VOL. 10, NO. 20, NOVEMBER 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

ANALYSIS OF TECHNIQUES FOR ANFIS RULE-BASE MINIMIZATION AND ACCURACY MAXIMIZATION

Kashif Hussain and Mohd. Najib Mohd. Salleh
Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Batu Pahat, Johor, Malaysia
E-Mail: hi130019@siswa.uthm.edu.my

ABSTRACT

Despite of acquiring popularity among researchers, the implementations of ANFIS-based models face problems when the number of rules surge dramatically and increase the network complexity, which consequently adds computational cost. Essentially, not all the rules in ANFIS knowledge-base are the potential ones. They contain those rules which have either minor or no contribution to overall decision. Thus, removing such rules will not only reduce complexity of the network, but also cut computational cost. Thus, there are various rule-base optimization techniques, proposed in literature, which are presented in motivation to simultaneously obtain rule-base minimization and accuracy maximization. This paper analyzes some of those approaches and important issues related to achieving both the contradictory objectives simultaneously. In this paper, Hyperplane Clustering, Subtractive Clustering, and the approach based on selecting and pruning rules are analyzed in terms of optimizing ANFIS rule-base. The optimized rule-base is observed in connection with providing high accuracy. The results and analysis, presented in this paper, suggest that the clustering approaches are proficient in minimizing ANFIS rule-base with maximum accuracy. Although, other approaches, like putting threshold on rules' firing strength, can also be improved using metaheuristic algorithms.

Keywords: ANFIS, neuro-fuzzy, fuzzy systems, fuzzy clustering, rule-base minimization, rule optimization.

INTRODUCTION

Adaptive Neuro Fuzzy Inference System (ANFIS) provides technique for efficiently solving non-linear real-world problems. It has been popular among other fuzzy inference systems due to flexibility, simplicity and ease in understanding. Therefore, it has been successfully applied to model various types of control systems, expert systems, and other complex systems in a variety of fields including economics, engineering, agriculture, medical, and social sciences (Kar, Das, and Ghosh, 2014; Taylan and Karagözoğlu, 2009). With proper number of rules, ANFIS can approximate almost every plant; thus considered as universal approximator (Liu, Leng, and Fang, 2013). In ANFIS, the structure of the rule node is formed by the linguistic fuzzy rule *If-Then* model that is self-generated by the system. The number of rule nodes is dependent on the *n* number of inputs and *m* number of linguistic fuzzy terms. Just like in grid partitioning method m^n rules are generated by default. Rule-base is the main part of any fuzzy inference system (FIS) and the quality of results in it depends on effectiveness of these rules (Neshat *et al.*, 2012).

However, it is noteworthy that increasing the number of the rules, increases the number of neurons in the hidden layer of the network (Abiyev, Mamedov, and Al-shanableh, 2007). Moreover, all the self-generated rules of ANFIS architecture are not the important ones or do not contribute enough for the accuracy improvement. There exist many which are inefficient as well and can be pruned to lessen the complexity of FIS system (Rini, Shamsuddin, and Yuhaniz, 2013). Gorzalczany (2001) also suggests in his book "Computational intelligence systems and applications: neuro-fuzzy and fuzzy neural synergisms" that pruning weaker rules from the fuzzy rule-base of ANFIS improves interoperability of the system. This will serve as to lessen the complexity of the ANFIS architecture while at the same time will save computational cost. It is also important to notice that over reducing rules may harm accuracy. Therefore, keeping balance between rule-minimization and accuracy maximization should be the key function of any rule-base optimization technique. Simultaneously achieving both the objectives is a trade-off problem (Ishibuchi and Nojima, 2009).

During the course of development in the research related to ANFIS, a number of methods have been proposed for learning rules to close the error gape and for obtaining an optimal set of rules (Teshnehlab, Shoorehdeli, and Sedigh, 2008). Though, the techniques which can efficiently minimize the number of rules in ANFIS knowledge-base and produce high accuracy are still to appear. The techniques discussed in this paper have been applied both on dataspace and the ANFIS rule-base. For extracting fuzzy rules from data and generating ANFIS with optimized rule-base, clustering techniques have been proposed in literature. These approaches group input data, output data or conjunct input-output data in a way to model the desired system behavior with maximum accuracy. On the other hand, some researchers have proposed putting threshold on fuzzy rules' firing strength in order to select the potential or efficient rules and remove inefficient or unnecessary ones to lighten the complexity of the ANFIS network. Some of these approaches have used non-linear classification algorithms while others are employing metaheuristic algorithms to search optimal number of rules. This is done to find optimal number of rules which meet both the low complexity and high accuracy while modeling ANFIS based systems.

VOL. 10, NO. 20, NOVEMBER 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

The core objective of this paper is to analyze different techniques proposed in literature to optimize ANFIS rule-base which mainly focused on clustering and rule-base minimization approaches. The rest of the paper is organized as follows: The next section gives a brief introduction of ANFIS architecture. Hyperplane clustering for ANFIS synthesis with optimal number of rules has been discussed later on. Then, we discuss Subtractive Clustering. Other than clustering, in this paper, we also present rule-base minimization techniques based on selecting and pruning rules. Results are discussed in the related section. The last section contains conclusion and future outline of rule-base optimization techniques for ANFIS network.

ANFIS CONCEPT

ANFIS was first developed by Jang (1993). It is one of the data learning techniques used in soft computing which utilizes training data to map the desired behavior through its rule-base. Formally, ANFIS comprises of *n* inputs with *m* dimensions per input variable. Thus, its rule-base comprises of m^n rules where *j*th rule can be expressed as:

$$Rule_j = \text{If } x^1 \text{ is } M^j_{i1} \text{ and } x^2 \text{ is } M^j_{i2} \dots \text{and } x^n \text{ is } M^j_{in} \text{ then } f \text{ is } C_j$$

where x^n are *n* input variables; M^j_{in} are *j* fuzzy sets/MFs (antecedents), *f* is the output of ANFIS network, and C_j is the consequence of the *j*th rule. The aggregated output of all fuzzy rules can be given by:

$$f = \frac{\sum_{j=1}^{m^n} w_j C_j}{\sum_{j=1}^{m^n} w_j} \quad (1)$$

$$\bar{w}_j = \frac{w_j}{w_1 + \dots + w_{m^n}} \quad (2)$$

where w_j is firing strength of fuzzy rules. As shown in Figure-1, it is a five layer network: Layer 1 computes the MF $M^j_{in}(x^n)$. Layer 2 computes the firing strength w_j of each rule in fuzzy rule-base. Layer 3 normalizes the firing strength of each rule (2). Layer 4 determines the consequent part of each rule $\bar{w}_j C_j$. Lastly, Layer 5 aggregates consequents of rules $\sum \bar{w}_j C_j$.

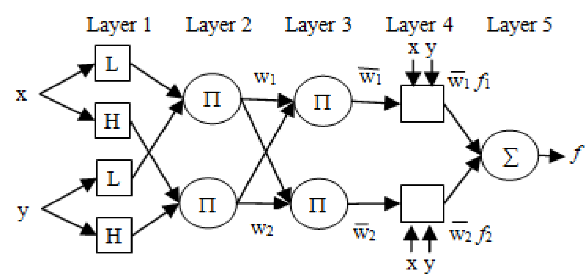

Figure-1. ANFIS architecture (Ishibuchi and Nojima, 2009).

Since, ANFIS is a data driven technique, therefore clustering procedures are upfront methods to the synthesis of ANFIS networks. These include clustering input data, output data, or joint input-output data. The choice depends on the way ANFIS rules are built (Panella and Gallo, 2005). The technique discussed next is based on clustering input-output dataspace. It is intended to improve the ANFIS accuracy with optimum rules by estimating the hyperplanes associated with the consequent parts of Sugeno first order rules.

HYPERPLANE CLUSTERING FOR ANFIS SYNTHESIS

Panella and Gallo (2005) proposed Optimized Hyperplane Clustering Synthesis (OHCS) for obtaining optimal number of rules in ANFIS network with high accuracy. For determining MF, they used fuzzy Min-Max classification on the input dataspace. In their proposed technique, ANFIS output is approximated by hyperplanes where each corresponds to an input-output cluster – representing a rule:

$$y_t = \sum_{j=1}^{N} a_j^{(k)} x_{tj} + a_0^{(k)}, k = 1, \dots, M \quad (3)$$

where N is the optimal number of rules. Here, the coefficients $a_j^{(k)}, j = 0, \dots, N$ of the linear consequent of the corresponding *k*th rule are determined by corresponding *k*th cluster. The step by step process, proposed by Panella and Gallo (2005), of hyperplane clustering in the joint input-output dataspace is as follows:

- **Initialization:** Given a value of M, the coefficients of each hyperplane are initialized randomly. Successively, each training pair $(x_t, y_t), t = 1, \dots, P$, is assigned to a hyperplane $A_q, 1 \le q \le M$, based on the procedure mentioned in Step 2.
- **Step 1:** The pair assigned to each hyperplane is used to update the coefficients of it. Following linear equation has to be solved for kth hyperplane:

$$y_t = \sum_{j=1}^{N} a_j^{(k)} x_{tj} + a_0^{(k)} \quad (4)$$

where *t* is index of all training pair assigned to the *k*th hyperplane. Any least squares technique can be used to solve (4).

- **Step 2:** Each training pair is assigned to hyperplane/cluster *k* with minimum orthogonal distance from output y_i.

$$d_t = \left| \frac{y_t - \left(\sum_{j=1}^{N} a_j^{(k)} x_{tj} + a_0^{(k)}\right)}{\sqrt{1 + \sum_{j=1}^{N} \left(a_j^{(k)}\right)^2}} \right| \quad (5)$$

- **Stopping criterion:** Has the overall error tolerance (6) reached then stop, otherwise go to Step 1.

$$E = \frac{1}{P}\sum_{i=1}^{P} e_i \tag{6}$$

After determining linear coefficients of the consequent part of the fuzzy rules, it is time to decide about MFs. However, it is not easy since the training patterns of hyperplanes may overlap the ones in input space. To avoid this, Panella and Gallo (2005) proposed the use of Adaptive Resolution Classifier (ARC) algorithm which is basically the Min-Max classifier. This algorithm is used in combination with hyperplane clustering to find the suitable input MFs with the help of hyperboxes (HBs). These HBs cover the training patterns such that $H_1^{(q)}, H_2^{(q)}, \ldots, H_R^{(q)}$ are the HBs associated with a class label q or one of the ANFIS rules, and their related MFs are $\mu_1^{(q)}(x), \mu_2^{(q)}(x), \ldots, \mu_R^{(q)}(x)$:

$$\mu_R^{(q)}(x) = max[\mu_1^{(q)}(x), \mu_2^{(q)}(x), \ldots, \mu_R^{(q)}(x)] \tag{7}$$

where max of HBs $\mu_R^{(q)}(x)$ is taken because each HB will represent one of many clusters related to the input space of the same hyperplane.

The procedure of clustering and determining the MFs mentioned above is termed as Hyperplane Clustering Synthesis (HPC) algorithm (Panella and Gallo, 2005). Thus, Optimized HCS (OHC) is used to obtain the ANFIS network with optimal number of rules with high accuracy by choosing the optimal value of M hyperplanes. It is done via the basic neural network learning theory where the minimum value of cost function is achieved:

$$F(M, A_o) = (1 - \lambda)\frac{E(M, A_o) - E_{min}}{E_{max} - E_{min}} + \lambda\frac{M}{P} \tag{8}$$

where $F(M, A_o)$ is cost function of given value of M and initialization value of A_o; E_{max} and E_{min} are maximum and minimum values of E, respectively, for multiple values of M and A_o; λ is weight between [0, 1].

The performance of this method, OHCS for ANFIS synthesis with optimal number of rules with high accuracy, was validated on various benchmark and real-world problems. According to (Panella and Gallo, 2005), further development of HCS or OHCS will result in better ANFIS rule-base optimization.

The major drawback of above mentioned technique is multiple initialization of *M* clusters. The lower number of initializations causes the lower probability of optimal ANFIS, while increasing it, increases computational cost (Panella, 2012). In order to solve this problem, Panella (2012) proposed Hierarchical HCS (HHCS). Here, HHCS starts with the initialization of only one hyperplane *M* since each training pattern belongs to one cluster. Then an iterative procedure of hierarchical construction of hyperplanes starts.

As illustrated in Figure-2, the procedure starts by initializing *M* hyperplanes and executing HCS algorithms on ANFIS with *M* hyperplanes. Then, an optional tuning of obtained ANFIS parameters can be performed. If maximum number of rules M_{max} is reached, then the iteration stops and the ANFIS with best cost function (8) is chosen. If $M < M_{max}$, the hyperplane having worst cost function is split into two new clusters/hyperplanes and the old one is removed. Subsequently, the iteration starts again with ANFIS having $M + 1$ hyperplanes/rules. As per Panella (2012), the performance of the resulting ANFIS is better than the previous related approaches in literature.

One of the popular clustering algorithms is Subtractive Clustering. Here, clustering strategy is based on input dataspace only. The next section gives brief introduction to this technique.

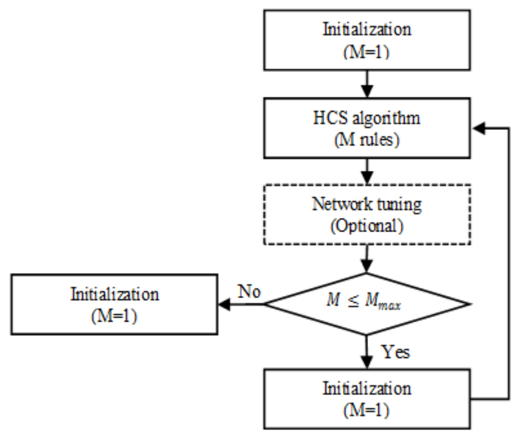

Figure-2. Flow chart of HHCS algorithm (Panella and Gallo, 2005).

SUBTRACTIVE CLUSTERING

To provide sufficient data for rule-base generation through ANFIS, a large amount of input-output data is needed (Zarandi *et al.*, 2007). That data needs to be grouped into multiple clusters. The famous clustering techniques applied while developing ANFIS networks, include grid partitioning and subtractive clustering (Kaur and Klair, 2012). Subtractive clustering (SC) is one of the fuzzy clustering methods in which rules are derived by grouping input dataspace (Yazdani-Chamzini *et al.*, 2013). It was first introduced by Chiu (1994) and is a fast one-pass algorithm for determining clusters and their centers in dataspace (Bezdek, 1981; Chiu, 1994). Here, the best optimum rule-base for ANFIS can be obtained by efficiently estimating the cluster centers. Each rule is represented by a cluster and it determines antecedent part of the rule. The consequent part is simple linear equation which can be tuned by any least square method. The subtractive clustering works as follows.

Here, each data point is supposed to be a potential cluster center to all other points. We calculate its potentiality measure for data point as:

VOL. 10, NO. 20, NOVEMBER 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

$$P_i = \sum_{j=1}^{n} e^{-\alpha \|x_i - x_j\|^2} \quad (9)$$

Where

$$\alpha = \frac{4}{r_\alpha^2} \quad (10)$$

And

P_i is the potential value for cluster center,

α is the weight between points x_i and x_j,

r_α is the positive constant for cluster radius,

$\|.\|$ is the Euclidean distance.

The higher is the number neighboring data points, the higher is the potential of a data point. The cluster radius r_α defines the neighborhood. The data point with highest potential P_1^* is taken as a first cluster center. The potential of the rest of the data points is calculated thereafter, as follows:

$$P_i \Leftarrow P_i - P_k^* e^{-\beta \|x_i - x_j^*\|^2}, k = 1,\dots,N \quad (11)$$

Where

$$\beta = \frac{4}{r_\beta^2} \quad (12)$$

and

β is the weight of i data point to cluster center,

r_β is the positive constant for cluster radius; greater than r_α to avoid closely distanced cluster centers,

x_j^* is the location of kth cluster center,

P_k^* is the potential value of cluster center x_j^*,

N is the number of total cluster centers.

Again, the data point with highest potential is considered as next cluster center. Once, the *k*th cluster center has been obtained, the potential of each data point is revised by (11). The process of obtaining new cluster center and calculating their potentials repeats until the remaining potential of all data points fall below some fraction of the first cluster center P_1^*.

The clusters found above, representing groups of similar data in input dataspace, are mapped to the related class in output dataspace. Thus, each cluster center represents a rule for identifying the related class:

$$\text{Rule } i = \text{If}(x \text{ is near } x_i^*) \text{ then class is } C_i \quad (13)$$

$$\mu_i = e^{-\alpha \|x - x_i^*\|^2} \quad (14)$$

where (14) defines the membership degree of data point x with the cluster center x_i^* and α is a constant defined by (10). In the form of MF, the above rule can rewritten as:

$$\text{Rule } i = \text{If } X_1 \text{ is } A_{i1} \text{ then class is } C_i \quad (15)$$

where X_1 is input variable and A_{i1} is the membership function in the ith rule.

Eftekhari and Katebi (2008) used Genetic Algorithm (GA) to find suitable cluster centers in subtractive clustering in order to develop ANFIS structure with optimum rule-set. Chen (2013) also proposed integration of metaheuristic algorithm Particle Swarm Optimization (PSO) with subtractive clustering for obtaining optimum rule-base with high accuracy.

Other than clustering methods for the synthesis of ANFIS with optimum rules-set, few researchers have also proposed techniques which are used to minimized knowledge base without compromising on accuracy. Following methods are one of those.

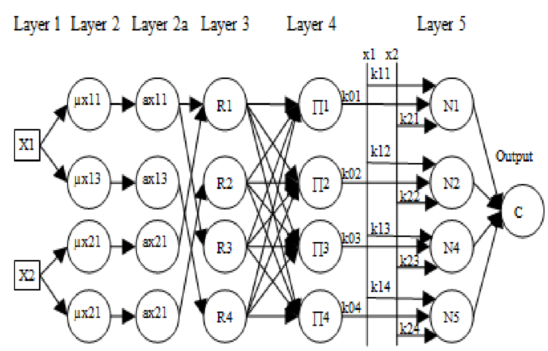

Figure-3. Modified ANFIS architecture (Rini, Shamsuddin, and Yuhaniz, 2013).

SELECTING AND PRUNING RULES

Many real-world optimization problems involve several conflicting objectives, such as accuracy and interpretability (Rini, Shamsuddin, and Yuhaniz, 2014). These two contradictory problems are also faced by ANFIS, while simultaneous optimization of both the aspects has been a trade-off problem (Rini *et al.*, 2013). The main purpose of an optimized ANFIS is modeling a real-world problem with high interpretability and maximum accuracy (Rini *et al.*, 2014). These two objectives are represented through equation (16) for accuracy and equation (17) for interpretability (Rini *et al.*, 2013).

$$f_1 = \min\left\{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2\right\} \quad (16)$$

where y_j and $\hat{y}_j$ are the actual and the desired output, respectively, and n is the number of data samples.

$$f_2 = \min\{\sum_{r=1}^{r} O_r\} \quad (17)$$

where r is the number of all possible rules in the ANFIS rule-base, and $O_r \in Q$ is a binary value used to indicate whether the rule node r is selected or not.

Interpretability refers to structure of ANFIS while accuracy refers to the ability of the network to closely

VOL. 10, NO. 20, NOVEMBER 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

resemble the response of desired model. The structure includes number of inputs, number of rules in the entire rule-base, the number and the shape of MFs. The structure influences the complexity and the computational cost of a system. Thus, optimizing the ANFIS rule-base would serve as reducing the network complexity and its computational cost. This can be done by pruning less important rules and selecting the most effective ones only (Rini *et al.*, 2013). This optimized rule-base should satisfy accuracy demand. Although, it can be further improved by tuning MFs. The following mentioned researchers have tried to meet both the requirements; accuracy maximization and complexity minimization, while optimizing the ANFIS network.

Rini *et al.* (2013) have optimized ANFIS for its learning through tuning MFs and finding the optimal rule-set by using PSO so that they could stabilize accuracy and interpretability trade-off problem in ANFIS modeling. In this approach of ANFIS optimization, an ANFIS is considered as one particle in PSO. The dimensions of the particle are denoted by ANFIS parameters for MF tuning. Simultaneously, the growing and pruning of the number of ANFIS rules is also done. Each particle or ANFIS process in the swarm of PSO would complete to achieve objective function value. The resulting optimal solution in PSO represented the optimized ANFIS. Figure-3 is the modified layered architecture of ANFIS by Rini *et al.* (2013): Layer 1 and 2 are the same as standard ANFIS architecture, though each node in Layer 2 is connected with each node in Layer 2a which represents the modified MFs. Layer 2a is used to tune the MF so that error measure between actual and the desired output could be minimized. Layer 3 and 4 are rule layer and normalization layer, respectively, just like in usual ANFIS network. But, in Layer 4 only the rules which have importance are selected here. Layer 5 is defuzzification layer which contain only the optimized number of rules. The proposed algorithm by Rini *et al.* (2013) is illustrated below. It shows how PSO is utilized in integration with ANFIS to minimize the number of rules and tune MF as well:

1. Initialize particle position and velocity with d number of dimensions;
2. Initialize fitness function (f_1) for PSO-ANFIS. Fitness function of PSO-ANFIS is the objective function of the ANFIS i.e. (16) and (17);
3. Find objective function of ANFIS using (16) and (17). Based on the fitness function, find particle's personal best position in each local best. If fitness is better than current personal best value then assign fitness value to the current personal best;
4. Find best value of global best. Set best of personal values as global best;
5. Update velocity and position of particles;
6. For each particle, find new fitness function. Check error function as fitness function based on Step 3 and find the global best value based on Step 4;
7. Check whether the value has converged then stop, otherwise go back to Step 5. Check if global best is better than stopping criteria then stop, else goto Step

Figure-4. Interpretability vs. accuracy in fuzzy system (Chen, 2013).

While validating the proposed approach, Rini *et al.* (2013) performed experiments on 4 UCI machine learning datasets. They noticed that the number of inputs and data samples help in finding optimal number of rules. They concluded via their research that the complexity of ANFIS network increases by the increase in the number of its rules. Thus, optimal number of rules reduces computational cost. The proposed approach of ANFIS rule-base optimization simultaneously enhanced the accuracy and reduced the complexity based on interpretability. Figure-4 comprehensively illustrates ANFIS accuracy and interpretability trade-off problem.

Based on Figure-4, it is implicit that when optimizing ANFIS rule-base, meeting both the aspects (high accuracy and high interpretability) is a tough job. In search of satisfaction of one aspect may compel to compromise to the other. Thus, as according to Rini *et al.* (2013), the optimization algorithm plays vital role here for balancing these two criteria of modeling any fuzzy inference system.

Rini *et al.* (2014) used PSO for achieving optimal number of rules in ANFIS architecture but they also modified linguistic hedges and put threshold on rules' firing strength. Just like the proposed method by Rini *et al.* (2013), they also used PSO to tune membership functions for maximizing accuracy. The layers architecture of ANFIS is also the same as in Rini *et al.* (2013). In this method, the strong rules are selected from all possible rules $\mathcal{R} = \{R_i, i = 1, ..., N_R\}$. The selected subset of rules, denoted as $\{O_j, j \in \{1, ..., N_R\}\}$, are those which have high accuracy. Here, the rules are assigned to the subset of strong rules based on their output: if the antecedent of rule

ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

$r \in R$ is satisfied with a degree exceeding a threshold value Θ, the rule r is enabled, otherwise it is disabled.

$$O = \begin{cases} 0 \; if \; R_i < \Theta \\ 1 \; if \; R_i \geq \Theta \end{cases} \tag{18}$$

For tuning MFs, they used hedges to change the shape of the MFs. This is done to find strong rules in the normalization layer. For better understanding of linguistic hedge (LH), let's assume a membership function $\mu_A(x)$ represents a continuous linguistic term for input variable x. For example, a modified linguistic term for input variable x says "Student understands math very well". This linguistic term, modified by hedge, can be expressed as:

$\mu_A^p(x)$= Student understands math [very (*LH*)] well

where $p > 0$ changes the meaning of linguistic term. Table 3 lists some popular LH and Figure 5 illustrates the shapes of modifiers.

Table-1. Popular linguistic hedges according to values of *P* (Chen, 2013).

Value of p	Hedge	Effect
0.25	Slightly	Dilation
0.50	More or less	
0.75	Minus	
1.00	-	-
1.25	Plus	Concentration
1.50	More	
1.75	Much more	
2	Very	
4	Absolutely	

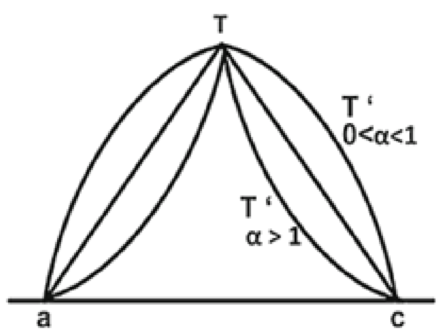

Figure-5. Linguistic hedge modifies basic membership function.

Here, ANFIS represents a particle *X* in PSO which has objective functions to satisfy equation (16) and (17).

$$X = \{p_{ji} | i \in I, j \in J\} \tag{19}$$

where p is the number of LH parameters of each particle, *I* is the membership function, and *J* is input variables to the ANFIS. Equation (20) is equation where *k* denotes consequent parameters and *r* represents the rule-set in a particle of PSO-ANFIS. The optimal number of rules are represented by equation (21). Collectively, each particle of PSO-ANFIS can be represented as equation (22)

$$X = \{k_{rj} | j \in J, r \in R\} \tag{20}$$

$$X = \{O_r | r \in R\} \tag{21}$$

$$X = \{p_{ji}, k_{rj}, O_r | i \in I, j \in J, r \in R\} \tag{22}$$

To validate the performance of the proposed model, Rini *et al.* (2014) executed tests on 6 datasets from the repositories of UCI machine learning and KEEL. They concluded that the proposed technique provides promising results in terms of better interpretability and acceptable accuracy. The researchers also foresee further improvement in this technique in future.

RESULTS AND ANALYSIS

The performances of ANFIS networks synthesized by the techniques discusses above are validated by several simulation tests. In this section, some of the significant results are illustrated which summarize the performance of ANFIS models with lower number of rules. The rule-base optimization methods have been compared in terms of optimized rule-set, approximation accuracy, and computational time to determine optimal technique. These quantities are represented by optimized rule-set, mean square error (MSE), and accuracy percentage. The computational cost is determined by the number of rules. The more rules in an ANFIS network, the more it takes to compute its output.

All of the methods illustrated in Table-2 are used to model function approximation problems containing 3 inputs and 1 output. The ANFIS networks generated by HHCS and OHCS (Sec. III) are used to model following 3-input non-linear function:

$$f(x) = 1 + x_1^{0.5} + x_2^{-1} + x_3^{-1.5} \tag{23}$$

The above function is also modeled using resilient propagation (RPROP) in combination with recursive least square error (RLSE), and gradient descent (GD) joined to RLSE approaches. Whereas, ANFIS networks separately generated by subtractive clustering (Sec. IV) and the approach of selecting and pruning rules (Sec. V) are used to model real-world benchmark problem of Haberman's which also contains 3 inputs and 1 output.

Table-2. Results of ANFIS rule-base optimization methods.

Rule-Base Optimization Method	Optimized Rule-set	Training MSE	Testing MSE	Accuracy %
HHCS	3	0.00048	0.0079	99.976
OHCS	4	0.00014	0.0127	99.993
RPROP+RLSE	9	0.00001	0.0474	99.999
GD+RLSE	9	0.00086	0.0669	99.957
ANFIS+SC	9	0.03836	0.0478	98.082
ANFIS+PSO	16	0.16300	0.1950	91.850

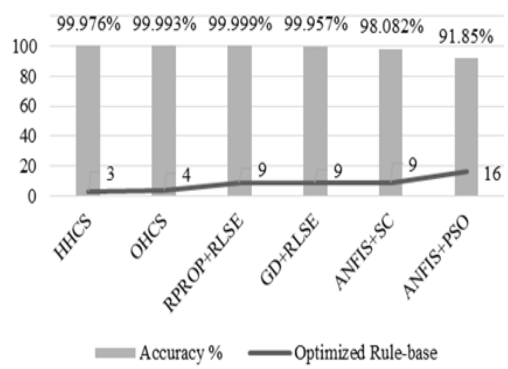

Figure-6. Analysis of accuracy of ANFIS with optimized rule-set.

According to the results presented in Table-2, the approach of HHCS proved to be the best for the synthesis ANFIS network. This method, optimized ANFIS rule-base upto 3 rules only with maximum acceptable accuracy which is 99.976%. The optimized rule-base reported in literature contains 4 rules in case of OHCS with accuracy of 99.993%.

The gradient based techniques are also popular in literature for the optimization of ANFIS networks. Thus, these methods have also been run into comparison with the ones analyzed in this paper. Although, PROP+RLSE and GD+RLSE result in competing accuracy but HHCS and OHCS achieve it with fewer rules. In case of benchmark problems, SC generated ANFIS with less number of rules and also brought better accuracy than the approach of selecting and pruning potential rules using PSO. The overall picture of performance of rule-base optimization techniques, discussed in sections III-V, is depicted in Figure-6. It clearly shows that HHCS achieves better generalization capability and accuracy of ANFIS network with fewest rules.

CONCLUSIONS

Based on study and analysis of research, covered in this paper, we can conclude that there exist two major bottlenecks in the implementation of ANFIS based models. These are rule-base minimization and accuracy maximization. Various approaches or techniques have been proposed in literature which try to simultaneously achieve rule-base minimization and accuracy maximization. Some of these use clustering of input-out data, input data or output data only, while the others are selecting and removing potential and non-potential rules from the entire ANFIS knowledge-base.

While analyzing previous research, it can be concluded that clustering techniques have been more effective in overcoming the above mentioned bottleneck issues. An efficient clustering technique not only helps in modeling membership functions but also optimizes the number of rules. Since, the rule-set is already minimized, there will be reduced number of consequent parameters. This means, reduced effort required to train these parameters.

The results presented in this research indicate the robustness of clustering techinques HHCS and OHCS over other rule-base optimization techniques. Despite of issues in clustering algorithms, this approach has the potential to be explored and improved further for the synthesis of ANFIS networks that show better accuracy with minimum number of rules. However, it is so important to keep balance between complexity minimization and accuracy maximization. The findings also indicate the utilization of metaheuristic algorithm could be efficiently integrated with clustering procedures to best group dataspace. This would lead to construct ANFIS network with best rule-set having better generalization capability.

REFERENCES

[1] Abiyev, Rahib, Mamedov, Fakhreddin, and Al-shanableh, Tayseer. 2007. Nonlinear Neuro-fuzzy Network for Channel Equalization Analysis and Design of Intelligent Systems using Soft Computing Techniques, pp. 327-336, Springer.

[2] Bezdek James C. 1981. Pattern recognition with fuzzy objective function algorithms: Kluwer Academic Publishers.

[3] Chen M. Y. 2013. A hybrid ANFIS model for business failure prediction utilizing particle swarm optimization and subtractive clustering. Information Sciences, 220, pp.180-195.

[4] Chiu Stephen L. 1994. Fuzzy model identification based on cluster estimation. Journal of intelligent and Fuzzy systems, Vol. 2, No. 3, pp. 267-278.

[5] Eftekhari M. and Katebi SD. 2008. Extracting compact fuzzy rules for nonlinear system modeling using subtractive clustering, GA and unscented filter. Applied Mathematical Modelling, Vol. 32, No. 12, pp.2634-2651.

[6] Gorzalczany Marian B. 2001. Computational intelligence systems and applications: neuro-fuzzy and fuzzy neural synergisms, Vol. 86, Springer.

[7] Ishibuchi Hisao, and Nojima, Yusuke. 2009. Multiobjective genetic fuzzy systems Computational Intelligence, pp. 131-173, Springer.

[8] Jang J. S. R. 1993. ANFIS: adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man and Cybernetics, Vol. 23, No. 3, pp.665-685.

[9] Kar S., Das S. and Ghosh P. K. 2014. Applications of neuro fuzzy systems: A brief review and future outline. Applied Soft Computing, Vol. 15, pp.243-259.

[10] Kaur Raminder Preet and Klair, Amanjot Singh. 2012. Investigation of Grid partitioning and Subtractive Clustering based Neuro-Fuzzy Systems for Evaluation of Fault Proneness in Open source software system. International Conference on Computer Graphics, Simulation and Modeling, pp.143-145.

[11] Liu P., Leng W. and Fang W. 2013. Training anfis model with an improved quantum-behaved particle swarm optimization algorithm. Mathematical Problems in Engineering.

[12] Neshat, Mehdi, Adeli, Ali, Sepidnam, Ghodrat and Sargolzaei, Mehdi. 2012. Predication of concrete mix design using adaptive neural fuzzy inference systems and fuzzy inference systems. The International Journal of Advanced Manufacturing Technology, Vol. 63, No. 1-4, pp. 373-390.

[13] Panella Massimo. (012. A hierarchical procedure for the synthesis of ANFIS networks. Advances in Fuzzy Systems, 20.

[14] Panella Massimo and Gallo Antonio Stanislao. (2005). An input-output clustering approach to the synthesis of ANFIS networks. IEEE Transactions on Fuzzy Systems, Vol. 13, No. 1, pp. 69-81.

[15] Rini D. P., Shamsuddin S. M. and Yuhaniz S. S. 2013. Balanced the Trade-offs problem of ANFIS Using Particle Swarm Optimisation. TELKOMNIKA Telecommunication, Computing, Electronics and Control, Vol. 11, No. 3, pp. 611-616.

[16] Rini D. P., Shamsuddin S. M. and Yuhaniz S. S. 2014. Particle swarm optimization for ANFIS interpretability and accuracy. Soft Computing

[17] Taylan O. and Karagözoğlu B. 2009. An adaptive neuro-fuzzy model for prediction of student's academic performance. Computers & Industrial Engineering, Vol. 57, No. 3, pp. 732-741.

[18] Teshnehlab, Mohammad, Shoorehdeli, Mahdi Aliyari, and Sedigh, Ali Khaki. 2008. Novel hybrid learning algorithms for tuning ANFIS parameters as an identifier using fuzzy PSO. ICNSC 2008 IEEE International Conference on Networking, Sensing and Control.

[19] Yazdani-Chamzini, Abdolreza, Razani, Mojtaba, Yakhchali, Siamak Haji, Zavadskas, Edmundas Kazimieras, and Turskis, Zenonas. 2013. Developing a fuzzy model based on subtractive clustering for road header performance prediction. Automation in Construction, Vol. 35, pp. 111-120.

[20] Zarandi, Mohammad Hossein Fazel, Alaeddini, Adel, Turksen, I Burhan, and Ghazanfari, Mahdi. 2007. A neuro-fuzzy multi-objective design of Shewhart control charts Analysis and Design of Intelligent Systems using Soft Computing Techniques, pp. 842-852, Springer.