Science & Technology Development Journal – Engineering and Technology 2024, ():1-8

Open Access Full Text Article

Research Article

The application of multivariable membership functions to the fuzzy neural model

Bui Truong An[1,*], Pham Tuan Anh[1], Pham Thi Nguyen[2]

Use your smartphone to scan this QR code and download this article

ABSTRACT

In recent years, fuzzy neural systems have become increasingly popular due to their powerful learning and interpreting capabilities. In the field of control, the fuzzy neural system is superior to other intelligent systems. In addition, the theory of the combination of neural networks and fuzzy logic is also used in many other fields such as prediction, simulation, decision support, etc. However, most neural systems are Fuzzy systems used today are a combination of neural networks and univariate membership functions in fuzzy theory. These functions have the advantage of being simple and easy to set up, but with that is a lack of interpretability for complex objects. For objects that need to be described by two or more quantities, unidirectional membership functions are not able to represent it. Application of multivariable membership function is necessary in this case. The application of multivariable membership functions encounters many barriers due to their complexity, the algorithms for applying multivariable membership functions are sketchy and have not fully promoted its advantages. In this article, we will introduce a method for applying multivariable Gaussian membership function that allows to improve simulation performance compared to previously introduced methods.

Key words: fuzzy set, multivariable membership functions, Gaussian functions, fuzzy neural model

[1]*Institute of Information Technology, AMST, Ho Chi Minh City, Vietnam.*

[2]*Ho Chi Minh City University of Natural Resources and Environment, Ho Chi Minh City, Vietnam*

Correspondence

Bui Truong An, Institute of Information Technology, AMST, Ho Chi Minh City, Vietnam.

Email: buitruonganmta92@gmail.com

History

- Received: 31-8-2023
- Accepted: 04-4-2024
- Published Online:

DOI :

Check for updates

VNUHCM PRESS

INTRODUCTION

Neutral networks consist of a large number of simple processing elements (neurons) that are interconnected, so when processing information in parallel, there is a huge computing power. However, the knowledge accumulated by the neural network is distributed among all its elements, which makes them practically inaccessible to the observer.

Fuzzy logic control systems do not have this limitation. However, control knowledge is required at the design stage of the control module and must come from experts, and therefore the fuzzy logic control system is not capable of learning.

Combining both approaches allows you to create a system that has both the ability to train a neural network and enhance the intellectual abilities of the system with fuzzy decision rules inherent in the "human" way of thinking.

Such neuro-fuzzy systems are very diverse and are increasingly being improved in accordance with the development of neural network learning algorithms. Among them are gradient descent methods [1]. The disadvantage of these algorithms is that they are slow if the definition of the training step is not satisfactory, and converge easily to local minima. Population algorithms solve these problems and are effective in optimizing a large space, divided into two groups, including evolutionary algorithms and swarm algorithms. The genetic algorithm (GA) belongs to the group of evolutionary algorithms based on such genetic processes as selection, mutation, and exchange [2]. Another algorithm belonging to the group of evolutionary algorithms is the differential evolution algorithm [3] also inspired by biology such as GA, the difference is that a mutant element is created by adding an efficiency number between two elements with previous generations. The swarm algorithm group often draws ideas from animal behavior, such as particle swarm algorithm [4], ant colony algorithm [5], bee swarm algorithm [6], cuckoo search algorithm [7]. There are also support vector algorithms [8] and extreme machine learning algorithms [9].

Most of the above works are built on the basis of one-dimensional membership functions, such as Gaussian, Bell, Triangular. The limitation of this approach is the complexity of the model in terms of the number of rules, which increases exponentially with the number of inputs (spatial curse).

As an effective solution to the above problem, the use of multivariable membership functions in a fuzzy inference system is proposed. In Abonyi *et al.* (2001) [10], a fuzzy model with triangular multivariable membership functions is introduced; these membership func-

Cite this article : An B T, Anh P T, Nguyen P T. **The application of multivariable membership functions to the fuzzy neural model**. *Sci. Tech. Dev. J. – Engineering and Technology* 2024; ():1-8.

tions are obtained by Delaunay triangulation of their characteristic points. The problem with this method is that each fuzzy set is created by linking "nodes" in space. For a multidimensional space, the number of "nodes" required to represent a fuzzy set increases rapidly, resulting in a large number of variables to describe the membership function. Compared to triangular multivariable Delaunay functions, multivariable Gaussian functions require fewer variables to describe a fuzzy set [11–13]. In Kang *et al.* (2007) [11], membership functions are identified using the clustering algorithm. Distance calculation is performed on input and output variables, so data in the same group may not have the same output properties. Data with similar outputs may be located in different clusters due to the greater distance in the input space. This lack of association reduces the explanatory power of the generated fuzzy system.

In Lemos *et al.* (2010) [12], Pratama *et al.* (2013) [13], multivariable Gaussian membership functions are used in developing fuzzy models. The algorithms in the above works are used to develop a fuzzy inference system based on a sequential set of input data. This leads to the fact that in the presence of training data sets, their high accuracy is not guaranteed.

In Basil *et al.* (2019) [14], multivariable Gaussian membership functions are used with incomplete covariance matrices, which is essentially another expression for using one-dimensional Gaussian membership functions. The use of such membership functions provides neither the advantage of the number of fuzzy rules nor the decomposition error reduction.

Most of the algorithms for determining the parameters of fuzzy membership functions in the above works are developed on the basis of algorithms for the synthesis of fuzzy rules for one-dimensional membership functions.

The fuzzy rules generated by these algorithms often overlap and cannot act as independent rules. The overlap of fuzzy rules in fuzzy systems does not allow assessing the reliability of individual fuzzy rules and at the same time creates limitations in extracting knowledge from fuzzy systems. When applying a fuzzy neural system based on a multivariable membership function to decision support systems, the ability to operate independently of fuzzy rules is very important, since it allows you to evaluate the accuracy of a solution given on the basis of individual fuzzy rules. Therefore, the task of constructing a multivariable membership function with fuzzy rules capable of independent operation is relevant.

MULTIVARIATE MEMBERSHIP FUNCTION

To represent multidimensional fuzzy sets, we use multivariable membership functions. Multivariable membership functions are also divided into linear and non-linear. A commonly used linear multivariate membership function is a triangular multivariate membership function.

A linear multivariable membership function is obtained by Delaunay triangulation [15] of their characteristic points.

Like one-dimensional linear membership functions, multivariable linear membership functions have limited flexibility in setting parameters, which complicates the formation of complex dependencies.

Compared to linear multivariate membership functions, non-linear multivariate membership functions, especially multivariate Gaussian functions, are more widely used.

In general, the one-dimensional Gaussian membership function uses an exponential function to project the distance D from a point x in space to the center of the fuzzy set d_1 on the interval [0,1] such that the distance between x and the greater d_1, the smaller the value of the membership function at the point x and vice versa. The multivariate Gaussian membership function also uses the same principle:

$$X(x) = e^{-D^2(x)} \quad (1)$$

where: D is the distance from x to the center C of the fuzzy set, $x = (x_1, x_2, \ldots, x_n)$ are the variables of the multivariable membership function, where n is the number of space dimensions.

The distance D can simply be defined as the Euclidean distance in the space x and C:

$$D = \sqrt{\sum_{i=1}^{n} (x_i - C_i)^2} \quad (2)$$

where x_i is the i-th variable of the multivariable membership function corresponding to the i-th dimension in space;

C_i - i-th coordinate of the fuzzy set center.

The limitation of using Euclidean distance is that the extent is the same in all directions (Figure 1). This reduces the spatial separation of fuzzy sets.

A generalization of Euclidean distance, called normalized Euclidean, allows you to narrow or expand the membership function in a direction parallel to the coordinate axes (Figure 2):

$$D = \sqrt{\sum_{i=1}^{n} \left(\frac{x_i - C_i}{\sigma_i}\right)^2} \quad (3)$$

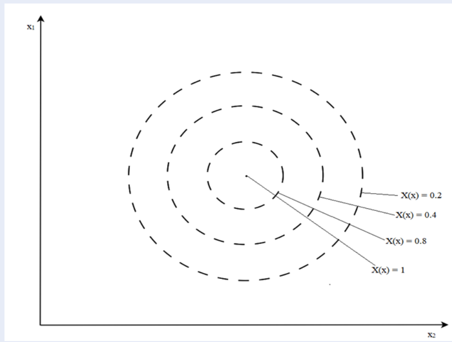

Figure 1: Spatial distribution of multivariate Gaussian membership function based on Euclidean distance

where $\sigma_1, \sigma_2, ..., \sigma_n$ - are the expansion coefficients in dimensions parallel to the coordinate axes.

For expansion or contraction in an arbitrary direction, it is proposed to use the distance based on the idea of Mahalanobis [16], according to which:

$$D(x) = \sqrt{(x-C)^T S^{-1} (x-C)} \quad (4)$$

where: $x = [x_1, x_2, ..., x_n]$- the matrix of variables of the membership function has the size $(1 \times n)$, n - the number of spatial dimensions; $C = [C_1, C_2, ..., C_n]$- matrix of coordinates of the center of the fuzzy set X of size $(1 \times n)$.

S is a matrix of expansion coefficients (variation) of size $(n \times n)$;S^{-1} is the inverse of S.

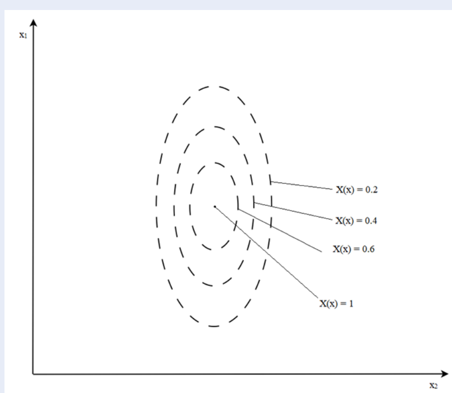

Figure 2: Spatial distribution of Gaussian membership function with two variables based on normalized Euclidean distance

On Figure 3 shows the spatial distribution of the Gaussian membership function based on the Mahalanobis distance with the parameters:

$C = [5\ 5]$, $S = \begin{bmatrix} 10 & 6 \\ 6 & 5 \end{bmatrix}$ with cut-offs

$\alpha = \left\{ 0.8 \quad 0.6 \quad 0.4 \right\}$

The matrix S is chosen and transformed to be a positive definite matrix such that the value under the radical is always positive for $x - C$ values.

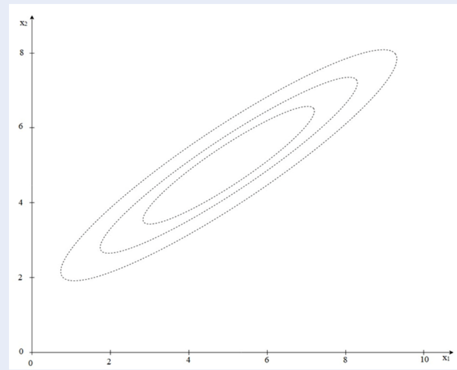

Figure 3: Spatial distribution of a two-variable Gaussian membership function based on the Mahalanobis distance

The Mahalanobis distance is a generalized form of the normalized Euclidean distance. If S is the identity matrix, then the Mahalanobis distance becomes equal to the Euclidean distance. If S is a diagonal matrix and has the value

$$S = \begin{bmatrix} \sigma_1^2 & 0 & \ldots & 0 \\ 0 & \sigma_2^2 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \sigma_n^2 \end{bmatrix},$$

then the Mahalanobis distance becomes the normalized Euclidean distance.

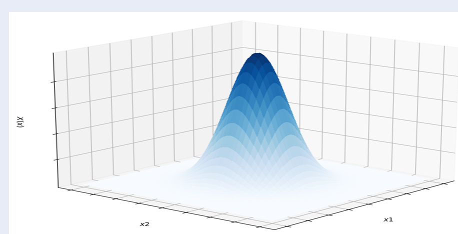

Figure 4: Membership function with two variables

Fuzzy sets with multivariable Gaussian membership functions (Figure 4) have more flexible spatial division than one-dimensional membership functions, and at the same time have a small number of parameters and are easier to implement than a triangular multivariable membership function. Therefore, in recent years it has often been used for the synthesis of fuzzy inference systems in various fields[17].

METHOD

Methods to combine two membership functions

Shrinath. G. A., Plamen. P. A., Lughofer E., Bouchot J. L. and Shaker A. proposed a method for combining two fuzzy sets as follows[13]:

$$C_+ = \left(max\left(U\right) + min\left(U\right)\right)/2 \tag{5}$$

$$S_+ = \left(max\left(U\right) - min\left(U\right)\right)/2 \tag{6}$$

where C_+, S_+ are the coordinates of the center shift point and the matrix of coefficients of the latitudes of the total fuzzy set, $U = \{C_1 \pm \sigma_1,\ C_2 \pm \sigma_2\}$, where C_1, C_2 are the coordinates of the points of the center shift of the component membership functions, σ_1, σ_2 are the width parameters of the fuzzy sets on the α-section. The coverage of the total fuzzy set by this method covers the coverage of component fuzzy sets in all dimensions in space.

Another method for combining two fuzzy membership functions was proposed by Mahardika P. as follows[18]:

$$C_+ = \frac{N_1 C_1 + N_2 C_2}{N_1 + N_2} \tag{7}$$

$$S = \left(\frac{N_1 S_1^{-1} + N_2 S_2^{-1}}{N_1 + N_2}\right)^{-1} \tag{8}$$

where C_1, C_2, S_1, S_2 are the coordinates of the points of displacement of the centers and the matrices of the coefficients of the latitudes of the component membership functions. C_+, S_+ - coordinates of the center offset point and matrix of latitude coefficients. N_1, N_2 are the corresponding proportional weights of the component fuzzy sets.

Proposed Method to combine two membership functions

The goal of combining two fuzzy sets is to replace two multivariable Gaussian fuzzy sets with a new multivariable Gaussian fuzzy set. The composite fuzzy set must cover two composite fuzzy sets on the α-section. Since the multivariate Gaussian membership function is built on the basis of the multivariate Gaussian distribution, we construct a method for combining two multivariate membership functions based on the synthesis of two multivariate Gaussian normal distributions.

The center and covariance matrix representing the distribution of a set of N data points X_1, X_2, ..., X_N in n-dimensional space is determined by the following formula:

$$C_N = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{9}$$

$$\Sigma_N = \frac{1}{N}\sum_{i=1}^{N} \left(X_i - C_N\right)^T \left(X_i - C_N\right) \tag{10}$$

where X_i is of size $(1 \times n)$, is the i-th data point in dataset N. C_N is of size $(1 \times n)$, which is the center of the multivariate Gaussian distribution. $\sum_N$ of size $(n \times n)$ is the covariance matrix of the multivariate Gaussian distribution.

Suppose we have two multivariate distributions $G_1 = \{C_{N_1}, \Sigma_{N_1}\}$ and $G_2 = \{C_{N_2}, \Sigma_{N_2}\}$ representing the distributions of two sets of N_1 and N_2 distinct data, respectively.

$$C_{N_1} = \frac{1}{N_1}\sum_{i=1}^{N_1} X_i \tag{11}$$

$$\Sigma N_1 = \frac{1}{N_1}\sum_{i=1}^{N_1} \left(X_i - C_{N_1}\right)^T \left(X_i - C_{N_1}\right) \tag{12}$$

$$C_{N_2} = \frac{1}{N_2}\sum_{i=1}^{N_2} Y_i \tag{13}$$

$$\Sigma N_2 = \frac{1}{N_2}\sum_{i=1}^{N_2} \left(Y_i - C_{N_2}\right)^T \left(Y_i - C_{N_2}\right) \tag{14}$$

The union of G_1 and G_2 is understood as finding the multivariate distribution of G_+, which represents the N_1 + N_2 distribution of their data points. The center and covariance matrix G_+ are defined as follows:

$$C_+ = \frac{\sum_{i=1}^{N_1} X_i + \sum_{i=1}^{N_2} Y_i}{N_1 + N_2} = \frac{N_1 C_{N_1} + N_2 C_{N_2}}{N_1 + N_2} \tag{15}$$

$$\Sigma_+ = \frac{\sum_{i=1}^{N_2} \left(X_i + C_+\right)^T \left(X_i - C_+\right)}{N_1 + N_2} + \frac{\sum_{i=1}^{N_2} \left(Y_i + C_+\right)^T \left(Y_i - C_+\right)}{N_1 + N_2} \tag{16}$$

We replace: $X_i - C_+ = \left(X_i - C_{N_1}\right) + \left(C_{N_1} - C_+\right)$, we get:

$$\begin{aligned}
&\textstyle\sum_{i=1}^{N_1} \left(X_i - C_+\right)^T \left(X_i - C_+\right) \\
&= \textstyle\sum_{i=1}^{N_1} \left[\left(X_i - C_{N_1}\right) + \left(C_{N_1} - C_+\right)\right]^T \times \\
&\left[\left(X_i - C_{N_1}\right) + \left(C_{N_1} - C_+\right)\right] \\
&= \textstyle\sum_{i=1}^{N_1} \left[\left(X_i - C_{N_1}\right) + \left(X_i - C_{N_1}\right)\right]^T + \\
&\textstyle\sum_{i=1}^{N_1} \left(X_i - C_{N_1}\right)^T \left(C_{N_1} - C_+\right) + \\
&\textstyle\sum_{i=1}^{N_1} \left(C_{N_1} - C_+\right)^T \left(X_i - C_{N_1}\right) + \\
&\textstyle\sum_{i=1}^{N_1} \left(C_{N_1} - C_+\right)^T \left(C_{N_1} - C_+\right) \\
&= \textstyle\sum_{i=1}^{N_1} \left(X_i - C_{N_1}\right)^T \left(X_{N_1} - C_{N_1}\right) + \\
&\left(\textstyle\sum_{i=1}^{N_1} \left(X_i - C_{N_1}\right)^T\right) \left(C_{N_1} - C_+\right) + \\
&\left(C_{N_1} - C_+\right)^T \textstyle\sum_{i=1}^{N_1} \left(X_i - C_{N_1}\right) + \\
&N_1 * \left(C_{N_1} - C_+\right)^T \left(C_{N_1} - C_+\right)
\end{aligned} \tag{17}$$

Since C_{N_1} is the center of G_1, then

$$\sum_i^{N_1} (X_i - C_{N_1})^T = 0 \quad (18)$$

And

$$\sum_i^{N_1} (X_i - C_{N_1}) = 0 \quad (19)$$

Substituting (12), (18) and (19) into (17), we get:

$$\begin{aligned}&\sum_{i=1}^{N_1} (X_i - C_+)^T (X_i - C_+) \\ &= N_1 * \Sigma_{N_1} * (C_{N_1} - C_+)^T (C_{N_1} - C_+)\end{aligned} \quad (20)$$

Similar analysis

$$\begin{aligned}&\sum_{i=1}^{N_2} (Y_i - C_+)^T (Y_i - C_+) \\ &= N_2 * \Sigma_{N_2} * (C_{N_2} - C_+)^T (C_{N_2} - C_+)\end{aligned} \quad (21)$$

Substituting (20) and (21) into (16), we get:

$$\begin{aligned}\Sigma_+ = &\frac{N_1 * \Sigma_{N_1} + N_2 * \Sigma_{N_2}}{N_1 + N_2} + \\ &\frac{N_1 * (C_{N_1} - C_+)^T (C_{N_1} - C_+)}{N_1 + N_2} + \\ &\frac{N_2 * (C_{N_2} - C_+)^T (C_{N_2} - C_+)}{N_1 + N_2}\end{aligned} \quad (22)$$

It is easy to see that the center $G+$ is a point on the line connecting the centers G_1 and G_2, and divides this segment into two segments corresponding to:

$$C_{N_1} - C_+ = \frac{N_2}{N_1 + N_2} (C_{N_1} - C_{N_2}) \quad (23)$$

$$C_{N_2} - C_+ = \frac{N_1}{N_1 + N_2} (C_{N_1} - C_{N_2}) \quad (24)$$

Substituting (23) and (24) into (22), we get:

$$\begin{aligned}\Sigma_+ = &\frac{N_1 * \Sigma_{N_1} + N_2 * \Sigma_{N_2}}{N_1 + N_2} + \\ &\frac{N_1 N_2}{(N_1 + N_2)^2} (C_{N_1} - C_{N_2})^T (C_{N_1} - C_{N_2})\end{aligned} \quad (25)$$

Formula (25) gives us the covariance matrix of the multivariate distribution G+ through the center and the covariance matrix of the original distributions.

Unlike combining two distributions with multiple variables, when combining two fuzzy sets, the coverage aspect of the combined fuzzy set must also be taken into account.

Apply a formula similar to (25) to combine two fuzzy sets with centers C_1, C_2 and the corresponding matrix of expansion coefficients S_1, S_2. Values N_1, N_2 are replaced by $\det(S_1)$, $\det(S_2)$:

$$\begin{aligned}S_+ = &\frac{det(S_1) * S_1 + det(S_2) * S_2}{det(S_1) + det(S_2)} \\ &+ \frac{det(S_1) * det(S_2)}{(det(S_1) + det(S_2))^2} (C_1 - C_2)^T (C_1 - C_2)\end{aligned} \quad (26)$$

The center of the fuzzy set of the sum is transformed from formula (15) as follows[13]:

$$C_+ = \frac{det(S_1) * C_1 + det(S_2) * C_2}{det(S_1) + det(S_2)} \quad (27)$$

As suggested above, we use a sufficiently small α-slice to define a cover of a multidimensional Gaussian fuzzy set. The criterion for the combined fuzzy set is that the cut area α must be a covering of two composite fuzzy sets. The results of combining two fuzzy membership functions according to the formula (26) are shown in Figure 5.

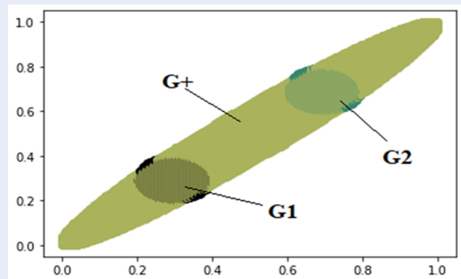

Figure 5: Multivariate Gaussian aggregation by formulas (26)

It is easy to see that the coverage of the combined fuzzy set does not correspond to the coverage of the two original fuzzy sets. To analyze the influence of the components of formula (22) on the coverage of the two combined fuzzy set, we proceed to the union of two fuzzy sets with its first term (equivalent to Mahardik P.'s algorithm):

$$S_+ = \frac{det(S_1) * S_1 + det(S_2) * S_2}{det(S_1) + det(S_2)}$$

The results of combining two fuzzy sets using the first term of formula (26) are shown in Figure 6.

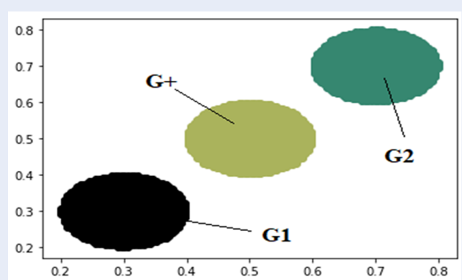

Figure 6: Multivariate Gaussian aggregation by the first term of formula (26)

Accordingly, we see that the first term allows us to average the coverage of component distributions, and the second element allows us to expand the coverage along the line connecting their centers.

To generate the most appropriate aggregate function, we will use the following union formula:

$$D = \frac{\overrightarrow{C_1C_2}}{2}$$

$$C_+ = C_1 + D + \left(\overrightarrow{r_\alpha^1} - \overrightarrow{r_\alpha^2}\right)/2 \quad (28)$$

$$S_+ = \frac{det\,(S_1) * S_1 + det\,(S_2) * S_2}{det\,(S_1) + det\,(S_2)} - \frac{D^T D}{2 * \ln(\alpha)} \quad (29)$$

where C_+, C_1, C_2 are the centers of the new fuzzy set and two component fuzzy sets, respectively, $\overrightarrow{r}_\alpha^1$ is the radius vector of the first partial fuzzy set on the section α in the direction $\overrightarrow{C_1C_2}$, $\overrightarrow{r}_\alpha^2$ is the radius vector of the second component fuzzy set on the section α in the direction $\overrightarrow{C_2C_1}$. All of the above quantities have size $(1\times n)$, where n is the number of dimensions of the input space. S_+, S_1, S_2 of size $(n \times n)$ is the covariance matrix of the new multidimensional fuzzy set and two-component fuzzy set, respectively, D is the radius vector in the direction $\overrightarrow{C_1C_2}$ on the section α of the total fuzzy set. The $D^T D$ multiplication is the $(n\times 1)*(1\times n)$ matrix multiplication.

The idea of the union method is to determine the extreme point of two fuzzy rules in the direction $\overrightarrow{C_2C_1}$, the center of the entire fuzzy set is the midpoint of the above two points, and the radius is a vector from the center to one of the two points.

Using the projection of the multivariable Gaussian function onto a plane perpendicular to the input plane and passing through the centers of the two combined fuzzy sets, we obtain the following one-dimensional Gaussian function:

$$H_1(x) = e^{-0.5\frac{(x-C)^2}{\sigma^2}}$$

The results of combining two fuzzy membership functions according to the formula (29) are shown in Figure 7.

We need to determine the value of σ so that the new membership function has α at the point x_0 (stretch the membership function to position x_0):

$$\begin{aligned} H(x_0) = \alpha &\Leftrightarrow e^{-0.5\frac{x_0^2}{\sigma^2}} = \alpha \\ \Leftrightarrow \sigma^2 &= \frac{x_0^2}{-2\ln(\alpha)} with\ (0 < \alpha < 1) \end{aligned} \quad (30)$$

Based on this, we expand and experiment to construct formula (29).

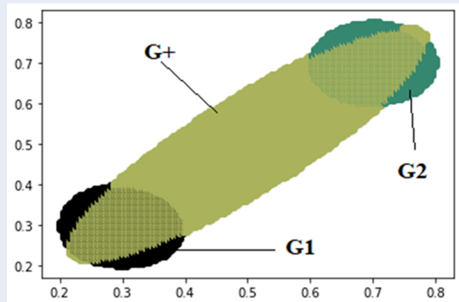

Figure 7: The union of two multivariable Gaussian functions by formula (29)

RESULTS

To evaluate the performance of the proposed method for combining fuzzy membership functions, a comparison is made with other methods. The initial conditions for modeling algorithms are the same:

n_term = 10; loss_thresold = 0.01; loss_max = 1

The fuzzy rule has the form

$$R^i: If\ x \in X^i(C_i, S_i)\ then\ y_i = c_i^0$$

The training data is a set of 25 values taken at regular intervals in the interval $[0;4] \times [0;4]$. The algorithm stops when 10 fuzzy rules are reached.

Table 1 displays the comparison results of combining two membership functions using the proposed method and existing methods.

$$O_{Rule} = \sum_{i=1}^{m} |\widehat{y}(C_i) - y(C_i)|$$

where ORule is an error according to fuzzy rules, C_i is the coordinates of the center of fuzzy rule i, is the model output at C_i, $y(C_i)$, is the true value of the linear function at C_i, input, m - is the number of fuzzy rules.

$$O_{Model} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2$$

where O_{Model} is the model error during testing, $\widehat{Y}_i$ is the model output at the i-th observation, Y_i is the true value of the linear function of the i-th observation, n is the number of observations.

DISCUSSION

Shrinath G.A. method has a much larger error than the other two methods. It is explained that this method was created for building fuzzy neural models without taking into account the overlap between fuzzy rules and is applied to models in which the input data are independent. This union method creates

Table 1: Results of the application of methods to combine two fuzzy membership functions

Mode type	Criteria	Method Shrinath G. A.	Method Mahardik	Suggested method
Linear model	OModel	4.343	1.466	0.651
	ORule	642.330	0.820	0.328
Square model	OModel	33.68	0.715	0.255
	ORule	146.34	0.974	0.958

a union fuzzy set that spans the component fuzzy sets but does not take into account the relative positions of the component fuzzy sets. This results in a combined fuzzy set containing too many "redundancies", which increases the intersection between fuzzy rules.

Mahardik P.'s pooling method takes the average of fuzzy sets of components. The advantages of this method are the simplicity of calculation and the restriction of superposition between fuzzy sets, but this method has the following disadvantages:

1. Many points in the input space are in two-component fuzzy sets, but not in the combined fuzzy set, which leads to an incomplete description of the information.

2. In the case when the extended matrices of membership functions belonging to a partial fuzzy set are diagonal matrices, the extended matrix of the combined fuzzy set is also a diagonal matrix. Therefore, the combined fuzzy set cannot represent the relative positions of the fuzzy sets of components.

The proposed union method is an improved version of Mahardika's method. This method extends the result of the Mahardika method along a line connecting two centers of a partial fuzzy set until they coincide with their extreme points. Thus, the proposed method can show the relative position between two component fuzzy sets without increasing the overlap between fuzzy rules.

The simulation results show that the proposed method of combining fuzzy sets improves the accuracy of the neuro-fuzzy model and the independent operation of the generated fuzzy rules compared to the Mahardika method by increasing the computational complexity.

CONCLUSION

In the content of the article, we have analyzed the advantages and disadvantages of the multi-dimensional association functions used up to now, in addition, we have analyzed the advantages and disadvantages of the methods of combining the membership multivariable functions. On the theoretical basis, we have developed a new method of aggregating multivariable membership functions. The use of this method of combining multivariable membership functions in the neuro-fuzzy model synthesis algorithm allows reducing the average error of fuzzy rules during independent work from 0.82 to 0.328 and reducing the total error of fuzzy rules of the model from 1.466 to 0.651 when solving modeling problems compared to the Mahardika method. The content of the article creates a basis for applying multidimensional membership functions to the problem of building a fuzzy neural system in practice. In the future, we will aim to develop a theoretical system to be able to apply multidimensional membership functions to describe complex objects, to create more intelligent systems.

CONFLICT OF INTEREST

The authors assure that there is no conflict of interest in publishing the article.

AUTHORS' CONTRIBUTION

Bui Truong An participated in coming up with ideas for writing articles, collecting data and writing the manuscript.

Pham Thi Nguyen contributed to data interpretation and proofreading the article.

Pham Tuan Anh contributed to the Vietnamese - English translation and edited the article format.

REFERENCES

1. Dutta A, Nayak A, Aditya, Panda RR, Nagwani NK. A Neuro Fuzzy System Based Inflation Prediction of Agricultural Commodities. In: 11th ICCCNT. Electronic ISBN: 978-1-7281-6851-7. October 2020;Available from: https://doi.org/10.1109/ICCCNT49239.2020.9225453.
2. Ivanova D, Dejanov M. Fuzzy Logic Control Design Based on the Genetic Algorithm for a Modular Servo System. In: 17th Conference on Electrical Machines, Drives and Power Systems (ELMA). Electronic ISBN: 978-1-6654-3582-6. August 2021;Available from: https://doi.org/10.1109/ELMA52514.2021.9503052.
3. Kien CV, Son NN, Anh HPH. Identification of 2-DOF pneumatic artificial muscle system with multilayer fuzzy logic and differential evolution algorithm. In: 12th IEEE Conference on Industrial Electronics and Applications (ICIEA). Electronic ISBN: 978-1-5090-6161-7. February 2018;.

4. Sharma S, Kalra U, Srivathsan S, Rana KPS, Kumar V. Efficient air pollutants prediction using ANFIS trained by Modified PSO algorithm. In: 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO). Electronic ISBN: 978-1-4673-7231-2. December 2015;Available from: https://doi.org/10.1109/ICRITO.2015.7359316.
5. Juang CF, Lin CH, Bui TB. Multiobjective Rule-Based Cooperative Continuous Ant Colony Optimized Fuzzy Systems With a Robot Control Application. IEEE Transactions on Cybernetics. 2018 Oct;50:650-663;PMID: 30296249. Available from: https://doi.org/10.1109/TCYB.2018.2870981.
6. Chou KP, Lin CT, Lin WC. A self-adaptive artificial bee colony algorithm with local search for TSK-type neuro-fuzzy system training. In: 2019 IEEE Congress on Evolutionary Computation (CEC). Electronic ISBN: 978-1-7281-2153-6. August 2019;Available from: https://doi.org/10.1109/CEC.2019.8790334.
7. Jin Q, Wang C, Wang H, Cai W, Niu Y. Hybrid Fuzzy Cuckoo Search Algorithm for MIMO Hammerstein Model Identification Under Heavy-Tailed Noises. In: 37th Chinese Control Conference (CCC). Electronic ISBN: 978-988-15639-5-8. October 2018;PMID: 30043400. Available from: https://doi.org/10.23919/ChiCC.2018.8482855.
8. Nurcahyono AH, Nhita F, Saepudin D, Aditsania A. Price Prediction of Chili in Bandung Regency Using Support Vector Machine (SVM) Optimized with an Adaptive Neuro-Fuzzy Inference System (ANFIS). In: 7th International Conference on Information and Communication Technology (ICoICT). Electronic ISBN: 978-1-5386-8052-0. September 2019;Available from: https://doi.org/10.1109/ICoICT.2019.8835367.
9. Abdusamad A, Aburakhis M. Adaptive Control of Nonlinear Systems Represented by Extreme Learning Machine (ELM) and the Fuzzy Logic Control (FLC). In: IEEE 1st International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering MI-STA. Electronic ISBN: 978-1-6654-1856-0. June 2021;Available from: https://doi.org/10.1109/MI-STA52233.2021.9464386.
10. Abonyi J, Babuska R, Szeifert F. Fuzzy modeling with multivariate membership functions: gray-box identification and control design. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 2001 Oct;31:755-767;PMID: 18244840. Available from: https://doi.org/10.1109/3477.956037.
11. Kang D, Yoo W, Won S. Multivariable TS fuzzy model identification based on mixture of Gaussians. In: 2007 International Conference on Control, Automation and Systems. December 2007;.
12. Lemos A, Caminhas W, Gomide F. Multivariable Gaussian Evolving Fuzzy Modeling System. IEEE Transactions on Fuzzy Systems. 2010 Oct;19:91-104;Available from: https://doi.org/10.1109/TFUZZ.2010.2087381.
13. Pratama M, Anavatti SG, Angelov PP, Lughofer E. PANFIS: A Novel Incremental Learning Machine. IEEE Transactions on Neural Networks and Learning Systems. 2013 Jul;25:55-68;PMID: 24806644. Available from: https://doi.org/10.1109/TNNLS.2013.2271933.
14. Basil JMA, Al-Hadithi M, Jiménez A. Multidimensional membership functions in T-S fuzzy models for modelling and identification of nonlinear multivariable systems using genetic algorithms. Applied Soft Computing. 2019 Feb;75:607-615;Available from: https://doi.org/10.1016/j.asoc.2018.11.034.
15. Abonyi J, Babuska R, Szeifert F. Fuzzy modeling with multivariate membership functions: gray-box identification and control design. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 2001 Oct;31:755-767;PMID: 18244840. Available from: https://doi.org/10.1109/3477.956037.
16. Mahalanobis PC. On the generalised distance in statistics. Proceedings of the National Institute of Sciences of India. 1936;2(1):49-55;.
17. Ferdaus MM, Pratama M, Anavatti SG, Garratt MA, Pan Y. Generic Evolving Self-Organizing Neuro-Fuzzy Control of Bio-Inspired Unmanned Aerial Vehicles. IEEE Transactions on Fuzzy Systems. 2020 Aug;28(8):1542-1556;Available from: https://doi.org/10.1109/TFUZZ.2019.2917808.
18. Pratama M, Anavatti SG, Lughofer E. GENEFIS: Toward an Effective Localist Network. IEEE Transactions on Fuzzy Systems. 2014 Jun;22(3):547-562;Available from: https://doi.org/10.1109/TFUZZ.2013.2264938.

Tạp chí Phát triển Khoa học và Công nghệ – Engineering and Technology 2024, ():1-1

Open Access Full Text Article

Bài nghiên cứu

Ứng dụng của hàm liên thuộc đa biến vào mô hình nơ-ron mờ

Bùi Trường An[1,*], Phạm Tuấn Anh[1], Phạm Thị Nguyên[2]

TÓM TẮT

Trong những năm gần đây, các hệ thần kinh mờ ngày càng trở nên phổ biến nhờ khả năng học tập và diễn giải mạnh mẽ của chúng. Trong lĩnh vực điều khiển, hệ thần kinh mờ tỏ ra vượt trội hơn các hệ thống thông minh khác. Ngoài ra, lý thuyết về sự kết hợp giữa mạng nơ-ron và logic mờ còn được sử dụng trong nhiều lĩnh vực khác như dự đoán, mô phỏng, hỗ trợ quyết định, v.v. Tuy nhiên, hầu hết các hệ thống nơ-ron mờ được sử dụng ngày nay đều là sự kết hợp giữa mạng nơ-ron và các hàm liên thuộc mờ đơn biến. Các hàm này có ưu điểm là đơn giản, dễ thiết lập nhưng đi kèm với đó là thiếu khả năng diễn giải đối với các đối tượng phức tạp. Đối với các đối tượng cần được mô tả bằng hai đại lượng trở lên, các hàm liên thuộc đơn chiều không thể biểu diễn nó. Việc áp dụng hàm thành viên đa biến là cần thiết trong trường hợp này. Việc ứng dụng hàm thành viên nhiều biến gặp nhiều rào cản do tính phức tạp, các thuật toán áp dụng hàm thành viên nhiều biến còn sơ sài và chưa phát huy hết ưu điểm của chúng. Trong bài viết này, chúng tôi sẽ giới thiệu một phương pháp áp dụng hàm thành viên Gaussian đa biến cho phép cải thiện hiệu suất mô phỏng so với các phương pháp đã giới thiệu trước đó.

Từ khoá: Tập mờ, hàm liên thuộc đa biến, hàm Gaussian, mô hình thần kinh mờ

[1]*Viện Công nghệ thông tin, Thành phố Hồ Chí Minh, Việt Nam.*

[2]*Trường Đại học Tài nguyên Môi trường thành phố Hồ Chí Minh, Thành phố Hồ Chí Minh, Việt Nam.*

Liên hệ

Bùi Trường An, Viện Công nghệ thông tin, Thành phố Hồ Chí Minh, Việt Nam.

Email: buitruonganmta92@gmail.com

Lịch sử

- Ngày nhận: 31-8-2023
- Ngày chấp nhận: 04-4-2024
- Ngày đăng:

DOI :

Trích dẫn bài báo này: An B T, Anh P T, Nguyên P T. **Ứng dụng của hàm liên thuộc đa biến vào mô hình nơ-ron mờ** . *Sci. Tech. Dev. J. - Eng. Tech.* 2024; ():1-1.