Information Sciences 179 (2009) 2123–2145

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# Type-1 and type-2 fuzzy inference systems as integration methods in modular neural networks for multimodal biometry and its optimization with genetic algorithms

Denisse Hidalgo, Oscar Castillo *, Patricia Melin

*Tijuana Institute of Technology, Division of Graduate Studies and Research, Department of Computer Science, 22500 Tijuana BC, Mexico*

A R T I C L E   I N F O

*Article history:*
Received 12 December 2007
Received in revised form 1 July 2008
Accepted 9 July 2008

*Keywords:*
Type-2 fuzzy logic
Pattern recognition
Neural networks
Hybrid intelligent systems

A B S T R A C T

We describe in this paper a comparative study between fuzzy inference systems as methods of integration in modular neural networks for multimodal biometry. These methods of integration are based on techniques of type-1 fuzzy logic and type-2 fuzzy logic. Also, the fuzzy systems are optimized with simple genetic algorithms with the goal of having optimized versions of both types of fuzzy systems. First, we considered the use of type-1 fuzzy logic and later the approach with type-2 fuzzy logic. The fuzzy systems were developed using genetic algorithms to handle fuzzy inference systems with different membership functions, like the triangular, trapezoidal and Gaussian; since these algorithms can generate fuzzy systems automatically. Then the response integration of the modular neural network was tested with the optimized fuzzy systems of integration. The comparative study of the type-1 and type-2 fuzzy inference systems was made to observe the behavior of the two different integration methods for modular neural networks for multimodal biometry.
© 2008 Elsevier Inc. All rights reserved.

## 1. Introduction

The primary goal of this research was to apply type-1 fuzzy logic and type-2 fuzzy logic as methods of response integration in modular neural networks. Response integration in this case consists in combining the obtained partial results of the modules by which the modular neural network is formed. Also, to optimize the type-1 and type-2 fuzzy systems with genetic algorithms with the purpose of obtaining the optimal results in the recognition, and making a fair comparison between the different fuzzy logic methods of response integration in modular neural networks. The application of combining several biometrics measures in pattern recognition is used because is a sufficiently complex problem to evaluate the difference between type-1 and type-2 fuzzy logic in integrating uncertain information of several sources [20,21].

Biometry, is a discipline that studies the recognition of the people through its physiological characteristics (fingerprint, face, retina...) or of behavior (voice, signature,...) [6]. The interest of the society to use biometric patterns to identify or verify the authenticity of the people has had a drastic increase, that it is reflected in the appearance of diverse practical applications like identity passports that include biometrics characteristics. Biometry provides a true identification of people, since this technology is based on the recognition of unique corporal characteristics, reason why recognizes the people based on who they are. Only biometric identification can provide a really efficient and precise control of the people, since it is possible to know with a high degree of certainty that the person that went through this form of recognition is the recognized person.

* Corresponding author.
*E-mail address:* ocastillo@hafsamx.org (O. Castillo).

0020-0255/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2008.07.013

As it is habitual in many scientific disciplines, before making a search of the solutions to the problem, it is reasonable and preferable to stop a moment and to make an analysis of the problem. The result of this analysis will be a vision of the different parts that form the whole, having transformed the initial task, probably complex, in a set of more elementary subtasks, susceptible to be approached in a simpler and efficient way. Once this is done, the problem is transformed into the opposite: to integrate the obtained partial results of each of the subtasks and of generating the solution to the complete problem.

## 2. Modular neural networks

As it is common in many scientific disciplines, before making the search of the solutions to the problem, it is reasonable and preferable to stop a moment and make an analysis of the problem. The result of this analysis will be a vision of the different parts that form the whole, having transformed the initial task, probably complex, in a set of more elementary subtask, able to be approached in a simpler and efficient way. Once this is done, the problem is transformed into the opposite: to integrate the obtained partial results of each one of the subtasks and of generating the solution to the complete problem. The first step is to divide the task (problem) in to subtasks, and later to create and to organize in a suitably way the constructed subsystems to allow the communication among them and thus to integrate them as a whole, which provides the desired solution. The idea of modularity, as it was proposed in the origins of the connectionist computation, has been inspired in the biological models. A review of the physiological structures of the nervous system in vertebrate animals reveals the existence of a representation and hierarchical modular processing of the information [7].

Considering as a basis the biological indications, one of the first modular approaches to complex systems was proposed by Jacobs and Jordan [22] that can use two types of different methods of learning:

- *Supervised learning*: during which an external teacher provides for each input the correct output. However, this teacher does not specify which module is the one that must learn the corresponding pair (input, desired output).
- *Unsupervised learning*: which basically consists of a competitive learning, in which the different modules are competing for learning for a given problem [7].

In general, a computational system can be considered a modular structure if it is possible to divide it in two or more modules, in which each individual module can evaluate different or the same inputs without communicating with the others. The outputs of the modules are aggregated by an integrating unit, which decides:

- How the modules are combined to form the final output of the system.
- How each module must learn the patterns.

It is important to mention that the use of the RNM to solve a problem in particular, requires ample knowledge of the problem to be able to make the subdivision of the problem, and to build the suitable modular architecture to solve it, in such a way that it is possible to train each of the modules independently, and later to integrate the knowledge learned by each module, in the global architecture [1].

### 2.1. Methods of integration

According to the form in which the division of the tasks takes place, the integration method allows to integrate or to combine the results given by each of the constructed modules. Some of the commonly used methods of integration are: average, gating network, fuzzy inference systems, mechanism of voting using softmax function, the winner takes everything to it, among others [1].

In this paper we will describe methods of integration based fuzzy inference systems, since it is one that is of interest to us in this research work.

We show next the block diagram of the modular neural network (see Fig. 1).

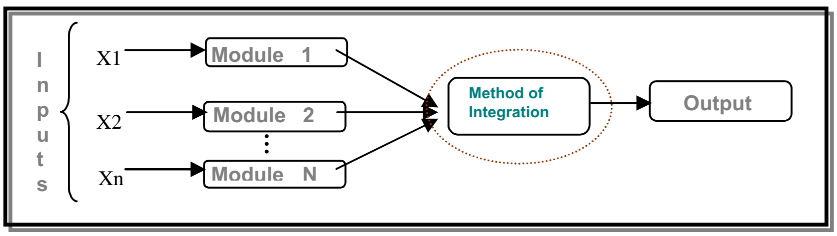


**Fig. 1.** Blocks diagram of a modular neural network.

## 3. Type-1 fuzzy logic

Fuzzy logic was born in 1965 with the publication of "fuzzy sets" [10] written by Lofti A. Zadeh in the University of California in Berkeley for the Information an Control magazine, which was based on the work of Lukasiewicz [11] on multiple-valued logic. Once the foundations of fuzzy logic became firm, their applications have grown in number and diversity, and its influence within basic sciences, has become more visible and more substantial [12]. Fuzzy logic creates mathematical approaches in the solution of certain kinds of problems. Fuzzy logic produces exact results from vague data, thus it is particularly useful in electronic or computational applications [9].

– *Type-1 fuzzy inference system*:

The basic structure of fuzzy inference system consists of three conceptual components: a set of rules, which contains a selection of fuzzy rules; a data base (or dictionary), that defines the used membership functions in the rules; and a reasoning mechanism, that makes the inference procedure (usually fuzzy reasoning). The basic fuzzy inference system can take fuzzy or traditional inputs, but the outputs that are produced are always fuzzy sets. Some times it is necessary to have a traditional output, especially when a fuzzy inference system is used as a controller. Then, a "defuzzification" method is needed to extract the numerical value of output (see Fig. 2).

A fuzzy inference system is a non linear mapping of its input space to its output space. This mapping is obtained by means of a set of fuzzy if-then rules, each of which describes the local behavior of the mapping (see Fig. 3).

In this paper we study the integration method of modular neural networks. We use type-1 and type-2 fuzzy inference systems as integration methods and genetic algorithms are used to optimize the structure of the fuzzy system.

## 4. Type-2 fuzzy logic

The original fuzzy logic (FL), was proposed by Lotfi Zadeh, more than 40 years ago, and this cannot fully handle all the uncertainty present in real-world problems [13]. "To handle," it is understood as "to model and to reduce to the minimum the effect of". That type-1 fuzzy logic cannot completely do this, sounds paradoxical because this has the uncertainty connotation. Type-2 fuzzy logic can handle uncertainty because it can model it and reduce to the minimum their effects. Also, if all the uncertainties disappear, type-2 fuzzy logic reduces to type-1 fuzzy logic, in the same way that, if the randomness disappears, the probability is reduced to the determinism [14]. Fuzzy sets and fuzzy logic are the foundation of fuzzy systems,

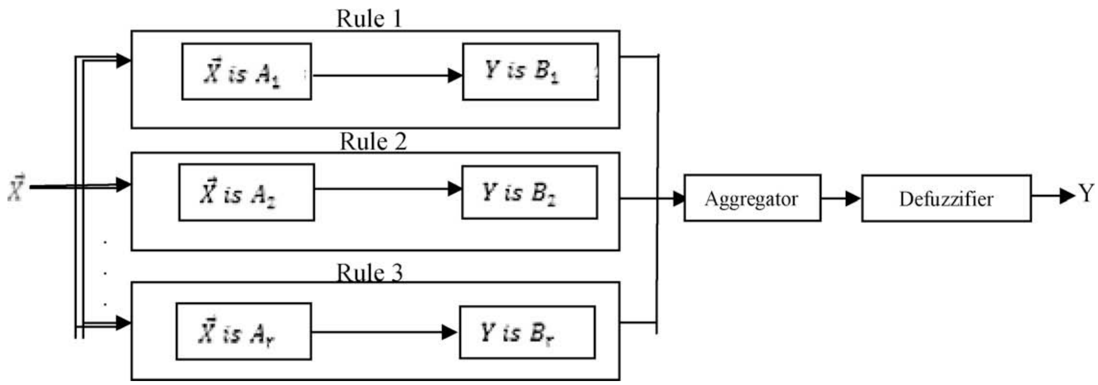


**Fig. 2.** Fuzzy inference system.

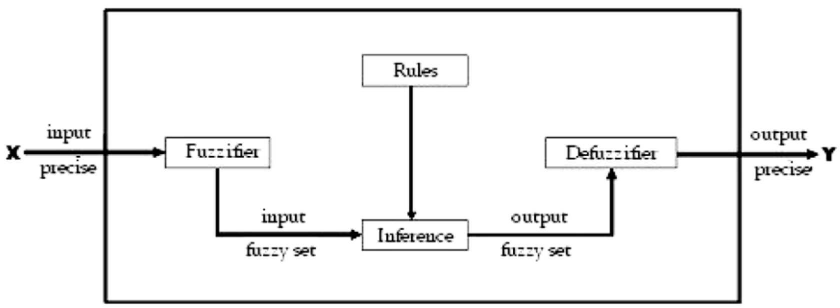


**Fig. 3.** Basic structure of Type-1 fuzzy inference system.

and have been developed looking to model the form as the brain manipulates inexact information. Type-2 fuzzy sets are used to model uncertainty and imprecision; originally they were proposed by Zadeh in 1975 and they are essentially "fuzzy-fuzzy" sets in which the membership degrees are type-1 fuzzy sets [15–19].

– *Type-2 fuzzy inference system*:

A fuzzy inference system is a system based on rules that uses fuzzy logic, instead of Boolean logic, to analyze data [5,**?**]. Its basic structure includes four main components:

- *Fuzzifier*. It translates inputs (real values) to fuzzy values.
- *Inference system*. Type-1 or type-2 applies a mechanism of fuzzy reasoning to obtain a fuzzy output.
- *Defuzzifier/type reducer*. The defuzzifier it translates an output to precise values; the type reducer transforms a fuzzy set of type-2 to type-1; and
- *Knowledge base*. It contains a set of fuzzy rules, known as base rules, and a set of membership functions known as the data base.

In Fig. 4 we can appreciate the basic structure of a type-2 fuzzy inference system.

Uncertainty is "the imperfection in the knowledge on the state or the processes of the nature". The statistical uncertainty is "the randomness or the originating error of several sources like described when using statistical methodology" [2].

## 5. Genetic algorithms

The genetic algorithm (GA), is a search technique based on Darwin's theory of evolution, and has received tremendous popularity anywhere in the world during the past few years [3]. They are adaptive methods that can be used to solve search and optimization problems. They are based on the genetic process of the living organisms. Throughout the generations, the populations evolve in the same form as in nature, with the principles of natural selection and the survival of fittest, postulated by Darwin. Simulating evolution, GA's are able to create solutions to problems of the real world. The evolution of these solutions towards optimal values of the problem depends largely on a suitable codification of the solutions. The use of new representations and the construction of new operators to manipulate information have caused that the present conception of a GA is quite different and more general than the original idea. The basic structure of a GA is shown in Fig. 5 [4].

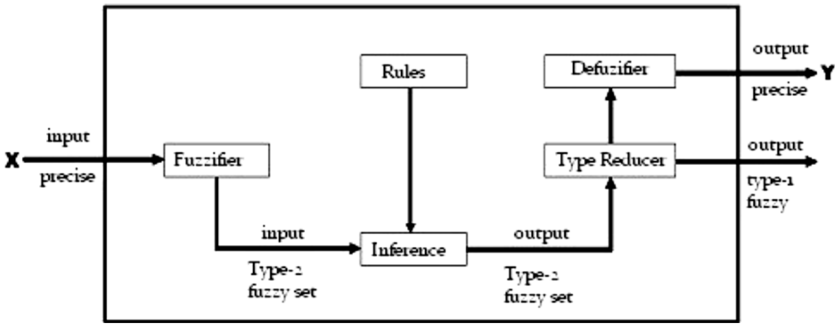


**Fig. 4.** Basic structure of Type-2 fuzzy inference system.

```
Procedure Genetic Algorithm

Begin
  t=0
  to initialize P(t)
  to evaluate P(t)
  While (the condition of shutdown is not fulfilled) do
       begin
            t=t+1
            to select P(t) from P(t-1)
            to apply crossover and mutation on P(t)
            to evaluate P(t)
       end
end
```

**Fig. 5.** Structure of a generational GA.

## 6. General idea of the modular architecture

In this paper we are making a comparison between the integration methods of a modular neural network; this comparison is done with the integration method as fuzzy systems, which used techniques of type-1 fuzzy logic and type-2 fuzzy logic. These methods of integration were implemented in modular neural networks for biometry. In others words, the MNN's were trained to make the recognition of persons using their face, fingerprint and voice. These features of a persons are the 3 more important biometric measures in the field of pattern recognition, at the moment [8]. Next, we show the general architecture of the modular system in Fig. 6.

Now we describe in more detail the pattern recognition system (Fig. 7).

We describe in more detail the modular of the pattern recognition system in the following section.

### *6.1. Description of the problem and the solution*

In this section we describe in detail the different parts of the recognition system.

### *6.2. Input data*

We describe at this point the input data used in the modular architecture for pattern recognition:

*Face*: Images of the faces of 30 different people were used to make the training of the MNN without noise, we also use 30 images of the face of these same people but with different gestures, to use them in the training with noise. The used images were preprocessed with a wavelet function to obtain better results in the training. These images were obtained from a group of students of Tijuana Institute of Technology, combined with some others of the ORL data base. The size of these images is of $268 \times 338$ pixels with extension .bmp. These are shown in Fig. 8.

*Fingerprint*: Images of the fingerprints of 30 different people were used to make the training without noise. Then it was added random noise to the fingerprint use them in the training with noise. The used images were preprocessed with a

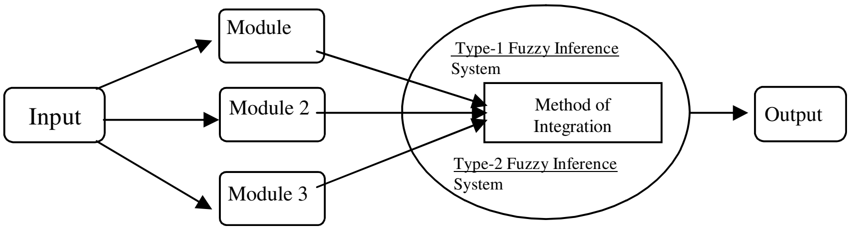


**Fig. 6.** General Scheme of the pattern recognition system.

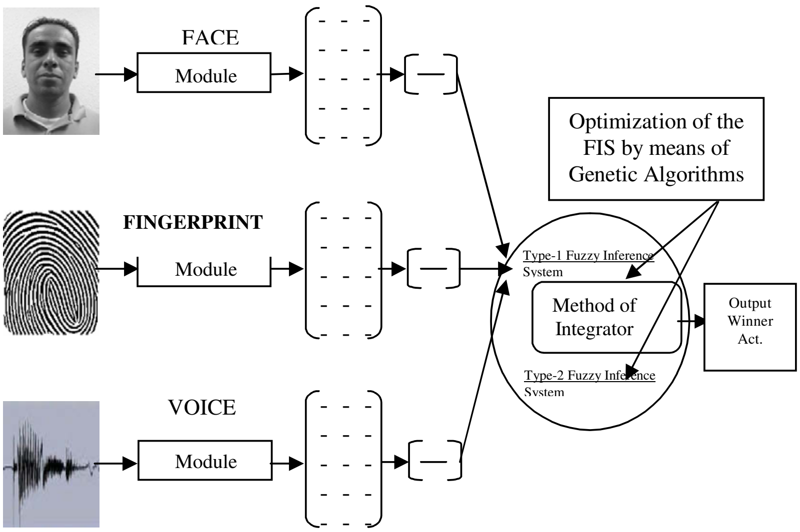


**Fig. 7.** Complete structure of the modular approach for pattern recognition.

**Fig. 8.** Data base used for the training of faces.

wavelet function to obtain better results in the training, as given in. These images were obtained from a group of students of Tijuana Institute of Technology, combined with some others of the ORL data base. The size of these images is of 268 × 338 pixels with extension .bmp. These are shown in Fig. 9.

*Voice*: For the training of module of voice was used word spoken by different persons. With samples of 30 persons as with the face and fingerprint. We applied the Mel cepstrals coefficients, as preprocessing for the training in the MNN.

The used three Spanish words as follows:

- Accesar
- Hola
- Presención

Where some people will say the words in Spanish "Accesar", others "Hola", and some other "Presentación".

We also have to mention that random noise was added to the voice signals to train the MNN with noisy signals.

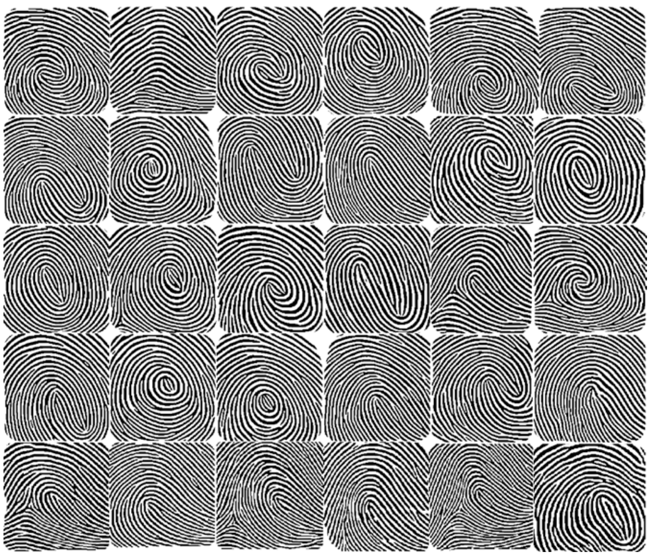

**Fig. 9.** Data base used for the training of fingerprints.

### 6.3. Modules of the MNN for the training

The modular neural network that is considered in this paper has three modules, one for the face, another one for the fingerprint and finally another for the voice, each of the three modules has three submodules. In others words, the architecture of the modules could be visualized as in Fig. 10.

It is possible to mention that for each trained module and each submodule, different architectures were used, that is to say, different number of neurons, layers, etc., and different training methods.

### 6.4. Output of the modular neural network (MNN)

The output of the MNN is a vector that is formed by 30 activations (in this case because the network has been trained with 30 different people).

### 6.5. Competition between activations

In this step the competition between the 30 winning activations is carried out, thus the final result from the MNN is obtained; where it is an activation by module, these activations were used to input the fuzzy system, this data represented then the higher activation from a each module, in others words, three results were obtained; one for the module of the face, another one for module of the fingerprint and another one for the module of the voice.

### 6.6. Integration of the MNN (Type-1 and type-2 fuzzy inference system)

In this part, once the winning activations are obtained for each module, they are input to the fuzzy inference system, in which these activations are evaluated and depending on the characteristics of the fuzzy system, a result of final output is obtained, which will give us, the winning module.

### 6.7. Final output (Winning activation)

The result of the fuzzy system will give us to which module belongs, once this data is obtained we will be able to know that a person has been recognized.

### 6.8. Optimization of the fuzzy systems by means of genetic algorithms

The optimization of the fuzzy systems consists of, as the name indicates it, optimizing the membership functions of the fuzzy systems, for the type-1 and type-2 fuzzy logic, with genetic algorithms.

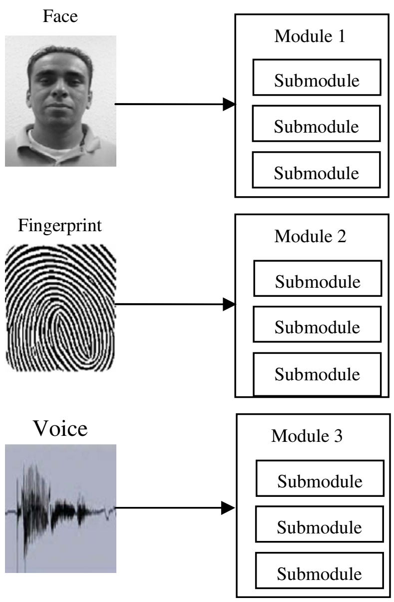


**Fig. 10.** Architecture of the modular neural network for the training.

## 7. Results

The simulation results were obtained after several steps were carried out. First, a set of trainings with the MNN were done. Second, several fuzzy systems (type-1 and type-2) were developed by using a genetic algorithm. The data used for training was obtained from previous research work and was also used to test the fuzzy integration modules.

Once we collected the data, the training phase of MNN with different architectures, was initiated, this with the purpose of being able to make comparisons between the different tests. We choose five of them (see Table 1) to test of integration module with the type-1 and type-2 fuzzy systems optimized with GA' s and to obtain a comparison of the results.

After we obtained the necessary trainings of the MNN's, the genetic algorithm was used to obtain the type-1 fuzzy inference system with triangular membership functions and to use it then as integration method, therefore it was optimized with the genetic algorithm that allowed to obtain the type-1 fuzzy inference system with trapezoidal membership functions, and then with the genetic algorithm to obtain the type-1 fuzzy inference system with Gaussian membership functions and to test the different methods to integrate the results given by the MNN's.

We show in Table 1 the training of the modular neural networks for response integration.

### 7.1. Type-1 fuzzy inference system

As was mentioned previously several type-1 fuzzy inference systems were build, using Triangular, Trapezoidal and Gaussian membership functions; next we show the obtained fuzzy systems optimized by the GA. It is worth mentioning that not all the systems that were obtained are shown here because they were too many; therefore we show only one of each membership function.

**Table 1**
Training results of the modular neural networks

| MNN | Training method | Layers by submodules | Neurons by layers | Performance function | Goal error | Error | Epochs | Recognized persons | % of recognition | Times of training |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | Trainscg | R: SubMod1: 2 | 400,150 | MSE | 0.01 | 0.00999066 | 4000 | 30 | 100 | 3 h 50 min |
| | | SubMod2: 2 | 420,100 | MSE | 0.01 | 0.00996782 | 4000 | 30 | 100 | |
| | | SubMod3: 2 | 410,125 | MSE | 0.01 | 0.00997436 | 4000 | 30 | 100 | |
| | | H: SubMod4: 2 | 350,130 | MSE | 0.01 | 0.00999281 | 4000 | 29 | 97 | |
| | | SubMod5: 2 | 250,140 | MSE | 0.01 | 0.00999318 | 4000 | 29 | 97 | |
| | | SubMod6: 2 | 300,135 | MSE | 0.01 | 0.00999893 | 4000 | 30 | 100 | |
| | | V: SubMod7: 2 | 85,95 | MSE | 0.001 | 0.00476453 | 3000 | 7 | 23 | |
| | | SubMod8: 2 | 85,90 | MSE | 0.001 | 0.000978912 | 3000 | 28 | 93 | |
| | | SubMod9: 2 | 90,90 | MSE | 0.001 | 0.000987834 | 3000 | 28 | 93 | |
| 3 | Trainscg | R: SubMod1: 2 | 200,160 | MSE | 0.01 | 0.00989252 | 4000 | 30 | 100 | 2 h 57 min |
| | | SubMod2: 2 | 300,150 | MSE | 0.01 | 0.00969909 | 4000 | 30 | 100 | |
| | | SubMod3: 2 | 320,85 | MSE | 0.01 | 0.00996425 | 4000 | 30 | 100 | |
| | | H: SubMod4: 2 | 77,53 | MSE | 0.01 | 0.0295881 | 4000 | 20 | 67 | |
| | | SubMod5: 2 | 72,61 | MSE | 0.01 | 0.0199445 | 4000 | 20 | 67 | |
| | | SubMod6: 2 | 74,53 | MSE | 0.01 | 0.00999007 | 4000 | 27 | 90 | |
| | | V: SubMod7: 2 | 40,55 | MSE | 0.001 | 0.000984098 | 3000 | 29 | 97 | |
| | | SubMod8: 2 | 38,58 | MSE | 0.001 | 0.00099729 | 3000 | 25 | 83 | |
| | | SubMod9: 2 | 42,56 | MSE | 0.001 | 0.000987742 | 3000 | 28 | 93 | |
| 4 | Trainscg | R: SubMod1: 2 | 250,150 | MSE | 0.01 | 0.0099579 | 4000 | 30 | 100 | 1 h 59 min |
| | | SubMod2: 2 | 310,100 | MSE | 0.01 | 0.00996268 | 4000 | 30 | 100 | |
| | | SubMod3: 2 | 315,90 | MSE | 0.01 | 0.00984851 | 4000 | 30 | 100 | |
| | | H: SubMod4: 2 | 15,25 | MSE | 0.01 | 0.170638 | 4000 | 1 | 3 | |
| | | SubMod5: 2 | 15,20 | MSE | 0.01 | 0.175748 | 4000 | 1 | 3 | |
| | | SubMod6: 2 | 15,30 | MSE | 0.01 | 0.0873482 | 4000 | 1 | 3 | |
| | | V: SubMod7: 2 | 45,50 | MSE | 0.001 | 0.000979182 | 3000 | 27 | 90 | |
| | | SubMod8: 2 | 47,55 | MSE | 0.001 | 0.000972833 | 3000 | 27 | 90 | |
| | | SubMod9: 2 | 25,30 | MSE | 0.001 | 0.00167415 | 3000 | 17 | 57 | |
| 5 | Trainscg | R: SubMod1: 2 | 40,30 | MSE | 0.01 | 0.000999915 | 4000 | 26 | 87 | 3 h 50 min |
| | | SubMod2: 2 | 30,20 | MSE | 0.01 | 0.00116395 | 4000 | 3 | 10 | |
| | | SubMod3: 2 | 25,35 | MSE | 0.01 | 0.0318143 | 4000 | 1 | 3 | |
| | | H: SubMod4: 2 | 23,38 | MSE | 0.01 | 0.0650265 | 4000 | 1 | 3 | |
| | | SubMod5: 2 | 10,42 | MSE | 0.01 | 0.473676 | 4000 | 1 | 3 | |
| | | SubMod6: 2 | 15,33 | MSE | 0.01 | 0.653342 | 4000 | 6 | 20 | |
| | | V: SubMod7: 2 | 25,82 | MSE | 0.001 | 0.128213 | 3000 | 21 | 70 | |
| | | SubMod8: 2 | 5,200 | MSE | 0.001 | 0.0648562 | 3000 | 15 | 50 | |
| | | SubMod9: 2 | 10,8 | MSE | 0.001 | 0.0281692 | 3000 | 0 | 0 | |

*7.1.1. Type-1 fuzzy system*

The type-1 fuzzy inference system shown in Fig. 11 is a fuzzy system that has three inputs (activation of the face, activation of the fingerprint and activation of the voice), which are composed by three membership functions each; and an output that defines the winning activation after going through the Mamdani motor of inference.

As it was already mentioned different types of membership functions were used, therefore it is possible to observe next the different optimized systems created by the GA. It can be noticed that different values of the membership functions are obtained for each type; and this is true well as for the inputs as for the outputs of each fuzzy system.

*7.1.1.1. Triangular type-1 fuzzy system.* In the previous Figs. 12–15, it is possible to appreciate the values that the parameters from each triangular membership function for each variable, as well as inputs as the output, for the type-1 fuzzy system. In this case the GA was used with a population of 55 individuals, a maximum of generations of 100, mutation of 0.001, crossover of 0.6 of simple point type; which lasted of 2 min and it was stopped at generation 12, the obtained error was 0.000017082.

*7.1.1.2. Trapezoidal type-1 fuzzy system.* In the previous Figs. 16–19, it is possible to appreciate the values that the parameters from each trapezoidal membership function for each variable, as well as of inputs as the output, for the type-1 fuzzy system. In this case the GA was used with a population of 85 individuals, a maximum of generations of 100, mutation of 0.0001, crossover of 0.25 of simple point type; which lasted of 12 min and it was stopped at generation 36, the obtained error was 0.000046453.

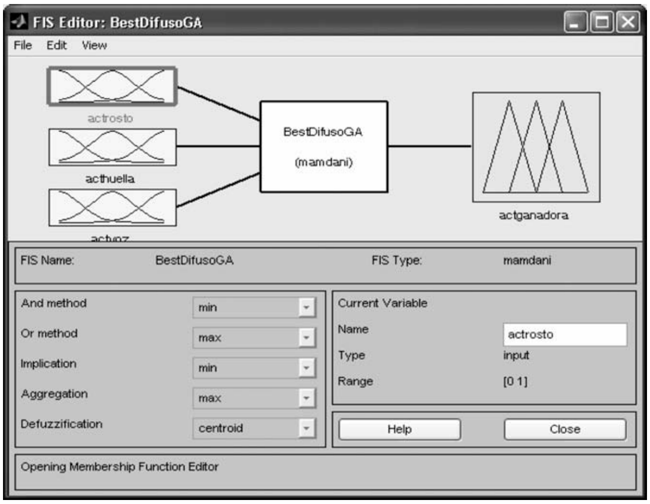


**Fig. 11.** Graphical representation of the Type-1 fuzzy inference system with its inputs and outputs.

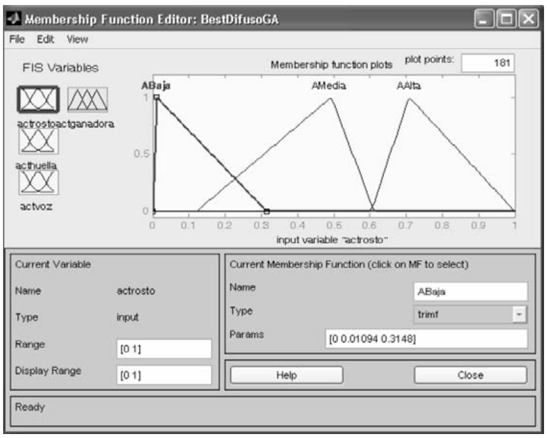


**Fig. 12.** First input variable (higher activation of the face).

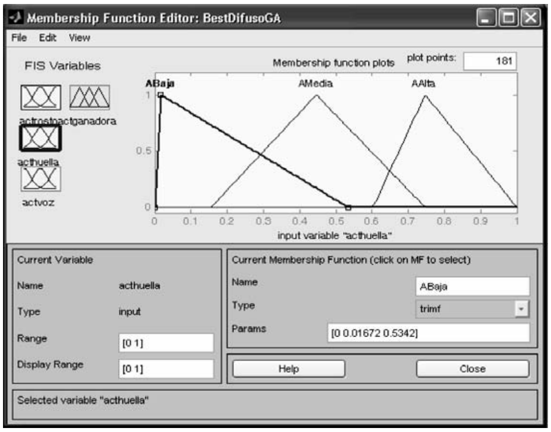


**Fig. 13.** Second input variable (higher activation of the fingerprint).

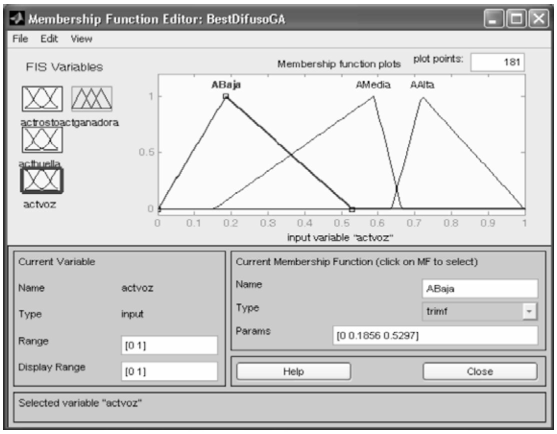


**Fig. 14.** Third input variable (higher activation of the voice).

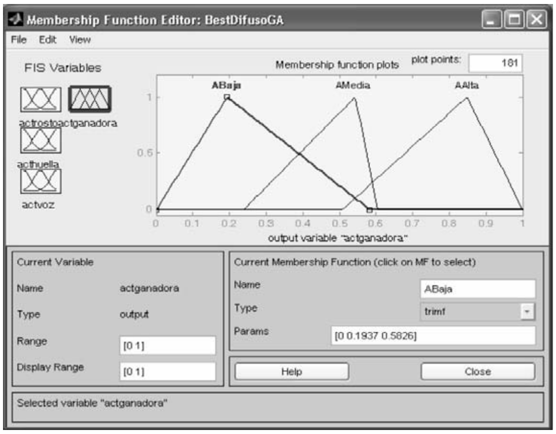


**Fig. 15.** Variable of output (winner activation).

*7.1.1.3. Gaussian type-1 fuzzy system.* In Figs. 16–19, it is possible to appreciate the values that the parameters from each Gaussian membership function for each variable, as well as inputs as the output, for the type-1 fuzzy system. In this case the GA was used with a population of 50 individuals, a maximum of generations of 100, mutation of 0.001, crossover of

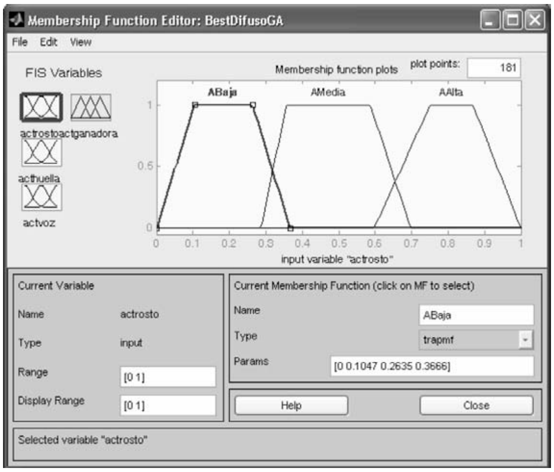


**Fig. 16.** First input variable (higher activation of the face).

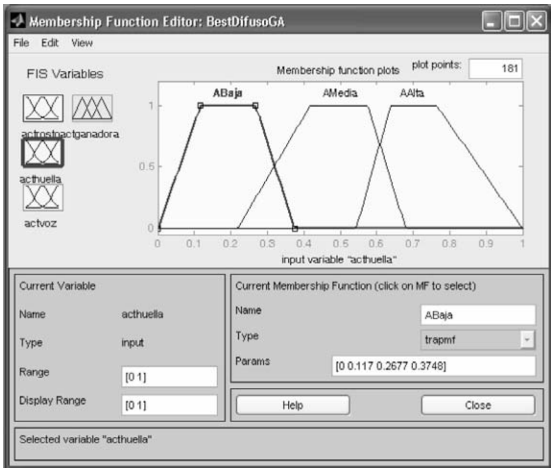


**Fig. 17.** Second input variable (higher activation of the fingerprint).

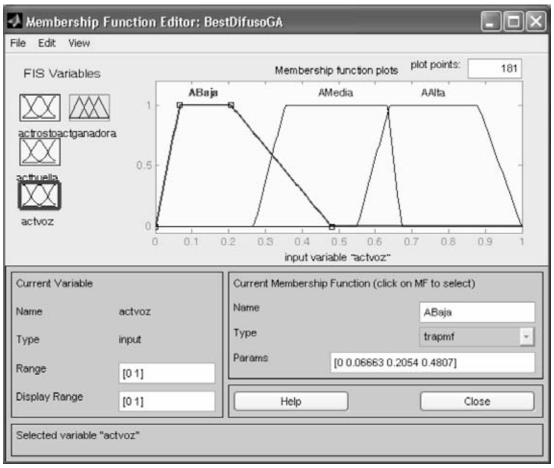


**Fig. 18.** Third input variable (higher activation of the voice).

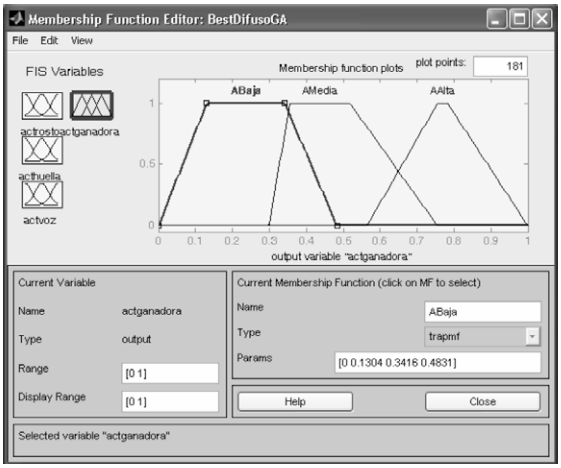

**Fig. 19.** Variable of output (winner activation).

0.65 of multipoint type; which lasted of 4 min and it was stopped at generation 21, the obtained error was 0.00011068.We have to mention that the previously shown fuzzy systems are the ones with the smallest errors. In Tables 2–4 we show all the obtained results (see Figs. 20–23).

### 7.2. Type-2 fuzzy inference system

As was mentioned previously several type-2 fuzzy inference systems were build, using triangular, trapezoidal and Gaussian membership functions; next we show the obtained fuzzy systems optimized by the GA. It is worth mentioning that not all the systems that were obtained are shown here because they were too many; therefore we show only one of each membership function.

#### 7.2.1. Type-2 fuzzy system

The type-2 fuzzy inference system shown in Fig. 24 is a fuzzy system that has three inputs (activation of the face, activation of the fingerprint and activation of the voice), which are composed by three membership functions each; and an output that defines the winning activation after going through the Mamdani motor of inference.

As it was already mentioned different types of membership functions were used, therefore it is possible to observe next the different optimized systems created by the GA. It can be noticed that different values of the membership functions are obtained for each type; and this is true well as for the inputs as for the outputs of each fuzzy system.

**Table 2**
Results of the GA applied Type-1 fuzzy inference systems with triangular memberships functions

| No. | Type MF's | Ind | Max Gen | Mutation | TypeMut | Crossover | TypeCross | GGAP | Stopped Gen | Lasted of GA | Best error |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Triangular | 100 | 150 | 0.0001 | mutbga | 0.1 | xovmp | 0.85 | 29 | 14 min | 0.000673922 |
| 2 | Triangular | 90 | 100 | 0.0001 | mutbga | 0.2 | xovmp | 0.85 | 100 | 35 min | 6.617518707 |
| 3 | Triangular | 80 | 100 | 0.0001 | mutbga | 0.3 | xovmp | 0.85 | 15 | 5 min | 0.000038401 |
| 4 | Triangular | 70 | 100 | 0.0001 | mutbga | 0.5 | xovmp | 0.85 | 31 | 28 min | 1.130730602 |
| 5 | Triangular | 60 | 100 | 0.0001 | mutbga | 0.45 | xovmp | 0.85 | 43 | 10 min | 0.0004582 |
| 6 | Triangular | 50 | 100 | 0.001 | mutbga | 0.65 | xovmp | 0.85 | 100 | 20 min | 0.00020151 |
| 7 | Triangular | 40 | 100 | 0.001 | mutbga | 0.7 | xovsp | 0.85 | 7 | 2 min | 0.000075054 |
| 8 | Triangular | 30 | 100 | 0.01 | mutbga | 0.6 | xovmp | 0.85 | 100 | 10 min | 0.0000014054 |
| 9 | Triangular | 20 | 100 | 0.01 | mutbga | 0.8 | xovsp | 0.85 | 10 | 1 min | 0.00050029 |
| 10 | Triangular | 10 | 100 | 0.1 | mutbga | 0.9 | xovsp | 0.85 | 4 | 10 s | 0.00522077 |
| 11 | Triangular | 5 | 100 | 0.1 | mutbga | 1 | xovsp | 0.85 | 11 | 15 s | 0.00263167 |
| 12 | Triangular | 15 | 100 | 0.1 | mutbga | 0.95 | xovsp | 0.85 | 8 | 27 s | 0.00123392 |
| 13 | Triangular | 25 | 100 | 0.01 | mutbga | 0.85 | xovsp | 0.85 | 26 | 20 s | 0.000073374 |
| 14 | Triangular | 35 | 100 | 0.01 | mutbga | 0.65 | xovmp | 0.85 | 96 | 12 min | 0.00040778 |
| 15 | Triangular | 45 | 100 | 0.001 | mutbga | 0.75 | xovsp | 0.85 | 19 | 3 min | 0.00104095 |
| **16** | **Triangular** | **55** | **100** | **0.001** | **mutbga** | **0.6** | **xovsp** | **0.85** | **12** | **2 min** | **0.000017082** |
| 17 | Triangular | 65 | 100 | 0.001 | mutbga | 0.4 | xovsp | 0.85 | 16 | 4 min | 0.0000412577 |
| 18 | Triangular | 75 | 100 | 0.0001 | mutbga | 0.35 | xovsp | 0.85 | 99 | 27 min | 0.000085732 |
| 19 | Triangular | 85 | 100 | 0.0001 | mutbga | 0.25 | xovsp | 0.85 | 18 | 6 min | 0.00119261 |
| 20 | Triangular | 95 | 100 | 0.0001 | mutbga | 0.15 | xovsp | 0.85 | 100 | 37 min | 0.02178055 |

**Table 3**
Results of the GA applied Type-1 fuzzy inference system with trapezoidal memberships functions

| No. | Type MF's | Ind | Max Gen | Mutation | TypeMut | Crossover | TypeCross. | GGAP | Stopped Gen | Lasted of GA | Best error |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Trapezoidal | 100 | 150 | 0.0001 | mutbga | 0.1 | xovsp | 0.85 | 42 | 16 min | 0.00099803 |
| 2 | Trapezoidal | 90 | 100 | 0.0001 | mutbga | 0.2 | xovmp | 0.85 | 100 | 33 min | 0.000164958 |
| 3 | Trapezoidal | 80 | 100 | 0.0001 | mutbga | 0.3 | xovmp | 0.85 | 100 | 31 min | 0.00203463 |
| 4 | Trapezoidal | 70 | 100 | 0.0001 | mutbga | 0.5 | xovmp | 0.85 | 100 | 28 min | 0.01164289 |
| 5 | Trapezoidal | 60 | 100 | 0.0001 | mutbga | 0.45 | xovmp | 0.85 | 100 | 23 min | 0.00162315 |
| 6 | Trapezoidal | 50 | 100 | 0.001 | mutbga | 0.65 | xovmp | 0.85 | 15 | 3 min | 0.00019062 |
| 7 | Trapezoidal | 40 | 100 | 0.001 | mutbga | 0.7 | xovsp | 0.85 | 13 | 2 min | 0.00075258 |
| 8 | Trapezoidal | 30 | 100 | 0.01 | mutbga | 0.6 | xovmp | 0.85 | 13 | 1 min | 0.00113649 |
| 9 | Trapezoidal | 20 | 100 | 0.01 | mutbga | 0.8 | xovsp | 0.85 | 100 | 8 min | 0.00158774 |
| 10 | Trapezoidal | 10 | 100 | 0.1 | mutbga | 0.9 | xovsp | 0.85 | 10 | 1 min | 0.0002004 |
| 11 | Trapezoidal | 5 | 100 | 0.1 | mutbga | 1 | xovsp | 0.85 | 100 | 2 min | 4.66257156 |
| 12 | Trapezoidal | 15 | 100 | 0.1 | mutbga | 0.95 | xovsp | 0.85 | 100 | 6 min | 0.00331437 |
| 13 | Trapezoidal | 25 | 100 | 0.01 | mutbga | 0.85 | xovsp | 0.85 | 100 | 10 min | 0.01206595 |
| 14 | Trapezoidal | 35 | 100 | 0.01 | mutbga | 0.65 | xovmp | 0.85 | 100 | 13 min | 0.00456134 |
| 15 | Trapezoidal | 45 | 100 | 0.001 | mutbga | 0.75 | xovsp | 0.85 | 100 | 6 min | 0.00254746 |
| 16 | Trapezoidal | 55 | 100 | 0.001 | mutbga | 0.6 | xovsp | 0.85 | 100 | 21 min | 0.00536313 |
| 17 | Trapezoidal | 65 | 100 | 0.001 | mutbga | 0.4 | xovsp | 0.85 | 100 | 25 min | 1.243885488 |
| 18 | Trapezoidal | 75 | 100 | 0.0001 | mutbga | 0.35 | xovsp | 0.85 | 35 | 10 min | 0.001008539 |
| **19** | **Trapezoidal** | **85** | **100** | **0.0001** | **mutbga** | **0.25** | **xovsp** | **0.85** | **36** | **12 min** | **0.000046453** |
| 20 | Trapezoidal | 95 | 100 | 0.0001 | mutbga | 0.15 | xovsp | 0.85 | 100 | 17 min | 0.01042749 |

**Table 4**
Results of the GA applied Type-1 fuzzy inference system with Gaussian memberships functions

| No. | Type MF's | Ind | Max Gen | Mutation | TypeMut | Crossover | TypeCross | GGAP | Stopped Gen | Lasted of GA | Best error |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Gaussian | 100 | 150 | 0.0001 | mutbga | 0.1 | xovmp | 0.85 | 3 | 1 min | 0.000164788 |
| 2 | Gaussian | 90 | 100 | 0.0001 | mutbga | 0.2 | xovmp | 0.85 | 41 | 15 min | 0.183854323 |
| 3 | Gaussian | 80 | 100 | 0.0001 | mutbga | 0.3 | xovmp | 0.85 | 5 | 2 min | 0.00181081 |
| 4 | Gaussian | 70 | 100 | 0.0001 | mutbga | 0.5 | xovmp | 0.85 | 100 | 26 min | 0.000615959 |
| 5 | Gaussian | 60 | 100 | 0.0001 | mutbga | 0.45 | xovmp | 0.85 | 10 | 4 min | 0.02483356 |
| **6** | **Gaussian** | **50** | **100** | **0.001** | **mutbga** | **0.65** | **xovmp** | **0.85** | **21** | **4 min** | **0.00011068** |
| 7 | Gaussian | 40 | 100 | 0.001 | mutbga | 0.7 | xovsp | 0.85 | 53 | 9 min | 0.06011866 |
| 8 | Gaussian | 30 | 100 | 0.01 | mutbga | 0.6 | xovmp | 0.85 | 5 | 34 s | 0.03585971 |
| 9 | Gaussian | 20 | 100 | 0.01 | mutbga | 0.8 | xovsp | 0.85 | 17 | 1 min | 0.006970346 |
| 10 | Gaussian | 10 | 100 | 0.1 | mutbga | 0.9 | xovsp | 0.85 | 5 | 12 s | 0.04190672 |
| 11 | Gaussian | 5 | 100 | 0.1 | mutbga | 1 | xovsp | 0.85 | 3 | 5 s | 0.0045917 |
| 12 | Gaussian | 15 | 100 | 0.1 | mutbga | 0.95 | xovsp | 0.85 | 25 | 1 min | 0.26194236 |
| 13 | Gaussian | 25 | 100 | 0.01 | mutbga | 0.85 | xovsp | 0.85 | 16 | 1 min | 0.10640646 |
| 14 | Gaussian | 35 | 100 | 0.01 | mutbga | 0.65 | xovmp | 0.85 | 5 | 46 s | 0.00346566 |
| 15 | Gaussian | 45 | 100 | 0.001 | mutbga | 0.75 | xovsp | 0.85 | 3 | 30 s | 0.00036119 |
| 16 | Gaussian | 55 | 100 | 0.001 | mutbga | 0.6 | xovsp | 0.85 | 100 | 21 min | 0.0474117 |
| 17 | Gaussian | 65 | 100 | 0.001 | mutbga | 0.4 | xovsp | 0.85 | 100 | 24 min | 0.01250863 |
| 18 | Gaussian | 75 | 100 | 0.0001 | mutbga | 0.35 | xovsp | 0.85 | 2 | 36 s | 0.000164788 |
| 19 | Gaussian | 85 | 100 | 0.0001 | mutbga | 0.25 | xovsp | 0.85 | 11 | 4 min | 0.00444927 |
| 20 | Gaussian | 95 | 100 | 0.0001 | mutbga | 0.15 | xovsp | 0.85 | 100 | 37 min | 0.03247838 |

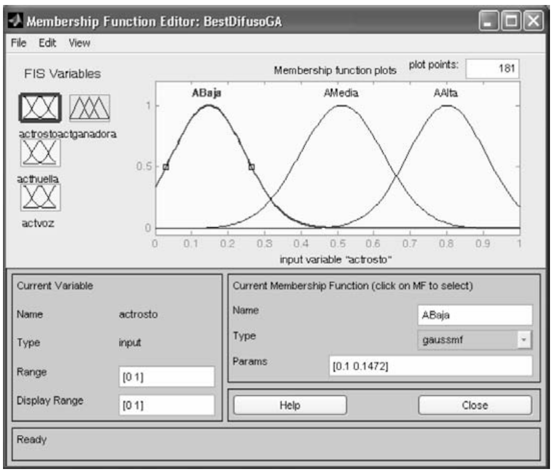


**Fig. 20.** First input variable (higher activation of the face).

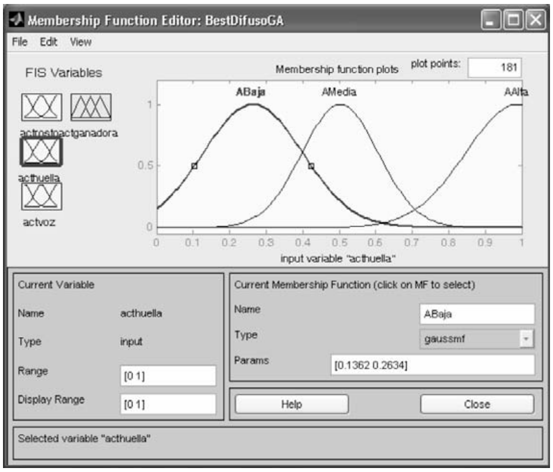


**Fig. 21.** Second input variable (higher activation of the fingerprint).

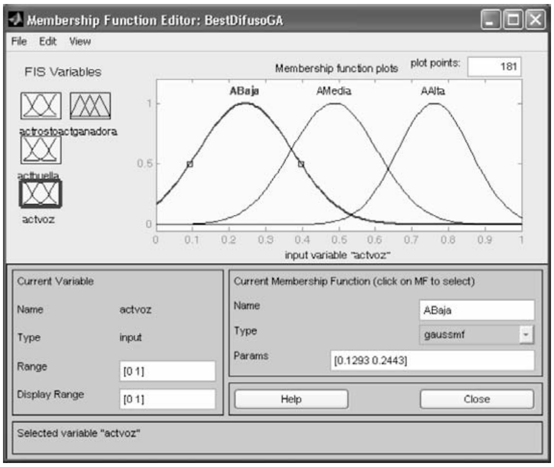


**Fig. 22.** Third input variable (higher activation of the voice).

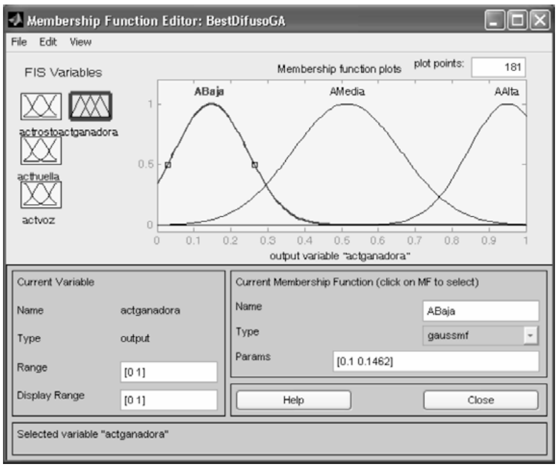


**Fig. 23.** Variable of output (winner activation).

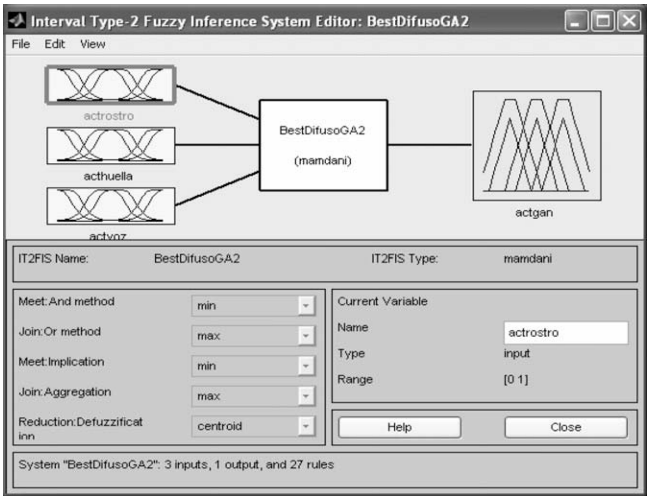


**Fig. 24.** Graphical representation of the Type-2 fuzzy inference system with its inputs and outputs.

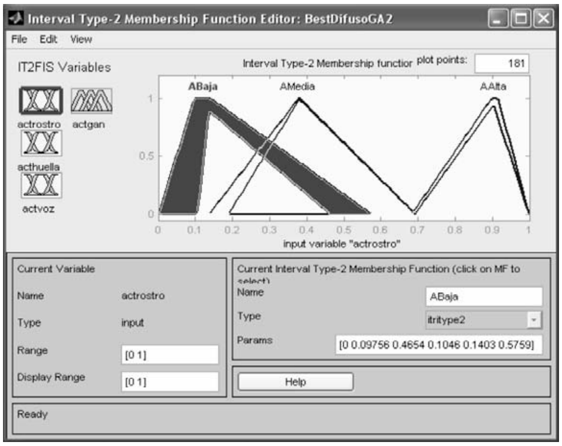


**Fig. 25.** First input variable (higher activation of the face).

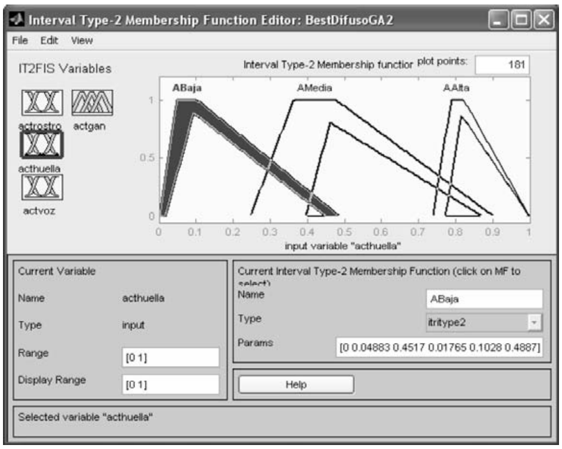


**Fig. 26.** Second input variable (higher activation of the fingerprint).

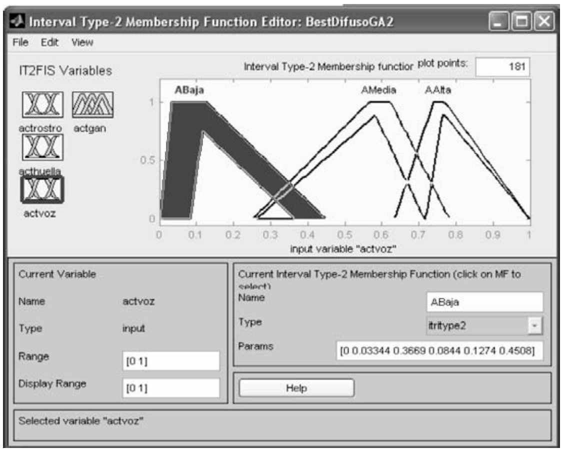


**Fig. 27.** Third input variable (higher activation of the voice).

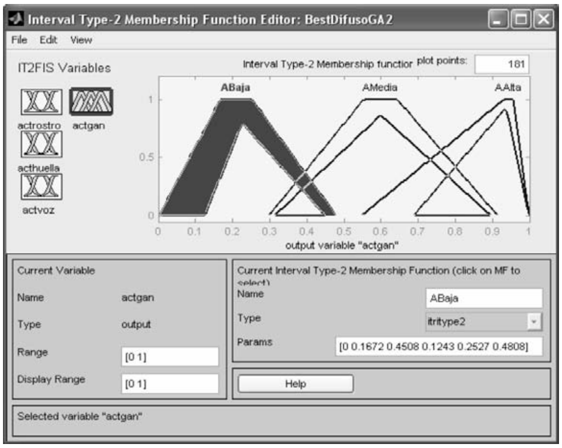


**Fig. 28.** Variable of output (winner activation).

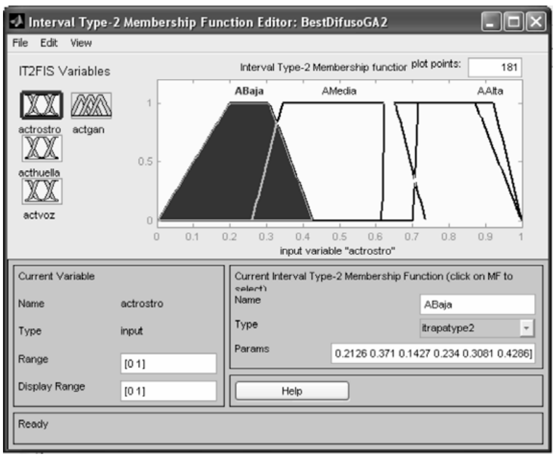


**Fig. 29.** First input variable (higher activation of the face).

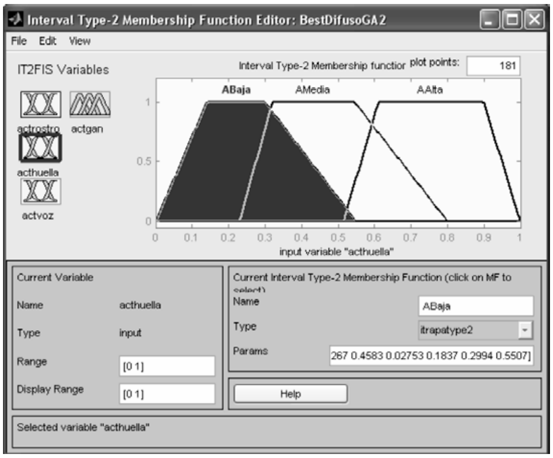


**Fig. 30.** Second input variable (higher activation of the fingerprint).

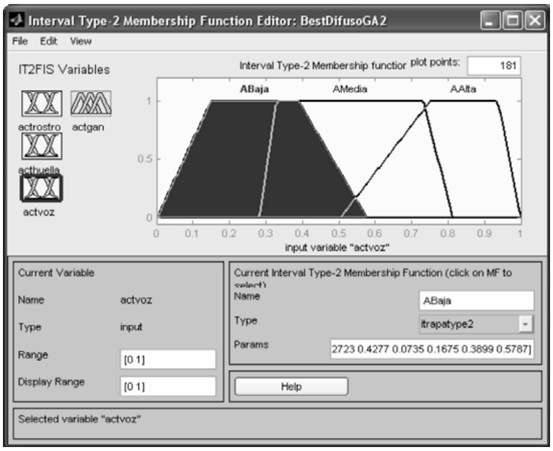


**Fig. 31.** Third input variable (higher activation of the voice).

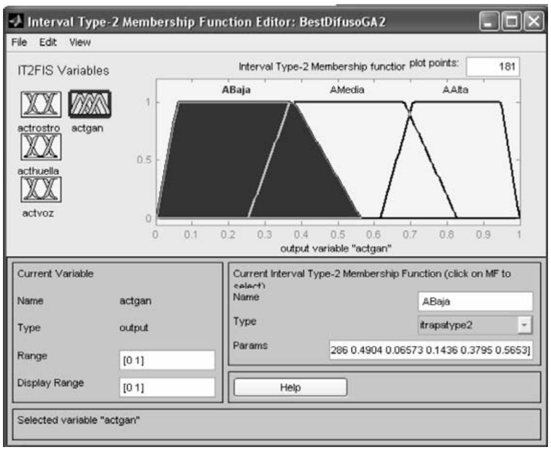


**Fig. 32.** Variable of output (winner activation).

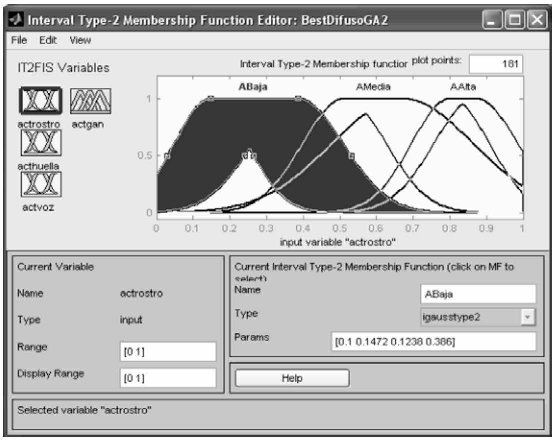


**Fig. 33.** First input variable (higher activation of the face).

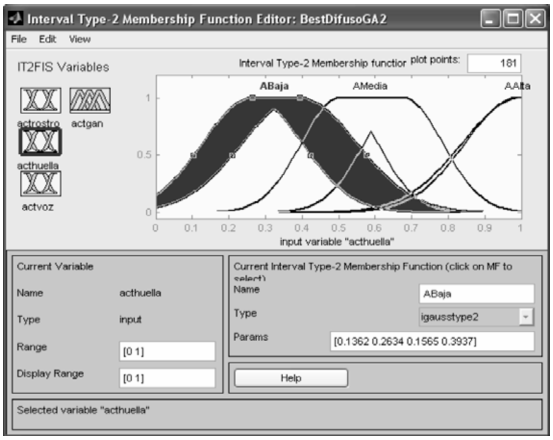


**Fig. 34.** Second input variable (higher activation of the fingerprint).

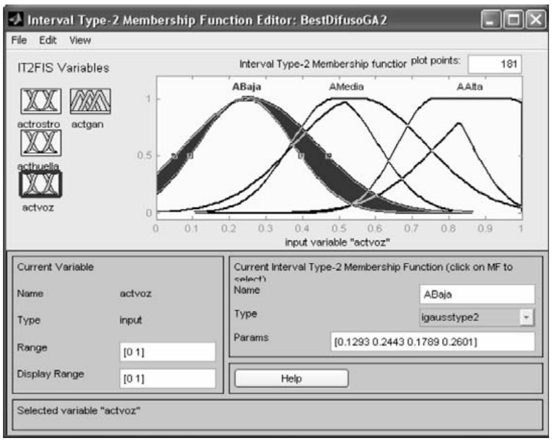


**Fig. 35.** Third input variable (higher activation of the voice).

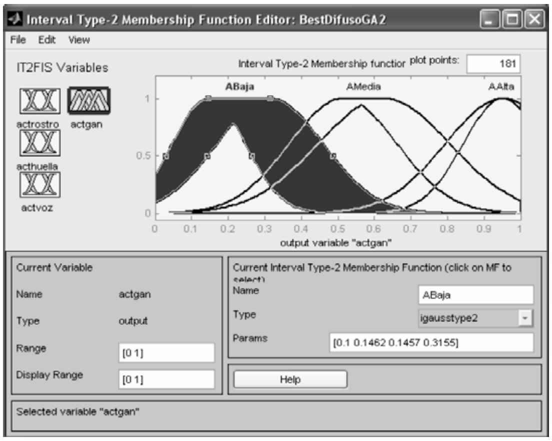


**Fig. 36.** Variable of output (winner activation).

**Table 5**
Results of the GA applied Type-2 fuzzy inference systems with triangular memberships functions

| No. | Type MF's | Ind | Max Gen | Mutation | TypeMut | Crossover | TypeCross | GGAP | Stopped Gen | Lasted of GA | Best error |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Triangular | 100 | 150 | 0.0001 | mutbga | 0.1 | xovmp | 0.85 | 150 | 1 h | 0.243493825 |
| 2 | Triangular | 90 | 100 | 0.0001 | mutbga | 0.2 | xovmp | 0.85 | 14 | 4 min | 0.001130642 |
| 3 | Triangular | 80 | 100 | 0.0001 | mutbga | 0.3 | xovmp | 0.85 | 100 | 30 min | 0.42196264 |
| 4 | Triangular | 70 | 100 | 0.0001 | mutbga | 0.5 | xovmp | 0.85 | 100 | 29 min | 0.441835768 |
| 5 | Triangular | 60 | 100 | 0.0001 | mutbga | 0.45 | xovmp | 0.85 | 100 | 22 min | 0.22838559 |
| 6 | Triangular | 50 | 100 | 0.001 | mutbga | 0.65 | xovmp | 0.85 | 100 | 20 min | 0.41049033 |
| **7** | **Triangular** | **40** | **100** | **0.001** | **mutbga** | **0.7** | **xovsp** | **0.85** | **51** | **19 min** | **0.000011385** |
| 8 | Triangular | 30 | 100 | 0.01 | mutbga | 0.6 | xovmp | 0.85 | 100 | 8 min | 0.53426876 |
| 9 | Triangular | 20 | 100 | 0.01 | mutbga | 0.8 | xovsp | 0.85 | 100 | 5 min | 0.240057 |
| 10 | Triangular | 10 | 100 | 0.1 | mutbga | 0.9 | xovsp | 0.85 | 100 | 4 min | 0.60376824 |
| 11 | Triangular | 5 | 100 | 0.1 | mutbga | 1 | xovsp | 0.85 | 44 | 1 min | 0.2344504 |
| 12 | Triangular | 15 | 100 | 0.1 | mutbga | 0.95 | xovsp | 0.85 | 100 | 6 min | 0.37926694 |
| 13 | Triangular | 25 | 100 | 0.01 | mutbga | 0.85 | xovsp | 0.85 | 100 | 10 min | 0.04789974 |
| 14 | Triangular | 35 | 100 | 0.01 | mutbga | 0.65 | xovmp | 0.85 | 8 | 1 min | 0.00900691 |
| 15 | Triangular | 45 | 100 | 0.001 | mutbga | 0.75 | xovsp | 0.85 | 100 | 17 min | 0.42580788 |
| 16 | Triangular | 55 | 100 | 0.001 | mutbga | 0.6 | xovsp | 0.85 | 100 | 21 min | 0.01624613 |
| 17 | Triangular | 65 | 100 | 0.001 | mutbga | 0.4 | xovsp | 0.85 | 88 | 25 min | 0.00232443 |
| 18 | Triangular | 75 | 100 | 0.0001 | mutbga | 0.35 | xovsp | 0.85 | 100 | 28 min | 0.559013461 |
| 19 | Triangular | 85 | 100 | 0.0001 | mutbga | 0.25 | xovsp | 0.85 | 100 | 31 min | 0.21063925 |
| 20 | Triangular | 95 | 100 | 0.0001 | mutbga | 0.15 | xovsp | 0.85 | 100 | 39 min | 0.38952551 |

**Table 6**
Results of the GA applied Type-2 fuzzy inference system with trapezoidal memberships functions

| No. | Type MF's | Ind | Max Gen | Mutation | TypeMut | Crossover | TypeCross | GGAP | Stopped Gen | Lasted of GA | Best error |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Trapezoidal | 100 | 150 | 0.0001 | mutbga | 0.1 | xovmp | 0.85 | 150 | 5 h 58 min | 0.542291089 |
| 2 | Trapezoidal | 90 | 100 | 0.0001 | mutbga | 0.2 | xovmp | 0.85 | 100 | 3 h 49 min | 2.325 |
| **3** | **Trapezoidal** | **80** | **100** | **0.0001** | **mutbga** | **0.3** | **xovmp** | **0.85** | **10** | **13 min** | **0.0247836** |
| 4 | Trapezoidal | 70 | 100 | 0.0001 | mutbga | 0.5 | xovmp | 0.85 | 100 | 29 min | 0.682264153 |
| 5 | Trapezoidal | 60 | 100 | 0.0001 | mutbga | 0.45 | xovmp | 0.85 | 100 | 24 min | 0.4822949 |
| 6 | Trapezoidal | 50 | 100 | 0.001 | mutbga | 0.65 | xovmp | 0.85 | 100 | 17 min | 0.73576041 |
| 7 | Trapezoidal | 40 | 100 | 0.001 | mutbga | 0.7 | xovsp | 0.85 | 36 | 34 min | 0.035355 |
| 8 | Trapezoidal | 30 | 100 | 0.01 | mutbga | 0.6 | xovmp | 0.85 | 100 | 12 min | 0.73576041 |
| 9 | Trapezoidal | 20 | 100 | 0.01 | mutbga | 0.8 | xovsp | 0.85 | 100 | 5 min | 0.76300909 |
| 10 | Trapezoidal | 10 | 100 | 0.1 | mutbga | 0.9 | xovsp | 0.85 | 100 | 3 min | 0.65602506 |
| 11 | Trapezoidal | 5 | 100 | 0.1 | mutbga | 1 | xovsp | 0.85 | 100 | 2 min | 0.70884404 |
| 12 | Trapezoidal | 15 | 100 | 0.1 | mutbga | 0.95 | xovsp | 0.85 | 100 | 4 min | 0.60458736 |
| 13 | Trapezoidal | 25 | 100 | 0.01 | mutbga | 0.85 | xovsp | 0.85 | 100 | 7 min | 0.32737116 |
| 14 | Trapezoidal | 35 | 100 | 0.01 | mutbga | 0.65 | xovmp | 0.85 | 100 | 9 min | 0.90410197 |
| 15 | Trapezoidal | 45 | 100 | 0.001 | mutbga | 0.75 | xovsp | 0.85 | 100 | 13 min | 0.60458736 |
| 16 | Trapezoidal | 55 | 100 | 0.001 | mutbga | 0.6 | xovsp | 0.85 | 100 | 15 min | 0.60458736 |
| 17 | Trapezoidal | 65 | 100 | 0.001 | mutbga | 0.4 | xovsp | 0.85 | 100 | 16 min | 0.76300909 |
| 18 | Trapezoidal | 75 | 100 | 0.0001 | mutbga | 0.35 | xovsp | 0.85 | 100 | 29 min | 0.875249036 |
| 19 | Trapezoidal | 85 | 100 | 0.0001 | mutbga | 0.25 | xovsp | 0.85 | 100 | 33 min | 0.81848755 |
| 20 | Trapezoidal | 95 | 100 | 0.0001 | mutbga | 0.15 | xovsp | 0.85 | 100 | 38 min | 0.57939829 |

**Table 7**
Results of the GA applied Type-2 fuzzy inference system with Gaussian memberships functions

| No. | Type MF's | Ind | Max Gen | Mutation | TypeMut | Crossover | TypeCross | GGAP | Stopped Gen | Lasted of GA | Best error |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Gaussian | 100 | 150 | 0.0001 | mutbga | 0.1 | xovmp | 0.85 | 3 | 1 min | 0.003115353 |
| 2 | Gaussian | 90 | 100 | 0.0001 | mutbga | 0.2 | xovmp | 0.85 | 6 | 2 min | 0.001610318 |
| 3 | Gaussian | 80 | 100 | 0.0001 | mutbga | 0.3 | xovmp | 0.85 | 9 | 3 min | 0.00294546 |
| 4 | Gaussian | 70 | 100 | 0.0001 | mutbga | 0.5 | xovmp | 0.85 | 100 | 25 min | 0.0000787919 |
| 5 | Gaussian | 60 | 100 | 0.0001 | mutbga | 0.45 | xovmp | 0.85 | 100 | 21 min | 0.00115187 |
| 6 | Gaussian | 50 | 100 | 0.001 | mutbga | 0.65 | xovmp | 0.85 | 100 | 18 min | 0.0001781 |
| 7 | Gaussian | 40 | 100 | 0.001 | mutbga | 0.7 | xovsp | 0.85 | 4 | 37 s | 0.0000218 |
| 8 | Gaussian | 30 | 100 | 0.01 | mutbga | 0.6 | xovmp | 0.85 | 11 | 2 min | 0.00298097 |
| 9 | Gaussian | 20 | 100 | 0.01 | mutbga | 0.8 | xovsp | 0.85 | 78 | 6 min | 0.00239668 |
| 10 | Gaussian | 10 | 100 | 0.1 | mutbga | 0.9 | xovsp | 0.85 | 3 | 8 s | 0.00681549 |
| 11 | Gaussian | 5 | 100 | 0.1 | mutbga | 1 | xovsp | 0.85 | 2 | 3 s | 0.00031891 |
| 12 | Gaussian | 15 | 100 | 0.1 | mutbga | 0.95 | xovsp | 0.85 | 100 | 5 min | 0.01700403 |
| 13 | Gaussian | 25 | 100 | 0.01 | mutbga | 0.85 | xovsp | 0.85 | 36 | 4 min | 0.00047306 |
| 14 | Gaussian | 35 | 100 | 0.01 | mutbga | 0.65 | xovmp | 0.85 | 15 | 2 min | 0.0000915214 |
| 15 | Gaussian | 45 | 100 | 0.001 | mutbga | 0.75 | xovsp | 0.85 | 100 | 16 min | 0.00089849 |
| **16** | **Gaussian** | **55** | **100** | **0.001** | **mutbga** | **0.6** | **xovsp** | **0.85** | **4** | **50 s** | **0.0000103302** |
| 17 | Gaussian | 65 | 100 | 0.001 | mutbga | 0.4 | xovsp | 0.85 | 20 | 5 min | 0.00135865 |
| 18 | Gaussian | 75 | 100 | 0.0001 | mutbga | 0.35 | xovsp | 0.85 | 25 | 7 min | 0.001623653 |
| 19 | Gaussian | 85 | 100 | 0.0001 | mutbga | 0.25 | xovsp | 0.85 | 8 | 3 min | 0.00373201 |
| 20 | Gaussian | 95 | 100 | 0.0001 | mutbga | 0.15 | xovsp | 0.85 | 14 | 5 min | 0.00031598 |

**Table 8**
Results of the response integration of the MNN's with Type-1 FIS for the best training

| Fuzzy System | % of Recognition for the FIS with triangular MF's | % of Recognition for the FIS with trapezoidal MF's | % of Recognition for the FIS with trapezoidal MF's |
|---|---|---|---|
| *Best training the MNN's* | | | |
| 1 | 100% (30/30) | 100% (30/30) | 100% (30/30) |
| 2 | 100% (30/30) | 100% (30/30) | 100% (30/30) |
| 3 | 100% (30/30) | 100% (30/30) | 100% (30/30) |
| 4 | 100% (30/30) | 100% (30/30) | 100% (30/30) |
| 5 | 100% (30/30) | 100% (30/30) | 100% (30/30) |

**Table 9**
Results of the response integration of the MNN's with Type-1 FIS for the worse training

| Fuzzy System | % of Recognition for the FIS with triangular MF's | % of Recognition for the FIS with trapezoidal MF's | % of recognition for the FIS with trapezoidal MF's |
|---|---|---|---|
| *Worse training the MNN's* | | | |
| 1 | 0% (0/30) | 0% (0/30) | 0% (0/30) |
| 2 | 0% (0/30) | 0% (0/30) | 0% (0/30) |
| 3 | 0% (0/30) | 0% (0/30) | 0% (0/30) |
| 4 | 0% (0/30) | 0% (0/30) | 0% (0/30) |
| 5 | 0% (0/30) | 0% (0/30) | 0% (0/30) |

**Table 10**
Results of the response integration of the MNN's with Type-2 FIS for the best training

| Fuzzy system | % of recognition for the FIS with triangular MF's | % of recognition for the FIS with trapezoidal MF's | % of recognition for the FIS with trapezoidal MF's |
|---|---|---|---|
| *Best training the MNN's* | | | |
| 1 | 100% (30/30) | 100% (30/30) | 100% (30/30) |
| 2 | 100% (30/30) | 100% (30/30) | 100% (30/30) |
| 3 | 100% (30/30) | 100% (30/30) | 100% (30/30) |
| 4 | 100% (30/30) | 100% (30/30) | 100% (30/30) |
| 5 | 100% (30/30) | 100% (30/30) | 100% (30/30) |

*7.2.1.1. Triangular type-2 fuzzy system.* In Figs. 25–28, it is possible to appreciate the values that the parameters from each triangular membership function for each variable, as well as inputs as the output, for the type-2 fuzzy system. In this case the GA was used with a population of 40 individuals, a maximum of generations of 100, mutation of 0.001, crossover of 0.7 of simple point type; which lasted of 19 min and it was stopped at generation 51, the obtained error was 0.000011385.

**Table 11**
Results of the response integration of the MNN's with Type-2 FIS for the worse training

| Fuzzy system | % of recognition for the FIS with triangular MF's | % of recognition for the FIS with trapezoidal MF's | % of recognition for the FIS with trapezoidal MF's |
|---|---|---|---|
| *Worse training the MNN's* | | | |
| 1 | 7% (2/30) | 7% (2/30) | 7% (2/30) |
| 2 | 7% (2/30) | 7% (2/30) | 7% (2/30) |
| 3 | 7% (2/30) | 7% (2/30) | 7% (2/30) |
| 4 | 7% (2/30) | 7% (2/30) | 7% (2/30) |
| 5 | 7% (2/30) | 7% (2/30) | 7% (2/30) |

**Table 12**
Average errors for the fuzzy inference systems with triangular membership functions

| Type-1 | Type-2 |
|---|---|
| *Average error* | |
| 0.389196189 | 0.269979242 |

**Table 13**
Average errors for the fuzzy inference systems with trapezoidal membership functions

| Type-1 | Type-2 |
|---|---|
| *Average error* | |
| 0.298306163 | 0.688638346 |

**Table 14**
Average errors for the fuzzy inference systems with Gaussian membership functions

| Type-1 | Type-2 |
|---|---|
| *Average error* | |
| 0.041501285 | 0.002356073 |

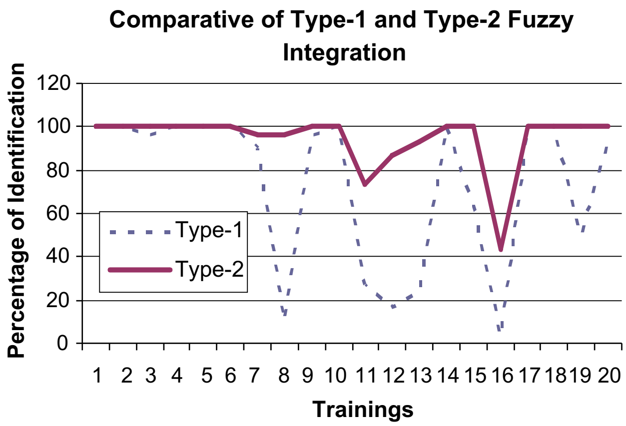


**Fig. 37.** Comparison of Integration with type-1 and type-2 fuzzy systems.

*7.2.1.2. Trapezoidal type-2 fuzzy system.* In Figs. 29–32, it is possible to appreciate the values that the parameters from each trapezoidal membership function for each variable, as well as of inputs as the output, for the type-2 fuzzy system. In this case the GA was used with a population of 80 individuals, a maximum of generations of 100, mutation of 0.0001, crossover of 0.3 of multipoint type; which lasted of 13 min and it was stopped at generation 10, the obtained error was 0.0247836.

**Table 15**
Comparison of integration with type-1 and type-2 fuzzy systems

| # of training | Architecture of the MNN (by layer neurons for each submodule) | Average % for Type-1 | Average % for Type-2 |
|---|---|---|---|
| *Percentages of identification obtained by test Type-1 and Type-2 fuzzy systems* | | | |
| 1 | ModuleFace: 65,50; 75,73; 80,85<br>ModuleFingerprint: 110,90; 115,95; 120,100<br>ModuleVoice: 30,35; 35,43; 40,32 | 100% | 100% |
| 2 | ModuleFace: 112,98; 112,98; 112,98<br>ModuleFingerprint: 121,93; 121,93; 121,93<br>ModuleVoice: 52,36; 52,36; 52,36 | 100% | 100% |
| 3 | ModuleFace: 96,78; 96,78; 96,78<br>ModuleFingerprint: 113,95; 113,95; 113,95<br>ModuleVoice: 56,44; 56,44; 56,44 | 96.67% | 100% |
| 4 | ModuleFace: 102,88; 102,88; 102,88<br>ModuleFingerprint: 111,83; 111,83; 111,83<br>ModuleVoice: 62,46; 62,46; 62,46 | 100% | 100% |
| 5 | ModuleFace: 99,87; 99,87; 99,87<br>ModuleFingerprint: 118,96; 118,96; 118,96<br>ModuleVoice: 78,53; 78,53; 78,53 | 100% | 100% |
| 6 | ModuleFace: 85,70; 85,70; 85,70<br>ModuleFingerprint: 90,80; 90,80; 90,80<br>ModuleVoice: 60,70; 60,70; 60,70 | 100% | 100% |
| 7 | ModuleFace: 60,40; 95,80; 90,75<br>ModuleFingerprint: 150,100; 150,100; 150,100<br>ModuleVoice: 50,25; 50,25; 50,25 | 90% | 96.67% |
| 8 | ModuleFace: 64,32; 64,32; 64,32<br>ModuleFingerprint: 120,60; 120,60; 120,60<br>ModuleVoice: 80,40; 80,40; 80,40 | 10% | 96.67% |
| 9 | ModuleFace: 100,80; 100,80; 100,80<br>ModuleFingerprint: 100,100; 100,100; 100,100<br>ModuleVoice: 65, 50; 65, 50; 65, 50 | 96.67% | 100% |
| 10 | ModuleFace: 50,70; 50,70; 50,70<br>ModuleFingerprint: 90,70; 90,70; 90,70<br>ModuleVoice: 75,35; 75,35; 75,35 | 100% | 100% |
| 11 | ModuleFace: 100,90; 100,90; 100,90<br>ModuleFingerprint: 50,25; 50,25; 50,25<br>ModuleVoice: 60,60; 60,60; 60,60 | 26.67% | 73.33% |
| 12 | ModuleFace: 110,80; 110,80; 110,80<br>ModuleFingerprint: 40,60; 40,60; 40,60<br>ModuleVoice: 80,70; 80,70; 80,70 | 16.67% | 86.67% |
| 13 | ModuleFace: 120,85; 120,85; 120,85<br>ModuleFingerprint: 20,30; 20,30; 20,30<br>ModuleVoice: 60,50; 60,50; 60,50 | 23.33% | 93.33% |
| 14 | ModuleFace: 130,90; 130,90; 130,90<br>ModuleFingerprint: 50,15; 60,20; 65,25<br>ModuleVoice: 68,32; 68,33; 68,34 | 100% | 100% |
| 15 | ModuleFace: 125,90; 125,90; 125,90<br>ModuleFingerprint: 100,150; 100,150; 100,150<br>ModuleVoice: 70,65; 70,65; 70,65 | 63.33% | 100% |
| 16 | ModuleFace: 120,85; 110,80; 125,90<br>ModuleFingerprint: 70,40; 60,35; 65,45<br>ModuleVoice: 70,30; 65,45; 75,35 | 3.33% | 43.33% |
| 17 | ModuleFace: 135,100; 135,100; 135,100<br>ModuleFingerprint: 200,100; 200,100; 200,100<br>ModuleVoice: 80,50; 80,50; 80,50 | 100% | 100% |
| 18 | ModuleFace: 150,100; 150,100; 150,100<br>ModuleFingerprint: 200,150; 200,150; 200,150<br>ModuleVoice: 80,50; 80,50; 80,50 | 100% | 100% |
| 19 | ModuleFace: 128,72; 128,85; 128,92<br>ModuleFingerprint: 95,100; 100,85; 90,78<br>ModuleVoice: 55,43; 55,43; 55,43 | 50% | 100% |
| 20 | ModuleFace: 165,100; 165,100; 165,100<br>ModuleFingerprint: 10,15; 30,50; 50,85<br>ModuleVoice: 75,30; 75,30; 75,30 | 93.33% | 100% |

*7.2.1.3. Gaussian type-2 fuzzy system.* In Figs. 33–36, it is possible to appreciate the values that the parameters from each Gaussian membership function for each variable, as well as inputs as the output, for the type-2 fuzzy system. In this case the GA was used with a population of 55 individuals, a maximum of generations of 100, mutation of 0.001, crossover of 0.6 of simple point type; which lasted of 50 s and it was stopped at generation 4, the obtained error was 0.0000103302.

We have to mention that the previously shown fuzzy systems are the ones with the smallest errors. In Tables 5–7 we show all the obtained results.

### 7.3. Results of the type-1 integration

See Tables 8 and 9.

### 7.4. Results of the type-2 integration

See Tables 10 and 11.

### 7.5. Comparison errors between type-1 and type-2 fuzzy inference systems

See Tables 12–14.

### 7.6. Comparative study of integration with type-1 and type-2 fuzzy inference systems

After the modular neural network trainings were obtained, we make the integration of the modules with the type-1 and type-2 optimized fuzzy systems. Next we show type-1 and type-2 graphics with the 20 modular neural network new trainings and the percentage of the identification (see Fig. 37 and Table 15). We can appreciate that type-2 is better.

## 8. Conclusions

In this paper a study of fuzzy integration methods for MNN's is presented. Type-1 and type-2 fuzzy system are considered as integration methods for a MNN's in biometry applications. The fuzzy systems were optimized using GA to be able to make an accurate comparison of type-1 and type-2 fuzzy logic as methods of integration. A comparison with simulation results for pattern recognition was made.

## References

[1] J.-S.R. Jang, C.-T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing, A Computational Approach to Learning and Machine Intelligence, Prentice Hall, 1997.
[2] Tutorial Type-2 fuzzy logic: theory and applications, Juan Ramón Castro Rodríguez, Universidad Autónoma de Baja California, Instituto Tecnològico de Tijuana, October 9, 2006. <www.hafsamx.org/cis-chmexico/seminar06/tutorial.pdf>.
[3] K.F. Man, K.S. Tang, S. Kwong, Genetic Algorithms, Concepts and Designs, Springer, 1999.
[4] <http://sci2s.ugr.es/publications/ficheros/estylf-2004-1-8.pdf> February, 2007.
[5] Tecnologías Biométricas Aplicadas a la Seguridad, Juan A. Sigüenza, Merino Tapiador Mateos (Ra-ma), March 2007. <http://www.agapea.com/Tecnologias-biometricas-aplicadas-a-la-seguridad–n214440i.htm>.
[6] Soluciones Biométricas Grupo INSYS, Torreón Coah. MEXICO, February 2007. <http://www.insys.com.mx/biometria/biometria.htm>.
[7] Sistemas Modulares, Mezcla de Expertos y Sistemas Híbridos Quiliano Isaac Moro Sancho, Universidad de Valladolid. Spain, Luis Alonso Romero, Universidad de Salamanca, Spain, February 2007. <http://lisisu02.usal.es/~airene/capit7.pdf>.
[8] Aplicaciones e Implementaciones de las Redes Neuronales en reconocimiento de Patrones (AIRENE) 30/06/99 cyted.html Luis Alonso Romero, Universidad de Salamanca, Spain, March 2007. <http://lisisu02.usal.es/~airene/airene.html> lalonso@tejo.usal.es.
[9] L.A. Zadeh, Knowledge representation in fuzzy logic, IEEE Transactions on Knowledge Data Engineering 1 (1989) 89.
[10] L.A. Zadeh, Fuzzy Sets, Information and Control 8 (1975).
[11] G. Chen, T.T. Phan, Introduction to Fuzzy Sets, Fuzzy Logic and Fuzzy Control Systems, CRC Press, EEUU, 2000.
[12] L.A. Zadeh, Fuzzy logic = computing with words, IEEE Transactions on Fuzzy Systems 4 (2) (1996) 103.
[13] Jerry M. Mendel, Uncertain Rule-Based Fuzzy Logic Systems, Introduction and New Directions, Prentice Hall, 2001.
[14] Jerry M. Mendel, Why We Need Type-2 Fuzzy Logic Systems? Article is provided courtesy of Prentice Hall, By Jerry Mendel, May 11, 2001. <http://www.informit.com/articles/article.asp?p=21312&rl=1>.
[15] Jerry M. Mendel, Uncertainty: General Discussions, Article is provided courtesy of Prentice Hall, By Jerry Mendel, May 11, 2001. <http://www.informit.com/articles/article.asp?p=21313>.
[16] Jerry M. Mendel, Robert I. Bob John, Type-2 Fuzzy Sets Made Simple, IEEE Transactions on Fuzzy Systems 10 (2) (2002) 117.
[17] Nilesh N. Karnik, Jerry M. Mendel, Operations on type-2 fuzzy sets, Signal and Image Processing Institute, Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, CA, USA, 11 May 2000.
[18] L.A. Zadeh, Fuzzy logic, Computer 1 (4) (1998) 83–93.
[19] Nilesh N. Karnik, Jerry M. Mendel, IEEE, y Qilian Liang, Type-2 fuzzy logic systems, IEEE Transactions on Fuzzy Systems 7 (6) (1999).
[20] Analysis and Design of Intelligent Systems Using Soft Computing Techniques, Advances in Soft Computing, vol. 41, Patricia Melin, Oscar Castillo, Eduardo Gómez Ramírez, Janusz Kacprzyk, Witold Pedrycz, Springer, 2007.
[21] The 2007 International Joint Conference on Neural Networks, IJCNN 2007 Conference Proceedings, Renaissance Orlando Resort. Orlando, Florida, USA. August 12–17, 2007, IEEE Catalog Number: 07CH37922C ISBN: 1-4244-1380-X, ISSN: 1098-7576, ©2007 IEEE.
[22] M.I. Jordan, R.A. Jacobs, Hierarchical mixtures of experts and the EM algorithm, Neural Computation 6 (1994) 181–214.