International Journal of Emerging Engineering Research and Technology
Volume 1, Issue 2, December 2013, PP 1-7

Improvement of NeuroFuzzy Systems Using K-means Clustering

Dr. Rafiqul Zaman Khan
Department of Computer Science,
Aligarh Muslim University, Aligarh, (U.P),
INDIA

Md. Haider Khalaf Jabbar
Universities of Misan, Misan, Iraq and
Department of Computer Science,
Aligarh Muslim University, Aligarh, (U.P),
INDIA

Abstract: *A Neuro Fuzzy System (NFS) is one of the most commonly used systems in the real life problems, it has been recognized as a powerful tool which can facilitate the effective development of models by combining information from different sources, such as empirical models, heuristics and data. Neuro-fuzzy models describe systems by means of fuzzy if–then rules represented in a network structure to which learning algorithms known from the area of artificial neural networks can be applied. This paper examines a K-means clustering algorithm to enhance NFS efficient. The enhancement of NFS by using K-means will be shown by running two of experiments on nonlinear functions. The obtained results will show that, the efficiency of NFS with K-means clustering gives better results than original NFS.*

Keywords: *Neuro fuzzy systems; fuzzy logic, neural networks, K-means clustering.*

1. Introduction

Existing fuzzy reasoning techniques suffer from the lack of a definite method to determine membership functions and a learning capability which can be overcome by neural networks (NNs) driven fuzzy reasoning. NNs are used to tune the membership functions of fuzzy systems (FSs) that are employed for controlling equipment. Although fuzzy system has the ability to convert expert knowledge directly into fuzzy rules, it usually takes a lot of time to design and adjust the linguistic labels (fuzzy sets) of the problem. In addition, the tuning of membership functions is a tricky procedure as it sometimes embodies a number of free parameters that must be assigned by an expert [1, 2]. NN techniques can automate this design procedure improving the performance and reducing the computational time. Neural network approach to design of membership functions was proposed in 1989 [1]. The parameters, which define the shape of the membership functions, are modified to reduce error between output of the fuzzy system and supervised data [3, 4, 5].

The resultant combined system for FS and the NN is called Neuro Fuzzy System (NFS). NFS has the advantage of both, FS and NN, and also it overcomes some of the drawbacks of individual approaches, such as black-box of neural networks and the limited learning capability of fuzzy systems [1, 3].

In many situations it is possible to model the initial parameters for the rules base of NS by using the expert's knowledge about the system according to input and output data of the system. The data could be obtained using perfect mathematical modelling from expert's knowledge or from experimental analysis of the system. Since these data can also contain uncertainty, a suitable approach is essential in order to include any information about the system. In such situations, the system identification can be done by clustering technique, which involves grouping of data into clusters, and modelling the input-output relationship with fuzzy IF-THEN rules by using these clusters. This paper examines a K-means clustering algorithm to enhance the NFS efficient.

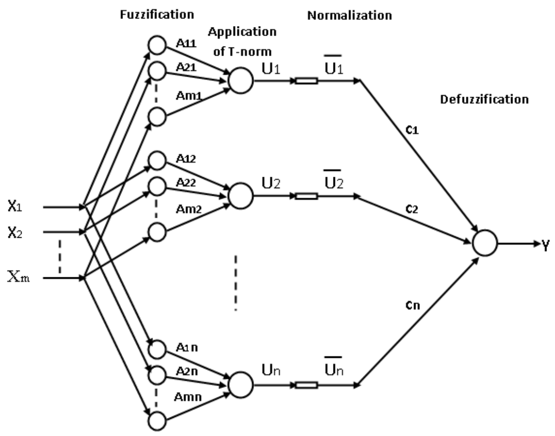


Fig1. *Neuro Fuzzy System Structures*

The rest of this paper is organized as follows. Part II presents architecture and identification algorithm of NFS. Part III describes all details of K-means. In part IV, we present our experimental comparisons. Finally, conclusion and discussion are presented in part V.

2. NEURO FUZZY SYSTEMS (NFS)

The structure of NFS is determined by the functions used to represent the fuzzy sets. A general layout of neuro fuzzy network, which used Mamdani fuzzy model, with multi-inputs and one output, is shown in fig. 1. The architecture of this network is analogous to that of artificial neural network with four-layers [1, 2, 5].

$$A_{ij}\ x_j\ = e^{-\frac{1}{2}\left(\frac{x_j - a_{ij}}{b_{ij}}\right)^2}.$$

In the first layer, the fuzzification operator for each linguistic variable is performed using Gaussian membership functions, which is given as:

$$A_{ij}\ x_j\ = e^{-\frac{1}{2}\left(\frac{x_j - a_{ij}}{b_{ij}}\right)^2}. \qquad (1)$$

where x_j is the j^{th} input variable; a_{ij}, and b_{ij} are the center and the width for the Gaussian membership function.

The outputs of the first layer are fed to the next layer that performs a T-norm operation (product operation) [1]. The output of this layer represents the firing strength of

Premise (antecedent) part for each rule, which could be calculated as following:

$$U_i = \prod_{j=1}^{m} A_{ij}\ . \qquad (2)$$

where i is rule number; and m is the number of input variables.

The firing strength is normalized in the third layer through dividing its value by the summation of all the firing strengths of all rules as:

$$\overline{U}_i = \frac{U_i}{\sum_{L=1}^{n} U_L}\ . \qquad (3)$$

where n is the number of rules.

Finally, in the fourth layer, the summation of all the normalized values of U_i are multiple by the corresponding weight c_i which represents the center of membership function in the consequent-part of the rules [1, 2, 6], to produce the center-of-gravity defuzzification operation. The output of the fourth layer represents the crisp output value for given inputs, which can be obtained by the following formula:

$$Y = \frac{\sum_{i=1}^{n} U_i\ c_i}{\sum_{i=1}^{n} U_i} \qquad (4)$$

The adjusted parameters in the neuro fuzzy network can be divided into two categories based on the IF-part (premise-part) and THEN-part (consequent-part) parameters of the fuzzy rules [1, 3, 5]. In the premise-part, the center a_{ij} and the width b_{ij} of the Gaussian membership functions are to be fine-tuned whereas, in the consequent-part, the adjusted parameters are the consequent weight c_i.

The adjusted parameters in the neuro fuzzy network can be divided into two categories based on the IF-part (premise-part) and THEN-part (consequent-part) parameters of the fuzzy rules [1, 3, 5]. In the premise-part, the centre a_{ij} and the width b_{ij} of the Gaussian membership functions are to be fine-tuned whereas, in the consequent-part, the adjusted parameters are the consequent weight c_i.

A gradient descent based back-propagation algorithm used LMS error-function to adjust the parameters of the neuro fuzzy network by using the training patterns [1]. The main goal of supervised learning algorithm is to minimize this error-function, which has the formula:

$$E^p(Z) = \tfrac{1}{2}\ \left(Yd^p(t) - Y^p(t)\right)^2 \qquad (5)$$

and

$$E(Z) = \sum_{p=1}^{P} E^p(Z) \qquad (6)$$

where: E^P is an error in the pattern p; E is the total error for the NFS; Yd^P is the desired output in the pattern p; Y^P is the system's output in the pattern p. P is the patterns number; and Z is the parameter vector (a_{11}, …, a_{nm} ;b_{11}, …, b_{nm} ; c_1, …, c_n).

Fig. 2 shows the structure of the identification

$$a_{ij}(t+1) = a_{ij}(t) + k_a \left[Yd^p(t) - Y^p(t) \right] * \left[c_i(t) - Y^p(t) \right] * \frac{u_i}{\sum_{L=1}^{n} u_L} * \frac{x_j - a_{ij}(t)}{b_{ij}(t)^2} \quad (7)$$

Problem with neurofuzzy model. The objective function E(Z) with respect to all parameters in NFS, such as the centres a_{ij} and the widths b_{ij} for the input membership functions, and the centres c_i for the output membership functions [6, 7,10].

$$b_{ij}(t+1) = b_{ij}(t) + k_b \left[Yd^p(t) - Y^p(t) \right] * \left[c_i(t) - Y^p(t) \right] * \frac{u_i}{\sum_{L=1}^{n} u_L} * \frac{\left(x_j - a_{ij}(t)\right)^2}{b_{ij}(t)^3} \quad (8)$$

$$c_i(t+1) = c_i(t) + k_c \left[Yd^p(t) - Y^p(t) \right] * \frac{u_i}{\sum_{L=1}^{n} u_L} \quad (9)$$

where: i =1… n; {n is no. of rules in NFS}; j =1… m ;{ m is no. of input variables}; ka, kb, and kc are the

Learning rates; and t means the learning iteration.

The NF algorithm can be explained it as a following algorithm [6, 7]:

Algorithm 1: NeuroFuzzy algorithm

1. Determine no. of membership functions for each input-output variable.
2. Read patterns table (inputs; and output desired).
3. Initialize as random values:
   - a_{ij} as centres of input membership functions.
   - b_{ij} as widths of input membership functions.
   - c_i as centres of output membership functions.

{i=1… no. of rules; j=1… no. of input variables}

4. For P=1: no Patterns
5. Compute the output of pattern P by (1), (2), (3), and (4).
6. Compute the error of pattern P by (5).
7. Using Back-propagation to update: a_{ij}, b_{ij}, and c_i by (7), (8), and (9); respectively.
8. Compute the total error by (6).
9. IF the total error less or equal acceptable error THEN Stop; otherwise go to step 4.

3. 3-K-MEANS CLUSTERING

The k-means algorithm partitions a collection of N vector into c groups (clusters G_i , i=1,..,c). The purpose of K-Means is to reduce the size and the complexity of the data set and identify the centres of the clusters of data set to produce a concise representation of the behaviour of a system [8]. It suggested by Moody and Darken [9] in 1989.

The algorithm minimizes a dissimilarity (or distance) function which is given in (10).

$$J = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{k, x_k \in G_i} d(x_k - c_i) \quad (10)$$

where: c_i is the centroid of cluster i; and $d(x_k - c_i)$ is the distance between ith centroid (c_i) and kth data point. For simplicity, the Euclidian distance is used as dissimilarity measure and overall dissimilarity function is expressed as in (11).

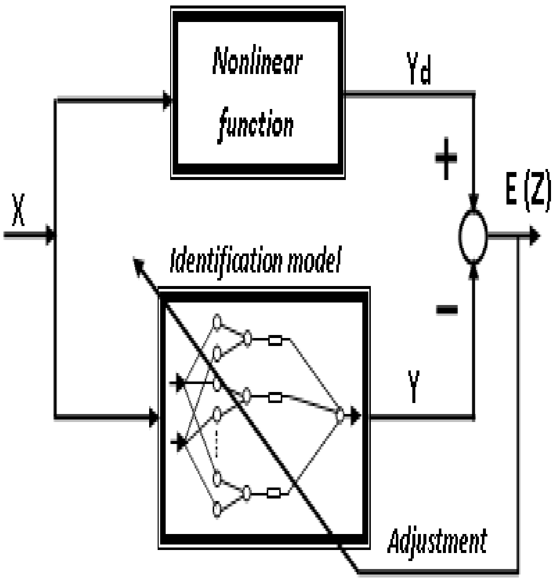


Fig2. *Identification problem with Neuro Fuzzy mdel*

$$J = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \left(\sum_{k, x_k \in G_i} \|x_k - c_i\|^2 \right) \quad (11)$$

Partitioned groups can be defined by a c*n binary membership matrix (U), where the element uij is 1 if the j[th] data point x_j belongs to group i, and 0 otherwise. This explanation is formulated in (12).

$$u_{ij} = \begin{cases} 1 & \text{if } \| x_j - c_i \|^2 \le \| x_j - c_k \|^2, \text{ for each } k \ne i, \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

Since a data point can only be in a group, the membership matrix (U) has two properties which are given in (13) and (14).

$$\sum_{i=1}^{c} u_{ij} = 1, \forall j = 1, \ldots, n \quad (13)$$

$$\sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij} = n \quad (14)$$

Centroids are computed as the mean of all vectors in group i:

$$c_i = \frac{1}{|G_i|} \sum_{k, x_k \in G_i} x_k \quad (15)$$

where: $|G_i|$ is the size of G_i

The k-means algorithm determines the following algorithm [8]:

Algorithm 2: K-means clustering algorithm

1. Initialize the centroids c_i, i=1…c. This is typically achieved by randomly selecting c points from among all of the data points.

2. Determine the membership matrix U by (12).

3. Compute the dissimilarity function by using (11). Stop if its improvement over previous iteration is below a threshold.

4. Compute new centroids using by (15). Go to step 2.

4. Experimental Results

The enhancement of NFS by using K-means clustering algorithm will be shown by running two of experiments on nonlinear functions.

In the two approaches (original NFS, NFS with K-means clustering), six membership functions are used. The numbers of input-output data (training patterns) were chosen to be constant for the two nonlinear functions and equal to 100.

4.1 Experiment I

$$y = \begin{cases} 10 - e^{0.5x} & x \le 0 \\ 10 - e^{-0.5x} & x > 0 \end{cases}$$

where $x \in [-5,5]$ is an input variable, and $y \in [9,10]$ is an output variable. K-means clustering algorithm used training patterns to calculate the optimal initial centre values for input-output membership functions. Table I. shows the initial values for the two approaches.

Results of K-means algorithm as initial centres i membership functions for NFS

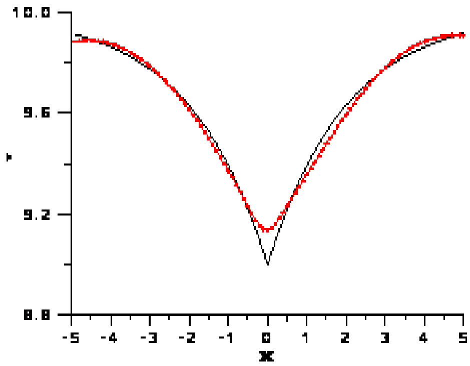
Approach	Input membership functions for X	Output membership functions for Y
	a_i	c_i
NFS with K-means	(-4.18, -2.49, -0.79, 0.89, 2.29, 4.28)	(9.19, 9.28, 9.38, 9.6, 9.76, 9.91)
Original NFS	Random values; according to step3 in algorithm 1	

After the training epochs for the two approaches used the initial valued in Table I, the results of these approaches for y function shown in Table II. It can be seen that NFS with K-means gives a much better performance.

Results of NFS and K-means algorithm for y function

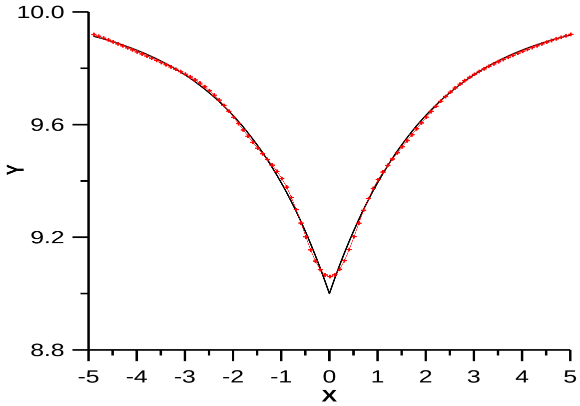
Approach	***Error***	***Real-Time***	***Epochs***
NFS with K-means	0.00005	2.9 sec.	500
Orginal NFS	0.00045	2.2 sec.	500

Fig. 3 shows the approximated outputs of two approaches, (original NFS, NFS with K-means), together with the desired function.



(a) Output of desired y function and NFS

From fig.3, it can be seen that NFS with K-Means clustering has a better result compared with original NFS.



(b) Output of desired y function and NFS with K-means **Figure3**. *Outputs of desire y function (solid-line) and the two approaches (point-line)*

4.2 Experiment II

$Z = (2x + 2y + 0.1)^2 / 37.21$

Where x, y $\in$ [-1,1] is input variables, and Z $\in$ [0,1] is a normalized output variable.

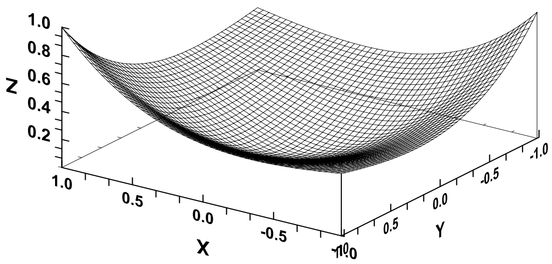
Table III. shows the optimal initial centre values for the two approaches. Results of K-means algorithm as initial centres i membership functions for NFS

Approach	Input membership functions for (X,Y)	Output membership functions for Z
	a_i	C_i
NFS with K-means	(-0.93, -0.61, -0.21, 0.19, 0.53, 0.91)	(0.08, 0.16, 0.38, 0.56, 0.73, 0.9)
Original NFS	Random values; according to step3 in algorithm 1	

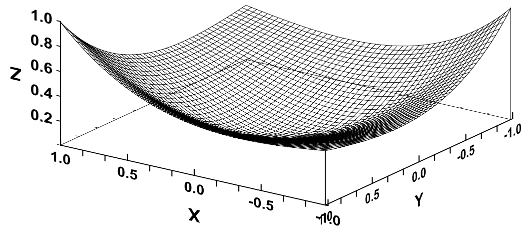
After the training epochs for the two approaches used the initial valued in Table III, the results of the approaches for z function shown in Table IV. It can be seen that NFS with K-means clustering gives a much better performance. Results of NFS and K-means algorithm for Z function

Approach	Error	Real-Time	Epochs
NFS with K-means	0.00127	3.5 sec.	280
Original NFS	0.00474	2.9 sec.	280

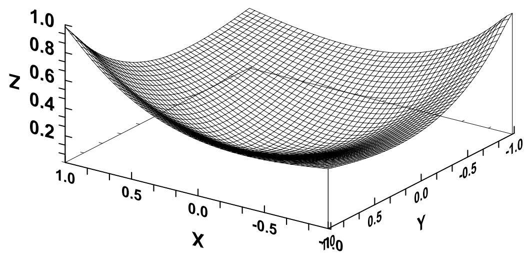
Fig. 4 presents a pictorial presentation of the desired and approximated outputs of two approaches (original NFS, NFS with K-means) , outputs for Z function.



(a) Desired Output



(b) Original NFS



(c) NFS with K-means algorithm

Figure 4. The desired and approximated outputs for form Table IV and Fig. 4, it can be seen that NFS with K-means algorithm has a better result compared with original NFS.

5. Conclusions and Discussion

Clustering is the process of assigning data objects into a set of disjoint groups called clusters so that objects in each cluster are more similar to each other than objects from different clusters. Therefore, this paper aimed to enhance the efficiency of Neuro Fuzzy system (NFS) by using clustering method. In this paper, the optimal initial centre values of input-output membership functions of IF-THEN rules in NFS are determined by using clustering. By these initial values the NFS gives better results compared with original NFS. We examined a K-means clustering algorithm by running two of experiments on nonlinear functions. The obtained results shown that the efficiency of NFS with K-means clustering gives better results.

REFERENCES

[1] Chin-Teng Lin, and Ya-Ching Lu, "A neural fuzzy system with fuzzy supervised learning," IEEE Trans. Syst. Man Cybern. B, Cybern., vol. 26, no. 5, Oct., 1996.

[2] Cheng-Hung Chen , Cheng-Jian Lin, Chin-Teng Lin, "A Recurrent Functional-Link-Based Neural Fuzzy System and Its Applications," in IEEE CIISP, pp.415-420, Apr., 2007.

[3] Ö. Ciftcioglu, M. S. Bittermann, and I. S. Sariyildiz, "A Neural Fuzzy System for Soft Computing," NAFIPS'07, pp. 489-495, Jun., 2007.

[4] Cheng-Jian Lin, Cheng-Hung Chen, and Chin-Teng Lin, "Efficient Self-Evolving Evolutionary Learning for Neurofuzzy Inference Systems," IEEE Trans. Fuzzy Syst., vol.16, no.6, pp.1476-1490, Dec., 2008.

[5] Lim Eng Aik, and Yogan S. Jayakumar, "A Study of Neuro-fuzzy System in Approximation-based Problems," MATEMATIKA, vol. 24, no. 2, pp.113–130, 2008.

[6] Babuška Robert, and Verbruggen Henk, "Neuro-Fuzzy methods for nonlinear system identification," Annual Reviews in Control, vol. 27, no. 1, pp.73-85, Jan., 2003.

[7] Zeshui Xu, and Junjie Wu, "Intuitionistic fuzzy C-means clustering algorithms," J. Syst. Eng. Electron., vol. 21, no. 4, pp.580–590, Aug., 2010.

[8] Tapas Kanungo, David M. Mount, nathan S. Netanyahu, Christine D. Piatko, Ruth Sileverman, and Angela Y. Wu, "An Efficient K-means Clustering Algorithm: Analysis and Implementation," IEEE Trans. Pattern Anal. Machine Intell. vol.24, no.7, pp.881-892, Jul., 2002.

[9] John Moody and Christian J. Darken, "Fast Learning in Networks of Locally-Tuned Processing Units," Neural Computation, vol.1, no.2, pp.281-294, Jun., 1989.

[10] Alshawi, Imad S., Haider Khalaf Allamy, and Rafiqul Zaman Khan. "Development Multiple Neuro-Fuzzy System Using Back-propagation Algorithm." International Journal of Management & Information Technology 6, no. 2 (2013): 794-804.

AUTHORS' BIOGRAPHY

Dr. Rafiqul Zaman Khan, is presently working as a Associate Professor in the Department of Computer Science at Aligarh Muslim University, Aligarh, India. He received his B.Sc Degree from M.J.P Rohilkhand University, Bareilly, M.Sc and M.C.A from A.M.U. and Ph.D (Computer Science) from Jamia Hamdard University. He has 18 years of Teaching Experience of various reputed International and National Universities viz King Fahad University of Petroleum & Minerals (KFUPM), K.S.A, Ittihad University, U.A.E, Pune University, Jamia Hamdard University and AMU, Aligarh. He worked as a Head of the Department of Computer Science at Poona College, University of Pune. He also worked as a Chairman of the Department of Computer Science, AMU, Aligarh. His Research Interest includes Parallel & Distributed Computing, Gesture Recognition, Expert Systems and Artificial Intelligence. Presently 04 students are doing PhD under his supervision. He has published about 40 research papers in International Journals/Conferences. Names of some Journals of repute in which recently his articles have been published are International Journal of Computer Applications (ISSN: 0975-8887), U.S.A, Journal of Computer and Information Science (ISSN: 1913-8989), Canada, International Journal of Human Computer Interaction (ISSN: 2180-1347), Malaysia, and Malaysian Journal of Computer Science(ISSN: 0127-9084), Malaysia. He is the Member of Advisory Board of International Journal of Emerging Technology and Advanced Engineering (IJETAE), Editorial Board of International Journal of Advances in Engineering & Technology (IJAET), International Journal of Computer Science Engineering and Technology (IJCSET), International Journal in Foundations of

Computer Science & technology (IJFCST) and Journal of Information Technology, and Organizations (JITO).

Haider allamy, is a research scholar in the Department of Computer Science; Aligarh Muslim University, Aligarh, India. He joined to his Ph.D course in 28-09-2012. His research interest includes Artificial Intelligence and Expert system. He did B.Sc Degree in computer science from University of Basrah, College of Science, Iraq, M.Sc in computer science from Jamia Hamdard University, Delhi, India. He is envoy by the university of Misan/ Iraq for complete his Ph.D course in Aligarh Muslim University, Aligarh, India.