1} f_1 + \overline{W_2} f_2 \\ &= (\overline{W_1} x) p_1 + (\overline{W_1} y) q_1 + (\overline{W_1}) r_1 \\ &\quad + (\overline{W_2} x) p_2 + (\overline{W_2} y) q_2 + (\overline{W_2}) r_2 \end{aligned} \tag{6}$$

which is linear in the consequent parameters *p1, q1, r1, p2, q2* and *r2*

$$f = XZ \tag{7}$$

If X matrix is invertible then,

$$Z = X^{-1} f \tag{8}$$

Otherwise solution of Z could be obtained using pseudo-inverse method.

$$Z = (X^T X)^{-1} X^T f \tag{9}$$

b) Backward pass: In the backward pass, the error signals propagate backward and the premise parameters are updated with the help of gradient descent method by keeping consequent parameters fixed. Parameter updating rule is given as

$$a_{ij}(t+1) = a_{ij}(t) - \eta . \frac{\partial E}{\partial a_{ij}} \tag{10}$$

where η is the learning rate for parameter a_{ij}, the gradient is obtained using chain rule as

$$\frac{\partial E}{\partial a_{ij}} = e.1. \frac{(p_i x + q_i y + r_i) - f}{\sum_{i=1}^{n} w_i} . \frac{w_i}{\mu_{A_{ij}}} . \frac{\partial \mu_{A_{ij}}}{\partial a_{ij}} \tag{11}$$

$$\frac{\partial E}{\partial b_{ij}} = e.1.\frac{(p_i x + q_i y + r_i) - f}{\sum_{i=1}^{n} w_i}.\frac{w_i}{\mu_{B_{ij}}}.\frac{\partial \mu_{B_{ij}}}{\partial b_{ij}} \quad (12)$$

By using forward pass and backward pass alternately, the ANFIS parameters have been trained. But there is some drawback while using HLA for training. The paper proposed a new algorithm to cope with the difficulties in hybrid learning algorithm and to train parameters in minimum time. Detailed description of proposed learning algorithm has been given in following section.

III. Extreme-ANFIS Learning Algorithm

Based on the concept of Extreme Learning Machine (ELM) [1], [7], the new *Extreme-ANFIS learning Algorithm* has been proposed in this Section. The proposed algorithm has helped to minimize learning time of ANFIS architecture. The advantages of ANFIS architecture and its approximating capabilities with great accuracy has been proved already by Jang in 1993 [3]. The architecture and Hybrid Learning Algorithm introduced by Jang is explained in previous Section. Hybrid Learning Algorithm uses two passes (forward and backward pass), which is a combination of LSE and back propagation based on gradient descent. As it uses gradient based method, it has certain drawbacks such as the calculation of gradient is possible with differentiable membership functions only, over fitting, and an iterative method hence time consuming.

The proposed algorithm, called Extreme-ANFIS learning algorithm is simple and derivative-less algorithm which eliminate drawbacks of gradient based Hybrid Learning Algorithm. The proposed algorithm can be summarized as follows:

Algorithm: Consider the training data scattered over the possible input range are available for given regression or modelling problem as:

$$[I_1 \ I_2 \ I_3 \ldots\ldots\ldots\ldots \ I_n; f]$$

where I_i, i=1, 2… n are inputs, and f is corresponding output.

Step 1: Calculate range of every input to get the universe of discourse for input membership functions as,

$$range_i = max\{I_i\} - min\{I_i\} \quad (13)$$

where i = 1, 2… n is number of input.

Step 2: Decide shape of membership function and number of membership functions which represent linguistic partitions of universe of discourse. Bell shaped membership function is commonly used because of its advantages over other membership functions such as smoothness in change in membership grade which is a drawback of triangular and trapezoidal membership functions, also it provides flexibility in core of membership function which is not possible in Gaussian type membership function. The mathematical representation of bell shape membership function is given as,

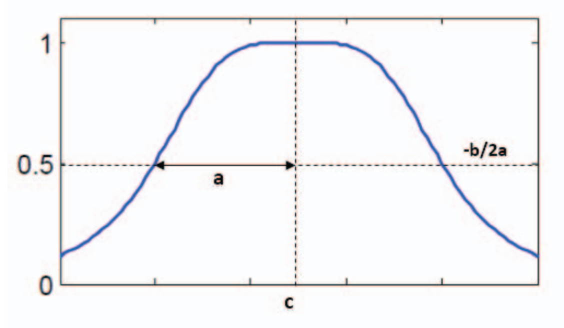

Figure 3. Bell shape membership function parameters

$$\mu_{ij} = \frac{1}{1 + \left|\frac{I_i - c_j}{a_j}\right|^{2b_j}} \quad (14)$$

where μ_{ij} represents membership grade of i^{th} input and j^{th} membership function, a_j, b_j, c_j are position and shape deciding parameters. The c_j represents the center of j^{th} membership function, a_j decides half width of membership function, $b_j/2a_j$ represents slope at membership grade μ_{ij} = 0.5. The significance of parameters are shown in Fig. 3.

Step 3: The premise parameters (a_j, b_j, c_j) values are generated randomly with some constraints on ranges of those parameters. These ranges depend on the number of membership functions used and size of universe of discourse.

Consider there are m uniformly distributed membership functions with parameters (a_j^*, b_j^*, c_j^*) in universe of discourse. The procedure to select random parameters is described below:

Selection of a_j: The parameter a_j decides the width of membership function. The default value of parameter in uniformly distributed membership functions is expressed as,

$$a_j^* = \frac{range_i}{2m - 2} \quad (15)$$

The range of selection of random value a_j is given as,

$$\frac{a_j^*}{2} \le a_j \le \frac{3a_j^*}{2} \quad (16)$$

Selection of b_j: The parameter b_j with the help of a_j gives slope as $b_j/2a_j$. The default value of b_j in uniformly distributed membership functions is 2. The slight change in this value significantly changes the slope hence its range is limited within 1.9 to 2.1.

Selection of c_j: The range for random value for center (c_j) of membership function is decided such that one center should not cross the center of consecutive membership function. The range of center value (c_j) selection is shown in equation (17)

$$\left(c_j^* - \frac{d_{cc}}{2}\right) < c_j < \left(c_j^* - \frac{d_{cc}}{2}\right) \quad (17)$$

where c_j^* is center of uniformly distributed membership function, d_{cc} is distance between two consecutive centers of uniformly distributed membership.

Step 4: Once the randomly generated membership functions are available, the node output up to fourth layer can be obtained easily. Now the final output f becomes a simple linear combination of consequent parameters as shown in equation (18)

$$f = \sum_{k=1}^{m^n} \bar{W}_k \left(\sum_{i=1}^{n} R_{ki} I_i + Q_k \right) \quad (18)$$

where k represents number of rule, m is number of membership functions, n is number of inputs, m^n are the maximum number of rules, i represents number of input, I_i is a value of i^{th} input, R_{ki} and Q_k are consequent parameters corresponding to k^{th} rule and i^{th} input.

Consider there are p numbers of training data pairs, then linear matrix of p equations are represented as,

$$F_{p\times 1} = \beta_{p\times m^n(n+1)} U_{m^n(n+1)\times 1} \quad (19)$$

where F is output matrix of β is weighted input parameter matrix, U is unknown consequent parameter matrix of shown dimension. It can be solved using least square error (LSE) method as has been explained in forward pass of Hybrid Learning Algorithm.

Step 5: Run algorithm after step 2 for 50-70 times and find out the root mean square error (RMSE) in each epoch. And finally generate the Sugeno type FIS model using the obtained premise and consequent parameters for least RMSE.

The proposed algorithm is implemented in MATLAB [8] and tested for its time efficiency and accuracy upon some benchmark problems. The simulation results and performance evaluation is discussed in next section.

IV. Simulation and Performance Evaluation

The performance analysis is conducted on the basis of time required to learn parameters from training data, training error and testing error of proposed algorithm. The results obtained have been compared with conventional ANFIS results. Section, results of two benchmark problems from previous work of Jang are compared [3], [5].

Example 1: consider two input nonlinear function given in equation (21)

$$z = \frac{\sin(x)}{x} . \frac{\sin(y)}{y} \quad (21)$$

Consider the inputs x and y varies in range of [-10, 10] × [-10, 10]. Equally spaced 121 training data pairs and 16 rules with four membership functions are used for adapting parameters of ANFIS using both conventional Hybrid Learning algorithm and proposed Extreme-ANFIS learning algorithm. And equally spaced 100 testing data are used to check generalization. The final membership functions after learning are given in Fig. 4 and Fig. 5. The training data and output surface obtained by using HLA and proposed algorithm are given in Fig. (a), (b) and (c) respectively. Time required to learn, training error and testing error are listed in Table 1. It is clearly seen that if the number of membership functions increases, the accuracy also increases. But in case of conventional ANFIS algorithm time to learn parameters also increases. This drawback is totally eliminated by Extreme- ANFIS learning algorithm without affecting accuracy and generalization of conventional method.

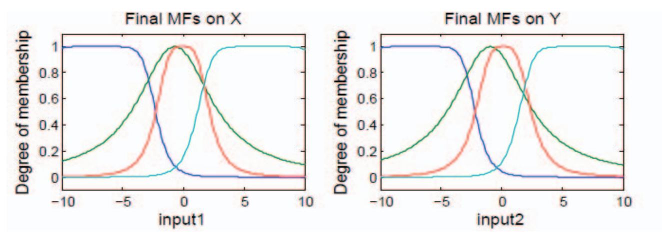

Figure 4. Final membership functions after learning with HLA

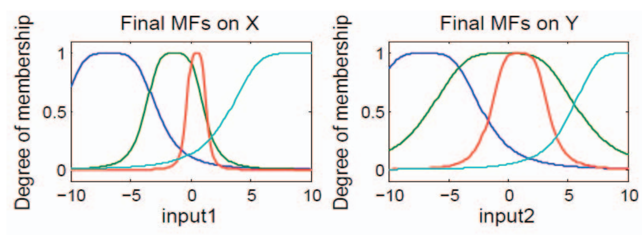

Figure 5. Final membership functions after learning with proposed algorithm

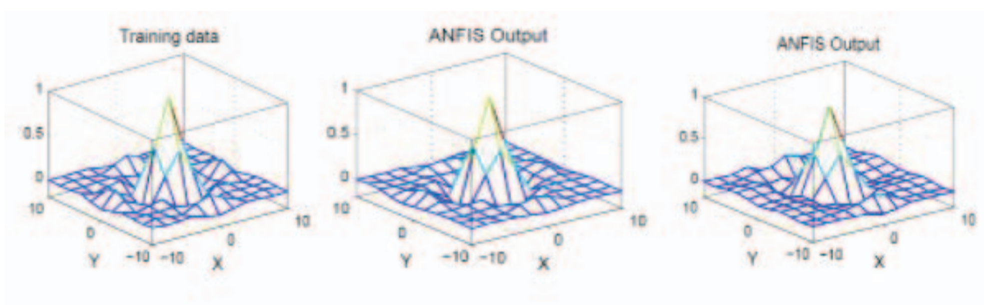

Figure 6. (a) Training data surface; (b)ANFIS output surface using HLA; (c) ANFIS output surface using proposed algorithm

TABLE 1. PERFORMANCE ANALYSIS OF EXAMPLE 1

	Number of membership functions	Time (s)	Training error (RMSE)	Testing error (RMSE)
HLA	4	0.814	0.0352	0.0862
Proposed		0.579	0.0418	0.0438
HLA	7	3.933	0.0004	0.0304
Proposed		1.177	0.0066	0.0411
HLA	11	23.4315	9.8×10^{-8}	0.0688
Proposed		1.918	1.35×10^{-14}	0.0625

TABLE 2. PERFORMANCE ANALYSIS OF EXAMPLE 2

	Number of membership functions	Time (s)	Training error (RMSE)	Testing error (RMSE)
HLA	2	0.588	0.0254	0.3128
Proposed		0.789	0.0942	0.4057
HLA	4	18.671	3.5×10^{-4}	2.1263
Proposed		2.603	0.0023	0.4825
HLA	5	96.06	7.9×10^{-5}	4.5413
Proposed		4.173	1.78×10^{-4}	1.4510

Example 2: The modelling of three input nonlinear function using ANFIS. The function is given as,

$$O = (1 + x^{0.5} + y^{-1} + z^{-1.5})^2 \tag{21}$$

where x, y and z are inputs with ranges [1,6]×[1,6]×[1,6]. 216 training data are used to train ANFIS. The time efficiency and training error is checked by using training data. Generalization of obtained ANFIS model is analyzed for 125 testing data in range [1.5, 5.5] × [1.5, 5.5] × [1.5, 5.5]. The overall performance analysis for different number of membership functions is listed in Table 2. It is seen from observed results, if the number of membership function increases the over- fitting occurred in conventional gradient based algorithm which affected generalization badly. But in the proposed algorithm generalization is not much affected while improving accuracy in terms of training error. Learning times and errors listed in tables are obtained by taking average of ten to fifteen trails. The conclusive features of proposed learning algorithm are discussed in next section.

V. DISCUSSION AND CONCLUSION

The paper proposed a simple and time efficient learning algorithm for adapting parameters of ANFIS architecture called Extreme-ANFIS. The significant features of proposed Extreme-ANFIS algorithm as compared to conventional hybrid learning algorithm are listed below:

- As the backward pass of hybrid learning algorithm is completely eliminated in proposed algorithm, the time required to find gradient and to update premise parameters iteratively reduced which resulted in significant reduction in overall learning time.
- The proposed algorithm is much simpler, faster and provides better generalization than conventional hybrid learning algorithm.
- The intuitive assumption of random premise parameters in the form of membership functions which are spread throughout the universe of discourse of input variable and local mapping ability of Sugeno type FIS in the form of rule base helps a lot in further improvement of learning speed.
- Proposed algorithm improves the flexibility of ANFIS architecture by eliminating differentiability constraint on membership function. In other words, the algorithm can also work with nondifferentiable membership functions. Also the different shapes of membership functions could be practiced easily within same universe of discourse of input.
- The reduction in steps of learning algorithm allows ANFIS architecture to increase number of inputs and number of membership functions within required time constraints to improve accuracy while modelling the complex nonlinear systems.

REFERENCES

[1] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, “Extreme learning machine: theory and applications,” Neurocomputing 70 (2006) 489–501.

[2] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, “Extreme learning machine for regression and multiclass classification,” IEEE Trans. Syst. Man Cyber. B Cyber. 42 (2) (2012) 513–529.

[3] J.-S. R. Jang, “ANFIS: Adaptive-network-based fuzzy inference systems,” IEEE Trans. Svst., Man, Cybern., vol. 23, no. 3, pp. 665-685, 1993.

[4] Sugeno M. and Kang G.T. “Structure identification of fuzzy model,” IEEE transactions on fuzzy sets and systems, Vol.28, No. 4, pp.15-33

[5] J . S. Jang, C. T. Sun and E. Mizutani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, NJ: Prentice-Hall, 1997.

[6] G. N. Pillai, “Extreme ANFIS: A New Learning Machine for Faster Learning,” accepted in International Conference on knowledge collaboration in engineering, to be held in Coimbatore, January 2014.

[7] Guang-Bin Huang; Qin-Yu Zhu; Chee-Kheong Siew, “Extreme learning machine: a new learning scheme of feedforward neural networks,” Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on , vol.2, no., pp.985,990 vol.2, 25-29 July 2004.

[8] Jang, J. S., and Gulley, N., Fuzzy Logic Toolbox User's Guide, the Math Works Inc., January 1995.