Comparison of Neuro-Fuzzy Systems with a Defuzzification-Based Algorithm for Learning Fuzzy Rules

Jean J. Saade and Adel Fakih

ECE Department, FEA, American University of Beirut,
P.O.Box: 11-0236, Riad El Solh 1107 2020, Beirut, Lebanon
E-mail: jsaade@aub.edu.lb

Abstract. This study deals in its first stage with some neuro-fuzzy algorithms used to learn fuzzy inference systems. Two categories of neuro-fuzzy learning approaches are described and compared. The first contains the conventional learning approaches, which were developed by Ichihashi, Nomura, Wang and Mendel. The second consists of another method developed by Shi and Mizumoto. The comparison is based on practical properties related to structure and parameters learning. Then, the drawn conclusions and the mentioned properties are used to provide a comparison between the considered neuro-fuzzy methods and a developed defuzzification-based learning algorithm for fuzzy systems. The advantages of this algorithm over the neuro-fuzzy ones are clearly emphasized.

1 Introduction

Due to the importance of fuzzy inference systems in the linguistic representation of human knowledge and expertise, the design of these systems has been given a great deal of attention in the literature. Design methods using data-driven neuro-fuzzy learning approaches, where a neural network learning procedure is used to identify Takagi-Sugeno-Kang (TSK) fuzzy model parameters, have been devised [1-7].

This study addresses first two categories of neuro-fuzzy learning approaches: The conventional ones and another new approach [1-6]. An overview of these learning methods is given and then they are compared using structure and learning-related properties. Based on the comparison results and the mentioned properties, a defuzzification-based learning algorithm for fuzzy systems [8-10] is brought into picture and compared with the considered neuro-fuzzy methods. The advantages of this algorithm as they relate to practically important properties; such as the simplicity of setting the initial fuzzy system, the avoidance of non-firing states, linguistic interpretability, etc., are emphasized.

2 TSK Fuzzy Inference Models

In a TSK fuzzy system of zero order [7], the antecedent part of each rule is composed of linguistic variables and the consequent is a crisp value. Hence, in a system with p inputs, x_j, j=1,2, …, p, and one output, y, the rth rule, $1 \le r \le k$, is expressed as follows:

$$R_r : IF\ x_1\ is\ A_{1r}\ and\ x_2\ is\ A_{2r}\ and\ \ldots and\ x_p\ is\ A_{pr}, THEN\ y\ is\ y_r. \qquad (1)$$

© A. Gelbukh, R. Monroy. (Eds.)
Advances in Artificial Intelligence Theory
Research on Computing Science 16, 2005, pp. 163-172

In (1), A_{jr} are the rth rule fuzzy sets assigned respectively over the input variables x_j and y_r is the rth rule crisp consequent. Based on the rules structure, the number of membership functions (MF's) on each input variable is equal to the number of rules and each MF on an input variable participates in only one rule.

The output value corresponding to input vector $\underline{x}_i = (x_{1i}, x_{2i}, \ldots, x_{pi})$ is computed using a weighted average formula as follows:

$$y_i = \sum_{r=1}^{k} h_{ri} y_r \Big/ \sum_{r=1}^{k} h_{ri} . \tag{2}$$

The use of product for "and," which is applied in neuro-fuzzy methods, gives the firing strength of rule r expressed as:

$$h_{ri} = \prod_{j=1}^{p} A_{jr}(x_{ji}) . \tag{3}$$

3 Neuro-Fuzzy Learning Methods

Two main neuro-fuzzy learning approaches are of interest here: the conventional one developed in [1-4] and the new approach [6].

3.1 Conventional Neuro-Fuzzy Methods

Referring back to Section 2, we note here that Eq. (2) was used by Wang and Mendel [4] and Nomura [3]. Ichihashi [1,2] used a simplified version of (2) to get the system output:

$$y_i = \sum_{r=1}^{k} h_{ri} y_r . \tag{4}$$

Ichihashi and Wang-Mendel used Gaussian MF's while Nomura used triangular ones.

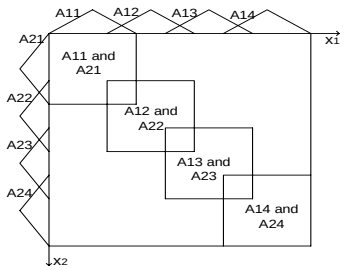

Fig. 1. Conventional neuro-fuzzy system with non-firing states

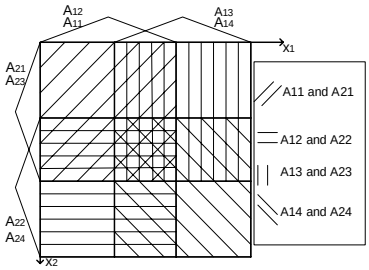

Fig. 2. Conventional neuro-fuzzy system with no non-firing states

To avoid initial non-firing states, the firing strength of at least one rule (See (3)) and for any input $\underline{x}_i$ must differ from zero. Hence, any region in the input space must be covered by all the MF's of at least one rule. This requires a special setting of the initial MF's. For a system with two inputs, say, there has to be as many MF's having the same shape on each input as there are distinct ones. Fig. 1 illustrates the case of a 4-rule and 2-input system where non-firing states exist if $\underline{x}_i$ is anywhere in the regions outside those assigned for the rules. In Fig. 2, however, non-firing states are avoided since the MF's are assigned as required. Undoubtedly, the process of initial MF's and rules assignments gets more difficult when the number of these MF's and rules increases and when the system has more than two inputs (See Section 3.3.4).

When n training input-output data ($\underline{x}_i$, y_{id}), where i = 1,2,…n, are given, then they are used in an error back-propagation learning to modify the parameters of an initial fuzzy inference system, whose form is given in (1), and minimize the data approximation error. The following error function is usually adopted:

$$E_i = (y_{id} - y_i)^2 / 2. \quad (5)$$

The center and width of triangular MF's, the mean and variance of Gaussian ones and the crisp consequents are updated by the gradient-descent method:

$$a(t+1) = a(t) - \alpha[\partial E_i(t) / \partial a] = a(t) + \alpha[y_{id} - y_i(t)]\, \partial y_i(t) / \partial a \ , \quad (6)$$

where α is a learning rate, t denotes the current iteration and a is the parameter of concern .

Once a data pair ($\underline{x}_i$, y_{id}) is presented to the system and the system parameters are updated based on (6), then the system output for the same input $\underline{x}_i$ changes at each update and also the error E_i. The tuning of the system parameters for the input data $\underline{x}_i$ stops when the step size, $d_i = | E_i(t+1) - E_i(t)|$, between two consecutive iterations drops below a given threshold. Then another data pair is presented to the system and the procedure is repeated. When all the data are presented to the system (learning epoch), the total error E is calculated as follows:

$$E = 2\sum_{i=1}^{n} E_i / n = \sum_{i=1}^{n} (y_{id} - y_i)^2 / n \, . \quad (7)$$

If this error is smaller than some desired error, E_d, the learning stops. If not a new learning epoch begins. However, the performance of repeated epochs would not necessarily lead to $E \leq E_d$. Hence, the number of epochs is also be considered as a stopping criterion.

In the above-described type of learning (pattern mode), the tuning of the system by a given data ($\underline{x}_i$, y_{id}) affects the tuning of the system by all the subsequent data points. This effect is absent in the batch learning mode since the parameters are updated only after the whole data set is used. Actually, for a point ($\underline{x}_i$,y_{id}), the adjustment $\Delta_i a = a(t+1) - a(t)$ of a given parameter a is still computed as in (6). But, this adjustment is stored. When all the data pairs have been used, the total adjustment is computed as $\Delta a = 2\sum_{i=1}^{n} \Delta_i a / n$. In fact, batch learning is equivalent to the use of (6) with E_i replaced by E .

3.2 New Neuro-Fuzzy Approach

The major difference between this approach (Shi and Mizumuto [6]) and the conventional ones is that all the combinations of the MF's assigned over the input variables are used to form the antecedents of the rules. This difference along with pattern learning entails modifications in the properties of the algorithm. An overview of the new approach is provided for a system with two inputs.

Let A_{1s}, s = 1,2, …, l_1 and A_{2q}, q = 1,2, …, l_2, be the MF's on input variables x_1 and x_2 respectively. Then, $k = l_1 \times l_2$ fuzzy rules are constructed in the form:

$$\text{Rule } (s-1)l_2 + q\text{: If } x_1 \text{ is } A_{1s} \text{ and } x_2 \text{ is } A_{2q}, \text{ THEN y is } y_{(s-1)l2+q}. \quad (8)$$

With $h_{[(s-1)l2+q]i}=A_{1s}(x_{1i})\times A_{2q}(x_{2i})$ denoting the firing strength of the rule in (8), then the output is calculated as follows:

$$y_i = \sum_{s=1}^{l_1}\sum_{q=1}^{l_2} h_{[(s-1)l_2+q]i} y_{(s-1)l_2+q} \Big/ \sum_{s=1}^{l_1}\sum_{q=1}^{l_2} h_{[(s-1)l_2+q]i} \tag{9}$$

3.3 Properties-Based Comparison

The conventional and new neuro-fuzzy learning approaches are compared here based on properties related to their structure and the applied learning procedure.

3.3.1 Type of Membership Functions

Both conventional and new neuro-fuzzy methods require that the MF's be differentiable with respect to their parameters. This is due to the gradient-descent method (6). Also, any change in the form of the used MF's requires that the parameters updating formulas be rederived.

3.3.2 Type of Logic Operations and Error Function

The use of Eqs. (2)-(4) and (9) in neuro-fuzzy means that the fuzzy AND, OR and THEN are respectively represented by product, sum and product. This is essential for the gradient-descent formula (6), and the involved derivative. Also, the adopted error function influences the parameters updating formulas.

3.3.3 Type of Learning

The considered neuro-fuzzy algorithms, use pattern learning (Section 3.1). Referring to Fig. 2 and (2)-(6) it can be seen that more than one training example affect the same system parameters. Hence, by adjusting these parameters based on a data point and then going to the next, should lead to a compromise between the parameters and their affecting points that is not as good as the one obtained using the batch mode of learning (Section 3.1). In fact, this aspect becomes more serious when more data points affect the same system parameters, as in the new neuro-fuzzy approach (Section 3.2) where the MF's are less localized. Hence, the batch mode of learning should be more suitable for the type of fuzzy inference structure used in the new neuro-fuzzy method.

3.3.4 Setting of Initial MF's and Rules

The major concern in the conventional methods is the avoidance of initial non-firing [5]. Hence, on each system input, there must be a number, d_{mf}, of distinct MF's with each repeated r_{mf} times to give rules antecedents covering the whole input space. As explained in Section 3.1, this is not simple especially when the number of rules, k, and input space dimension, p, increase. To make things easier, two formulas are set. With $d_{mf}\times r_{mf}=k$ and $r_{mf}=d_{mf}\times(p-1)$, then

$$d_{mf} = \sqrt{k/(p-1)} \ \text{and}\ r_{mf} = \sqrt{k(p-1)}\ . \tag{10}$$

Of course, p is application-dependent. Hence, k needs to be chosen to give integer d_{mf} and r_{mf}. In the new neuro-fuzzy approach, the assignment of initial MF's and rules is

simple. As long as the adjacent MF's on each input overlap (Fig. 3), then the coverage of the entire input space is guaranteed and initial non-firing is avoided (Section 3.2).

3.3.5 Simplicity of Learning Formulas

The learning formulas in the conventional and new methods are determined using (6). In the conventional methods, each MF is used once in the rules. This makes the application of (6) with y_i as in (2) or (4) and, thus, the learning formulas simple and easily extended to systems with a high number of inputs as compared to the formulas in the new approach, where each MF on a specific input is used with all the combinations of MF's on the remaining inputs (See (9) and also [6]).

3.3.6 Number of Tuning Parameters

For a given number of rules, $k \geq 2$, and a number of input variables, $p \geq 2$, the number of tuning parameters in the conventional approach , $(2p+1)k$, is greater than that in the new approach, which is given by $2(l_1+l_2+...+l_p)+k$, where l_j is the number of membership functions on input x_j. This can be verified as follows: Since $l_1 \leq k$, $l_2 \leq k, ..., l_p \leq k$, with equalities that cannot be satisfied simultaneously except for: (a) k = 1 for any p, (b) p=1 for any k, then for $k \geq 2$ and $p \geq 2$, $l_1+l_2+...+l_p < pk$. In cases (a) and (b), which rarely occur in practice, the conventional and new neuro-fuzzy approaches have the same number of parameters.

3.3.7 Fitting toTraining Data

A MF in the new neuro-fuzzy method covers a larger area in the input space as compared to the conventional approach (Compare Figs. 2 and 3). Hence, for the same set of data points, the 2 parameters of a MF in the new approach need to be adjusted to accommodate a larger number of data. Consequently, the fitting to training data in the new approach is less precise than that in the conventional one for the same number of rules. Data fitting results are provided in Section 3.3.8.

3.3.8 Linguistic Interpretability

Based on the studies in [11,12], The linguistic interpretability problem in neuro-fuzzy learning relates to the highly overlapping and complex MF's obtained after training. This prevents the simple assignment of linguistic labels to these MF's and leads to the generation of rules lacking a clear linguistic meaning. The issue of tradeoff between precision and interpretability has also been noted in the mentioned studies.

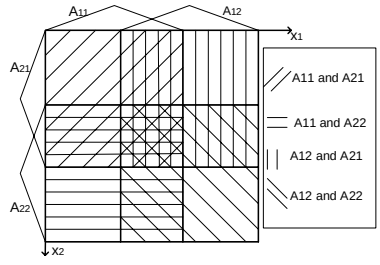

Fig. 3. Initial MF's and rules for the new neuro-fuzzy approach

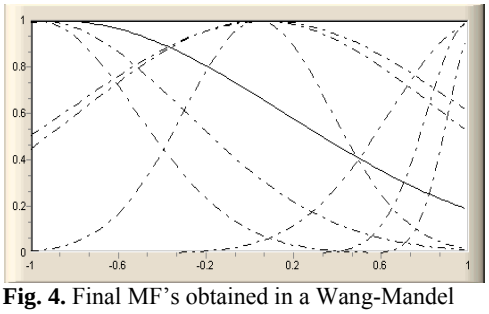

Fig. 4. Final MF's obtained in a Wang-Mandel system

The main reason behind the MF's complexity obtained in the neuro-fuzzy methods relates to the unconstrained learning of the MF's parameters. As can be seen in Figs. 2 and 3, the parameters of the MF's assigned on an input variable are changed based on common data points; i.e., located in overlapping regions of the input space, and also on separate data points. Hence, these MF's tend to pass each other, exchange positions, etc., as shown in Figs. 4 and 5. This would hinder the linguistic interpretability of the final fuzzy system. The use of crisp rules consequents does also contribute to the deterioration of the linguistic interpretability aspect.

Fig. 4 shows the final MF's obtained over input x_1 after training a 9-rule neuro-fuzzy, Wang-Mendel system using 81 data points retrieved from the non-linear function used in [6] and given below. The least data approximation error E=0.000612 was obtained after performing 100 epochs. The MF's interpretability did not improve for a smaller number of epochs.

$$y = (2x_1 + 4x_2 + 0.1)^2 / 37.21 \ , \ -1 \le x_1, x_2 \le 1 \ . \tag{11}$$

Fig. 5 shows the final MF's over input variable x_1 for a 9-rule fuzzy system trained by the new neuro-fuzzy approach and the same data used in Wang-Mendel's method. The least data approximation error E=0.00194 was obtained after performing 16 epochs. After epoch 16, the error value got bigger and no improvement was obtained in the MF's interpretability.

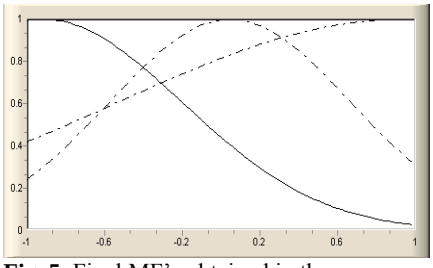

Fig. 5. Final MF's obtained in the new neuro-fuzzy method

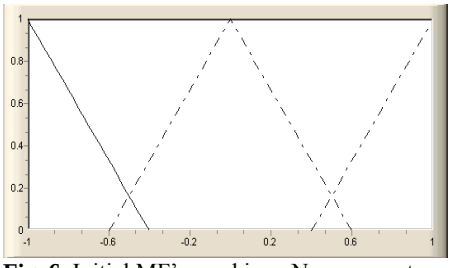

Fig. 6. Initial MF's used in a Nomura system

3.3.9 Firing State Problem

In the considered neuro-fuzzy methods, the learning process changes the parameters of the MF's and even duplicated ones (conventional) become distinct during or after learning (See [5]). Hence, even if the initial MF's are as in Figs. 2 or 3, they may turn out to be similar to those in Fig. 1 or having no overlap between adjacent ones due to the unconstrained learning. This causes non-firing states. Hence, the learning may not complete the specified number of epochs. Fig. 6 shows the initial MF's used on x_1 and x_2 in a 9-rule Nomura system trained using the 81 data points noted in Section 3.3.8. The learning stopped after the second epoch.

4 Defuzzification-Based Learning

A new defuzzification-based algorithm for learning fuzzy rules [8-10] is first summarized here, and then, compared with the considered neuro-fuzzy ones. The comparison is based on the properties addressed in Section 3.3.

4.1 Defuzzification-Based Algorithm

Consider a two-input, one-output fuzzy inference system. Let A_{1s}, s = 1,2…l_1, and A_{2q}, q = 1,2,…,l_2 be overlapping MF's assigned on input variables x_1 and x_2 respectively and in a manner that the specified ranges of these variables are covered. Then, $k=l_1 \times l_2$ fuzzy rules are constructed as in (8) but with $y_{(s-1)l2+q}$ replaced by overlapping MF's assigned on the output variable y and denoted as $C_{(s-1)l2+q}$ for $1 \leq [(s-1)l_2+q] \leq l_1 \times l_2$. These MF's do not need to be all distinct but they have to cover the entire specified range of the output variable.

The fuzzy output, corresponding to a crisp input pair $\underline{x}_i=(x_{1i}, x_{2i})$, is obtained using the CRI [13]:

$$C_{0i}(y) = \max_{1 \leq [(s-1)l_2+q] \leq l_1 \times l_2} [A_{1s}(x_{1i}) \wedge A_{2q}(x_{2i}) \wedge C_{(s-1)l_2+q}(y)] \quad . \tag{12}$$

The fuzzy OR, AND and THEN are represented here by maximum, minimum and minimum respectively. Other operations can be used as well and (12) can be generalized easily to systems with higher dimensional input spaces. Now, defuzzification applies to the normalized version of $C_{0i}(y)$, denoted $C_{0in}(y)$, as [8-10]:

$$F_{\delta}[C_{0in}(y)] = \int_0^1 [\delta c_1(\alpha) + (1-\delta) c_2(\alpha)] d\alpha \quad . \tag{13}$$

$[c_1(\alpha),c_2(\alpha)]$ is the α-level set of $C_{0in}(y)$ and δ is a parameter whose values are in [0,1]. Eq.(13) is used to train initial fuzzy systems based on input-output data. All initial rules consequents are required to be equal to the left-most output fuzzy set, which is to be formed by a flat and a decreasing part or a decreasing part only.

Given the training input-output data ($\underline{x}_i$, y_{id}), with $\underline{x}_i = (x_{1i}, x_{2i}, \ldots, x_{pi})$, and $\underline{x}_i$, y_{id} being within the specified input and output ranges, the learning starts with an initial fuzzy system as specified above. The algorithm computes the fuzzy outputs for all $\underline{x}_i$'s using (12) and then defuzzifies their normalized versions using (13) with δ=1. Here, due to the above-noted initial rules consequents, all the defuzzified values will be equal to the smallest value of the output range. Hence, $F_1[C_{0in}(y)] \leq y_{id}$ for all i=1, 2, …, n. For these defuzzified values, the total error E is computed using some error function, which could be as in (7) or any other function, and compared with a desired error E_d. If $E \leq E_d$, then the learning stops. Otherwise, δ is decreased from 1 to 0 passing by discrete values. For each δ, the error is computed and compared with E_d. The decrease in δ causes an increase in the defuzzified values. They are then made closer to the desired outputs. Whether the change in δ satisfies the error goal, then the learning stops. Otherwise, the algorithm starts another learning round (or epoch) from δ = 1 but with new rules.

These new rules are obtained by raising each rule consequent by one fuzzy set. If this leads to a violation of $F_1[C_{0in}(y)] \leq y_{id}$, it can be reestablished by repeatedly lowering the consequents of the rules triggering one fuzzy output with defuzzified value greater than its desired counterpart. Once the inequality is reinstated, then the decrease in δ is repeated and the error is computed and compared with E_d . This process is repeated until either the error goal is satisfied or no more raise in the rules consequents is

possible or when the raise and lowering of the rules consequents result in a previously obtained system. When the learning ends, the algorithm delivers the final fuzzy system, the resulting error and the final δ value. A complete description and justification of the learning steps in this algorithm was offered in [9].

4.2 Properties-Based Comparison with the Neuro-Fuzzy Algorithms

4.2.1 Type of Membership Functions

Unlike the considered neuro-fuzzy learning methods, the defuzzification-based algorithm can accommodate any type of MF's. This is because the learning is based on the use of (12) and (13) with no derivatives involved. Also, changing the form of the MF's does not require new formulas for learning.

4.2.2 Type of Logic Operations and Error Functions

Again, since no derivatives are used, then there is no restriction on the use of operations for AND, OR and THEN as in the considered neuro-fuzzy approaches. Furthermore, since the error function is not differentiated, then any error function; such as, the mean-square error, (7), root mean-square error, mean absolute error, etc., can be used.

4.2.3 Type of Learning

The objective of the learning process applied in the deffuzification-based algorithm is to reduce the total error resulting from the whole data set rather than the point-wise error. So, the type of learning applied here is compatible with batch mode. This is preferable due to the existence of only one parameter and a fixed number of output fuzzy sets from which the choice is made to form a good compromise for all the data points (See Sections 3.1 and 3.3.8).

4.2.4 Setting of Initial MF's and Rules

As explained in Section 4.1, the setting of the rules antecedents is easy and is done in the same way as in the new neuro-fuzzy approach. Hence, with overlapping MF's over each input, initial non-firing is avoided (See Eq. (12) and Section 3.3.4). Further, the initial rules consequents are equal to the left-most of the fuzzy sets assigned over the output. This guarantees that for $\delta=1$ the defuzzified output for any crisp input be equal to the lowest value of the output range. Requiring also that the right-most of the fuzzy sets assigned over the output be formed by a flat and an increasing part or an increasing part only guarantees that no defuzzified output for any input and any $\delta\in[0,1]$ exceeds the highest value of the output range. These can be checked easily by referring to (12), (13). In the considered neuro-fuzzy approaches, however, it is not specified how the initial crisp consequents are assigned. Also, we do not have bounds on the system outputs nor specified values for the range of the output variable.

4.2.5 Simplicity of Learning Formulas

The defuzzification formula (13) is the one used for learning and it applies to the output fuzzy set after it is determined using (12). Hence, the learning formula remains simple even if the dimensionality of the system input or the number of rules increases.

4.2.6 Number of Tuning Parameters

In the defuzzification-based algorithm, there is only one crisp parameter, δ, to be updated. However, if we consider the fuzzy consequents of the rules, which are also, changed, then the total number of parameters is equal to (k+1). This is less than the number of parameters used in the neuro-fuzzy methods (Section 3.3.6).

4.2.7 Fitting to Training Data

The data fitting in the defuzzification-based algorithm is expected to be less precise than that in the considered neuro-fuzzy approaches. This is because the algorithm has a smaller number of parameters.

4.2.8 Linguistic Interpretability

The learning process described in Section 4.1 does not change the initial MF's assigned over the system inputs. Also, the consequents of the rules are selected from specified fuzzy sets over the output variable. Hence, with the input and output fuzzy sets assigned appropriately to permit a simple and clear linguistic labeling, then the generated rules will have a clear linguistic meaning. This serves well the issue of linguistic representation of knowledge but it is at the expense of accuracy as expected (See [11,12]). Data over-fitting, however, hinders the noise insensitivity and the generalization capability of the learning algorithm as shown in [9,14].

A 9-rule fuzzy system, with three triangular MF's on each input (as in Fig.6) and 7 triangular MF's on the output, was trained by the defuzzification-based algorithm. The 81 data noted in Section 3.3.8 were used. The final system had an error E=0.01234 and δ= 0.45.

4.2.9 Firing State Problem

Since the input MF's are not changed by learning, then unlike the considered neuro-fuzzy methods, the problem of non-firing states does not arise during or after learning.

5 Conclusion

This study has first provided a description and comparison between conventional and a new neuro-fuzzy system from the point of view of structure and learning-related properties. Both approaches require differentiable MF's, use fixed logic operations and error function, apply pattern learning and suffer from non-firing states during or after learning and from the lack of good linguistic interpretability. The new neuro-fuzzy approach, however, turned out to have a simpler setting of initial MF's and rules to avoid initial non-firing and smaller number of tuning parameters. Yet, the conventional approach has less complex learning formulas and more precise data fitting.

Then, a defuzzification-based algorithm has been summarized and shown to possess better properties than the considered neuro-fuzzy approaches. It does not require differentiable MF's nor fixed logic operations and error function. Setting the initial MF's and rules is even simpler than that in the new neuro-fuzzy method. The possibility of non-firing during of after learning is eliminated. Also, the algorithm provides completely interpretable Mamdani-type fuzzy systems with rules having fuzzy antecedents and consequents. Further, the algorithm employs batch learning, and its formulas apply easily to higher dimensional input spaces.

Although the algorithm provides less precise data fitting, this is not a disadvantage since it first provides fuzzy systems, which can easily be given a clear linguistic meaning. Besides, the available data in practice are noisy. This makes the reduced precision a needed aspect to improve noise insensitivity and generalization capabilities, as shown in [9,14]. Also, fuzzy modeling becomes more consistent with Zadeh's principle of "tolerance for imprecision" [13].

In fact, performance criteria related to noise insensitivity and generalization capabilities were introduced in [14] and the algorithm was examined and compared with ANFIS [15] using non-linear functions and a practical robot navigation case [14,16]. The performance advantages of the algorithm were demonstrated in these studies. Criteria-based performance comparison should also be done with the considered neuro-fuzzy approaches and also with an advanced method [17] accounting for noisy data.

References

1. Ichihashi, H.: Iterative Fuzzy Modeling and a Hierarchical Network, Proc. 4th IFSA Congr., Brussels, (1992) 49 -52.
2. Ichihashi, H., Turksen, I.B.: A Neuro-Fuzzy Approach to Data Analysis of Pairwise Comparisons," Int. J. Approximate Reasoning, 9 (3), (1993) 227 -248.
3. Nomura, H., Hayashi, I., Wakami, N.: A Self-tuning Method of Fuzzy Control by Descent Method, Proc. IEEE Int. Conf. on Fuzzy Systems, San Diego, 1992, 203- 210.
4. Wang, L.X., Mendel, J.M. :Back-propagation Fuzzy System as Nonlinear Dynamic System Identifiers, Proc. IEEE Int. Conf. on Fuzzy Systems, San Diego, 1992, 1409-1416.
5. Shi, Y., Mizumoto M.: Some Considerations on Conventional Neuro-fuzzy Learning Algorithms by Gradient Descent Method, Fuzzy Sets and Systems, 112, (2000) 51-63.
6. Shi, Y., Mizumoto M.: A New Approach of Neuro-fuzzy Learning Algorithm for Tuning Fuzzy Rules, Fuzzy Sets and Systems, 112, (2000) 99-116.
7. Takagi, T., Sugeno, M.: Fuzzy Identification of Systems and its Application to Modeling and Control, IEEE Trans. Systems, Man and Cybernetics, 15(1), (1985) 116-132.
8. Saade,J.J.: New Algorithm for the Design of Mamdani-type Fuzzy Controllers, Proceedings of EUSFLAT-ESTYLF Joint Conference, Palma, Spain, Sept. 22-25, (1999) 55-58.
9. Saade, J.J.: A Defuzzification-based New Algorithm for the Design of Mamdani-type Fuzzy Controllers, Mathware and Soft Computing, 7, (2000) 159-173.
10. Saade, J.J.: A Unifying Approach to Defuzzification and Comparison of the Outputs of Fuzzy Controllers" IEEE Trans. Fuzzy Systems, 4 (3), (1996) 227-237.
11. Li, Y., Deng, J.M., Wei, M.Y.: Meaning and Precision of Adaptive Fuzzy Systems with Gaussian-type Membership Functions," Fuzzy Sets and Systems, 127, (2002) 85-97.
12. Paiva, R.P. ,Dourado, A.: Interpretability and Learning in Neuro-fuzzy Systems, Fuzzy Sets and Systems, 147, (2004) 17-38.
13. Zadeh, L.A.: Outline of a New Approach to the Analysis of Complex Systems and Decision Processes, IEEE Trans. Systems, Man and Cybernetics, 3(1), (1973) 28-44.
14 Saade, J.J., Al-Khatib, M.: Efficient Representation of Non-linear Functions by Fuzzy Controllers Design Algorithms, 7th IEEE International Conference on Electronics, Circuits and Systems, Dec. 17-19, (2000) 554-557.
15. Jang, J.S.R.: ANFIS: Adaptive-network-based Fuzzy Inference System, IEEE Trans. Systems, Man and Cybernetics, 23, (1993) 665-685.
16. Al-Khatib, M., Saade, J.J.: An Efficient Data-driven Fuzzy Approach to the Motion Planning Problem of a Mobile Robot, Fuzzy Sets and Systems, 134, (2003) 65-82.
17. Shi, Y., Mizumoto, M.: An Improvement of Neuro-fuzzy Learning Algorithm for Tuning Fuzzy Rules, Fuzzy Sets and Systems, 118, (2001) 339-350.