Hybrid Intelligent Systems for Pattern Recognition using Soft Computing Techniques

Patricia Melin, Oscar Castillo

Dept. of Computer Science

Tijuana Institute of Technology

Mexico

Emails: pmelin@tectijuana.mx

ocastillo@tectijuana.mx

Outline

- Summary
- Modular Neural Networks (MNN)
- Fuzzy Logic for Response Integration in MNN
- Face and Fingerprint Recognition with MNN
- Voice Recognition with MNN and HGA
- Time Series Prediction using MNN
- Conclusions

Summary

- We describe in this tutorial a new approach for pattern recognition and time series prediction using modular neural networks with a fuzzy logic method for response integration.
- We propose a new architecture for modular neural networks for achieving pattern recognition and time series prediction.
- Also, the method for achieving response integration is based on the fuzzy Sugeno integral with some modifications.

Summary

- Response integration is required to combine the outputs of all the modules in the modular network.
- We have applied the new approach for fingerprint, face, and voice recognition with real databases.
- We have also applied the approach for time series prediction with real data of agricultural products of Mexico and USA.

Summary

- Response integration methods for modular neural networks that have been studied, to the moment, do not solve well real recognition problems with large sets of data or in other cases reduce the final output to the result of only one module.
- Also, in the particular case of face recognition, methods of weighted statistical average do not work well due to the nature of the face recognition problem.
- For these reasons, a new approach for pattern recognition using modular neural networks and fuzzy integration of responses was proposed in this work.

Modular Neural Networks

- There exists a lot of neural network architectures in the literature that work well when the number of inputs is relatively small, but when the complexity of the problem grows or the number of inputs increases, their performance decreases very quickly.
- For this reason, there has also been research work in compensating in some way the problems in learning of a single neural network over high dimensional spaces.
- In some research work has been shown that the use of multiple neural systems have better performance or even solve problems that monolithic neural networks are not able to solve, in the case of multiple networks we can have the ensemble and modular types.

Modular Neural Networks cont.

- The term “ensemble” is used when a redundant set of neural networks is utilized.
- In this case, each of the neural networks is redundant because it is providing a solution for the same task, as it is shown in Figure 1.

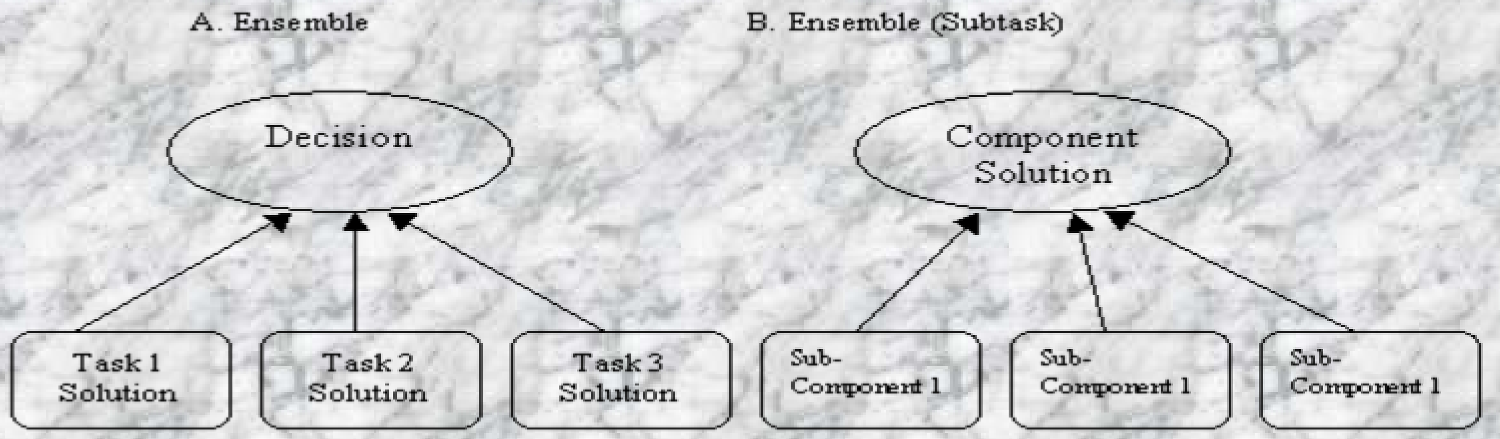


Fig. 1 . Ensembles for one task and subtask.

Modular Neural Networks cont.

- On the other hand, in the modular approach, one task or problem is decomposed into subtasks, and the complete solution requires the contribution of all the modules, as it is shown in Figure 2.

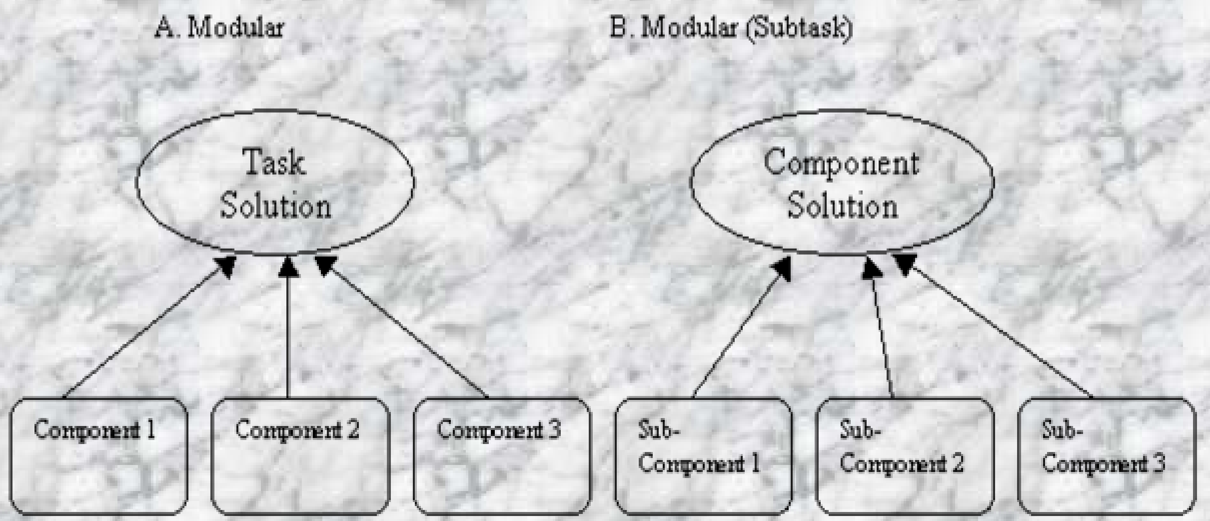


Fig. 2. Modular approach for task and subtask

Modular Neural Networks cont.

- ***Multiple Neural Networks***

- In this approach we can find networks that use strongly separated architectures.
- Each neural network works independently in its own domain.
- Each of the neural networks is build and trained for a specific task.
- The final decision is based on the results of the individual networks, called agents or experts.

Modular Neural Networks cont.

- One example of this decision is shown in Figure 3, where a multiple architecture is used, one module consists of a neural network trained for recognizing a person by the voice, while the other module is a neural network trained for recognizing a person by the image.

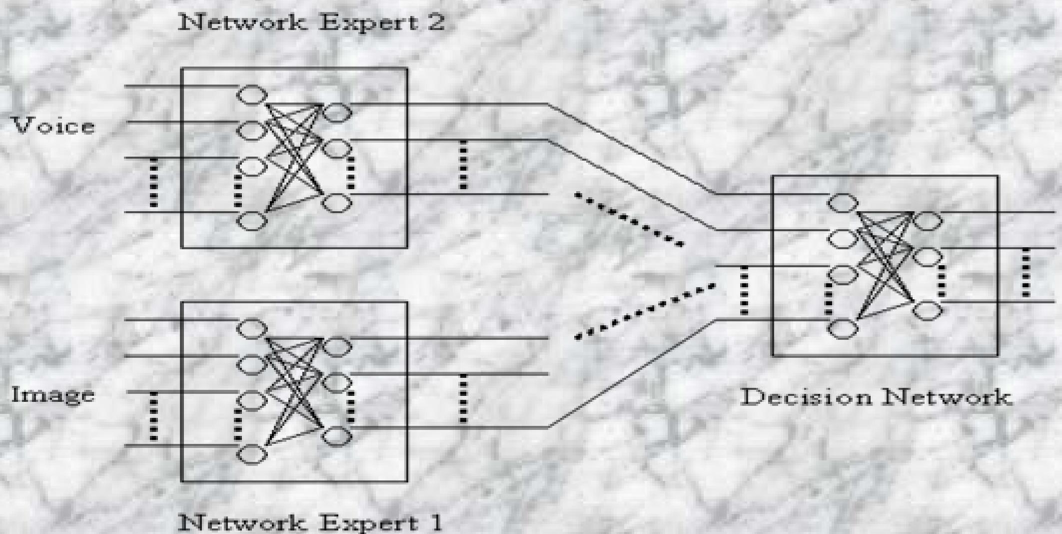


Fig. 3 Multiple networks for voice and image.

Modular Neural Networks cont.

Main Architectures with Multiple Networks

- Mixture of Experts (ME): The ME can be viewed as a modular version of the multi-layer networks with supervised training or the associative version of competitive learning. In this design, the local experts are trained with the data sets to mitigate weight interference from one expert to the other.
- Gate of Experts: In this case, an optimization algorithm is used for the gating network, to combine the outputs from the experts.
- Hierarchical Mixture of Experts: In this architecture, the individual outputs from the experts are combined with several gating networks in a hierarchical way.

Modular Neural Networks cont.

- When considering modular networks to solve a problem, one has to take into account the following points:

1) Decompose the main problem into subtasks.
2) Organizing the modular architecture, taking into account the nature of each subtask.
3) Communication between modules is important, not only in the input of the system but also in the response integration.

We will concentrate in these three points, and in particular the third point, the communication between modules, more specifically information fusion at the integrating module to generate the output of the complete modular system.

Advantages of Modular Neural Networks

- They give a significant improvement in the learning capabilities, over monolithic neural networks, due to the constraints imposed on the modular topology
- They allow complex behavior modeling, by using different types of knowledge, which is not possible without using modularity
- Modularity may imply reduction of number of parameters, which will allow and increase in computing speed and better generalization capabilities
- They avoid the interference that affects “global” neural networks, they help determine the activity that is being done in each part of the system, helping to understand the role that each network plays within the complete system

Methods for Response Integration

- The importance of this part of the architecture for pattern recognition is due to the high dimensionality of this type of problems. As a consequence in pattern recognition is good alternative to consider a modular approach.
- This has the advantage of reducing the time required of learning and it also increases accuracy.
- In our case, we consider dividing the images or time series into several different regions.

Now the question is How to integrate the different outputs given by the different modules of the system to generate the final output of the complete system

Fuzzy Integral and Sugeno Measures

- Fuzzy integrals can be viewed as non-linear functions defined with respect to fuzzy measures.
- In particular, the "gλ-fuzzy measure" introduced by Sugeno [9] can be used to define fuzzy integrals.
- The ability of fuzzy integrals to combine the results of multiple information sources has been mentioned in previous works.
- <u>Definition 1.</u> A function of sets $g:2^x \rightarrow (0, 1)$ is called a fuzzy measure if:
- 1) $g(0)=0 \quad g(x)=1$

  2) $g(A) \leq g(B)$ if $A \subset B$

  3) if $\{Ai\}i\alpha$ =1 is a sequence of increments of the measurable set then (1)

  $$\lim_{i \rightarrow \infty} g(Ai) = g(\lim_{i \rightarrow \infty} Ai)$$

Fuzzy Integral and Sugeno Measures cont.

- From the general definition of the fuzzy measure, Sugeno introduced what is called "gλ-fuzzy measure", which satisfies the following additive property:

  For every A, B $\subset$ X and A $\cap$ B = θ,

  $$g(A \cup B) = g(A) + g(B) + \lambda g(A)g(B), \qquad (2)$$

  for some value of λ>-1.

- This property says that the measure of the union of two disjunct sets can be obtained directly from the individual measures.

.

Fuzzy Integral and Sugeno Measures cont.

- Using the concept of fuzzy measures, Sugeno [9] developed the concept of fuzzy integrals, which are non-linear functions defined with respect to fuzzy measures like the gλ-fuzzy measure

- One can interpret fuzzy integrals as finding the maximum degree of similarity between the objective and expected value.

Modular Neural Networks with Fuzzy Logic for Response Integration Applied to Face and Fingerprint Recognition

Introduction cont.

- The basic idea of the approach is to divide a human face into three different regions:

  1) the eyes, 2) nose 3) mouth,

- and the fingerprint also into three parts:

  1) top, 2) middle 3) bottom.

- Each of these regions is assigned to one module of the modular neural network.
- In this way, the modular neural network has three different modules, one for each of the regions of the human face and the fingerprint.

Introduction cont.

- At the end, the final decision of face and fingerprint recognition is done by an integration module, which has to take into account the results of each of the modules.
- In our approach, the integration module uses the fuzzy Sugeno integral to combine the outputs of the three modules.
- The fuzzy Sugeno integral allows the integration of responses from the three modules of the eyes, nose and mouth of a human specific face and the integration of the responses from the three modules of the fingerprint parts

Proposed Method for Face Recognition

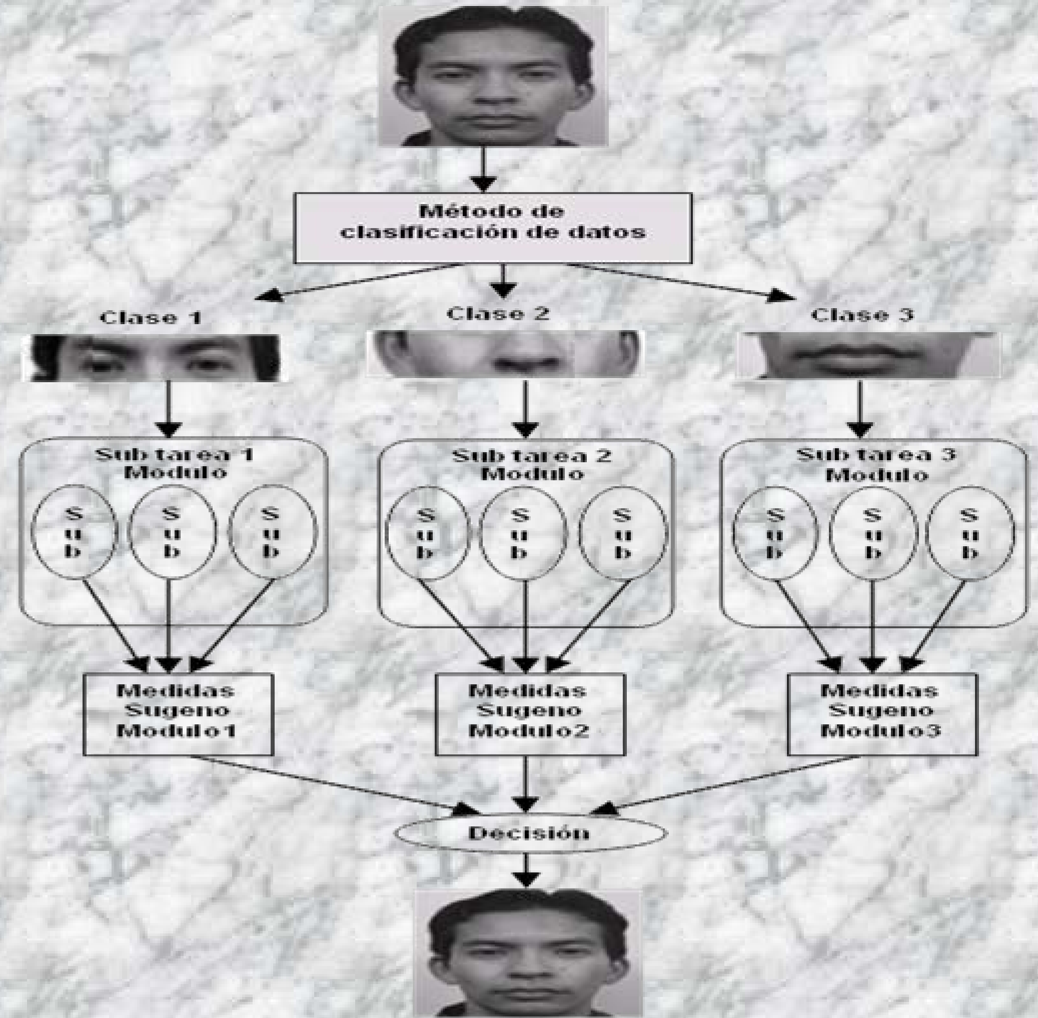
1. A selection of representative faces for each person is obtained.

2. A classification method is applied to the prototypes selected to find the classes.

3. Perform data extraction to the classes to send them to each of the modules of the modular neural network.

4. Perform training with the data of the classes that were found, modifying the different parameters of training in each module.

Neural Network Architecture



Phase of Data Classification

To perform the classification with the methods the image was converted into vector form, see figure 4.

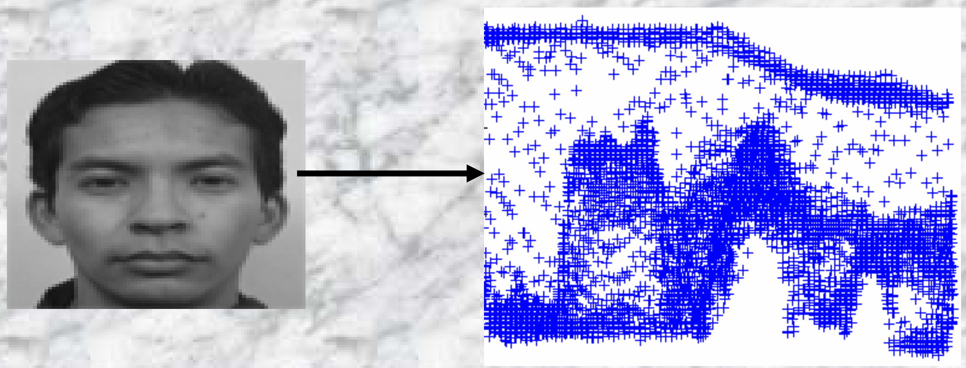
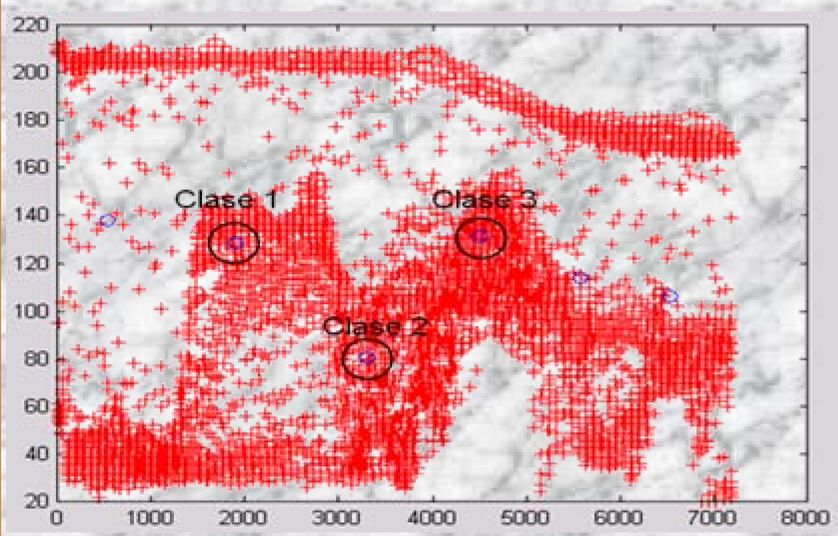
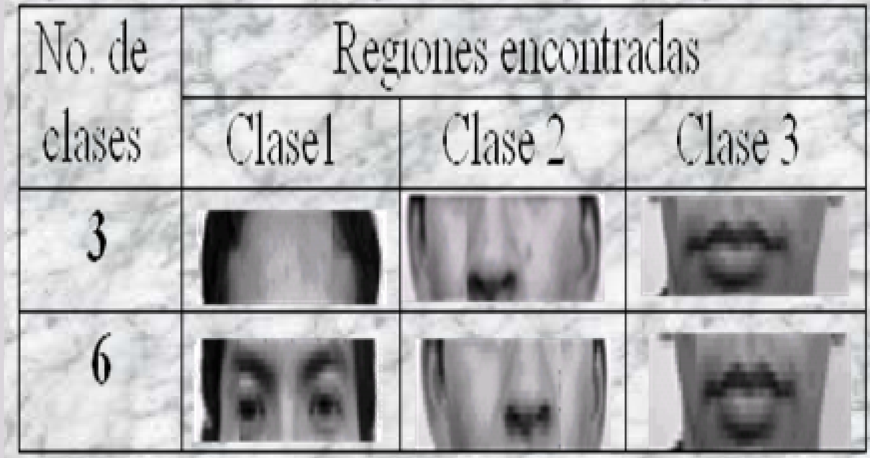


Figure 4. Image converted into vector form.

Method of Competitive Learning

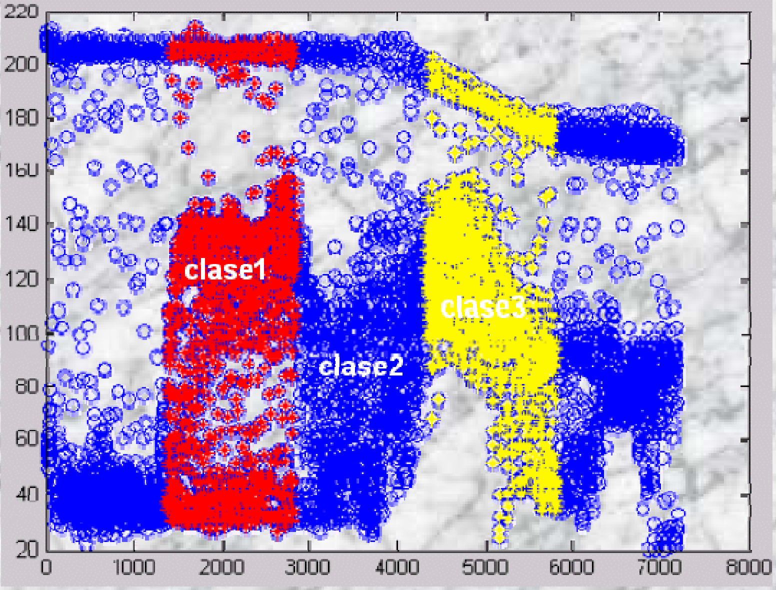
The tests that were done with this method were obtained by varying the number of classes to be considered, these variations were made from three up to six groups.





Fuzzy C-Means Method

Another method of classification that was used is the Fuzzy C-Means clustering algorithm. The results that were obtained with this fuzzy method are shown in the following table .



No. de clases	Regiones encontradas		
	Región 1	Región 2	Región 3
3			
4			
5			
6			

Results from training and recognition

Once the classes are obtained for the different regions of the eyes, nose and mouth, the data is input into each of the modules of the neural network, to perform the training.

Modules	No. layers	Neurons by layer
1st Module	2	250, 250
2nd Module	2	220, 260
3rd Module	2	230, 240

Proposed Architecture and Results

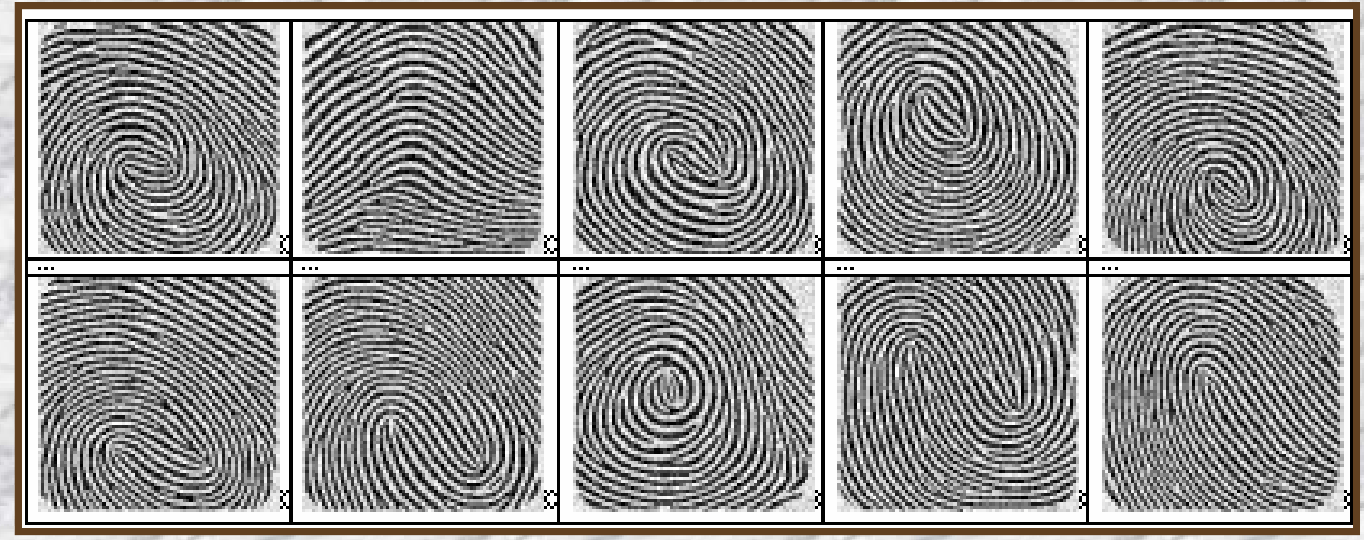
- In the experiments performed in this research work, we used photographs that were taken with a digital camera and fingerprints with fingerscan from students and professors of our Institution.
- The photographs were taken in such a way that they had 148 pixels wide and 90 pixels high, with a resolution of 300x300 ppi, and representation of a gray scale.

Proposed Architecture and Results cont.

- In addition to the training data we did use photographs that were obtained by applying noise in a random fashion to each of the original photos, which was increased from 10 to 80% noise level.

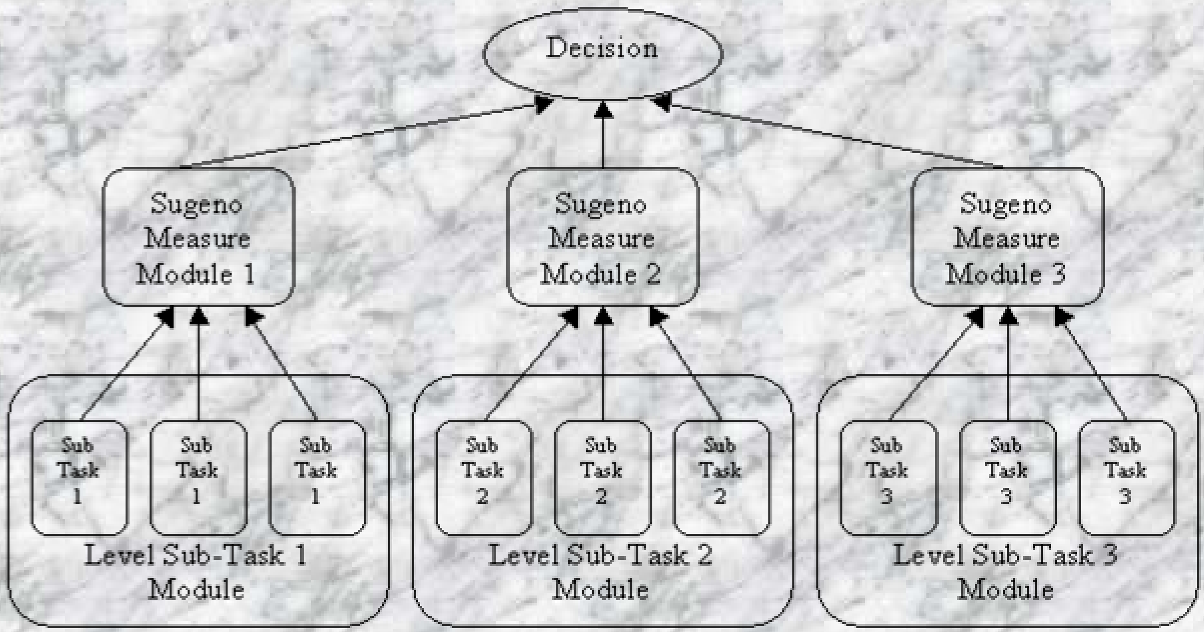
Proposed Architecture and Results cont.

- The images of fingerprints were taken in such a way that they had 198 pixels wide and 200 pixels high, with a resolution of 300x300 ppi, and a representation of a gray scale, some of these images are shown in the next Figure.



Proposed Architecture and Results cont.

- The architecture proposed in this work consist of three main modules, in which each of them in turn consists of a set of neural networks trained with the same data, which provides the modular architecture shown in the next Figure.

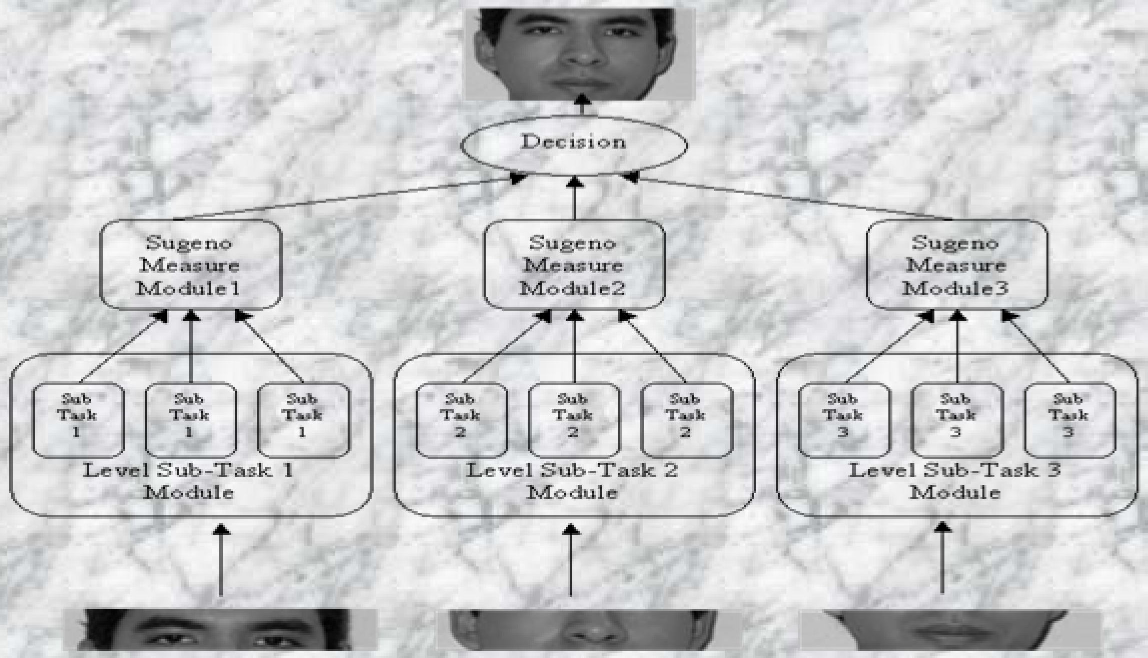


Proposed Architecture and Results cont.

- The input to the modular system is a complete photograph.
- For performing the neural network training, the images of the human faces were divided in three different regions with the method of clasification.
- An example of this image division is shown in Figure

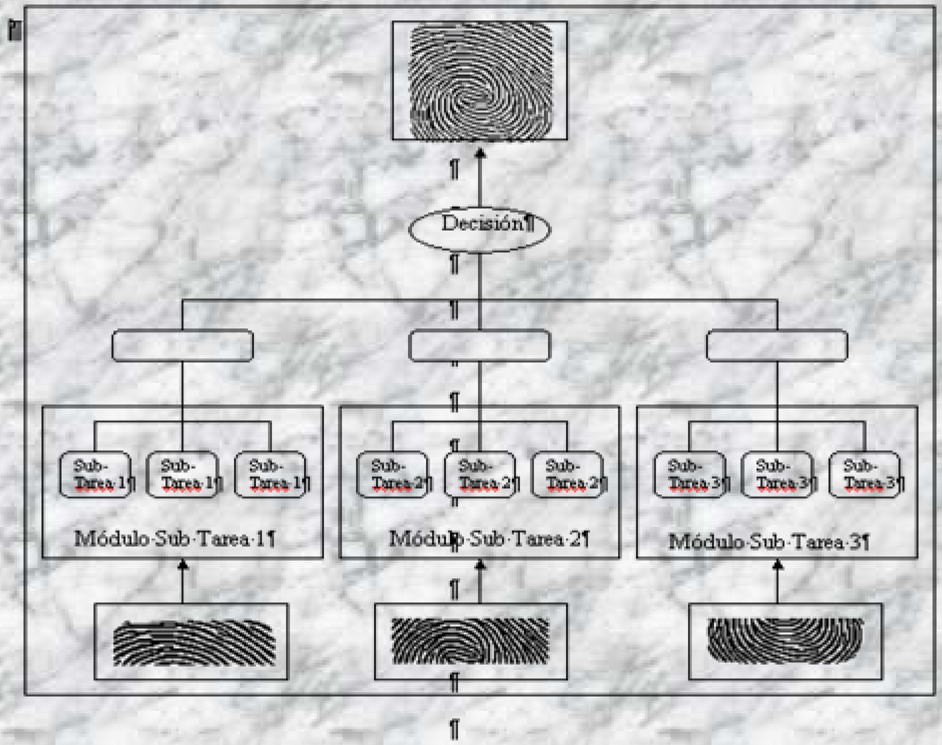
Proposed Architecture and Results cont.

- As output to the system we have an image that corresponds to the complete image that was originally given as input to the modular system, we show in Figure an example of this for face recognition.



Proposed Architecture and Results cont.

- In the same way, the fingerprints are divided in three parts and given to the corresponding Sub task module. This is illustrated in the next Figure.



Proposed Architecture and Results cont.

Description of the Integration Module

- The integration modules performs its task in two phases. In the first phase, it obtains two matrices.
- The first matrix, called h, of dimension 3x3, stores the larger index values resulting from the competition for each of the members of the modules.
- The second matrix , called I, also of dimension 3x3, stores the photograph number corresponding to the particular index.
- Once the first phase is finished, the second phase is initiated, in which the decision is obtained.

Proposed Architecture and Results cont.

- Before making a decision, if there is consensus in the three modules, we can proceed to give the final decision, if there isn't consensus then we have search in matrix g to find the larger index values and then calculate the Sugeno fuzzy measures for each of the modules, using the following formula,

$$g(M_i) = h(A) + h(B) + \lambda\, h(A)\, h(B) \qquad (5)$$

- Where λ is equal to 1 in this particular case. Once we have these measures, we select the largest one to show the corresponding photograph.

Proposed Architecture and Results cont.

Summary of Results

- We describe in this section the experimental results obtained with the proposed approach using the 20 photographs as training data. We show in Table the relation between accuracy (measured as the percentage of correct results) and the percentage of noise in the figures.

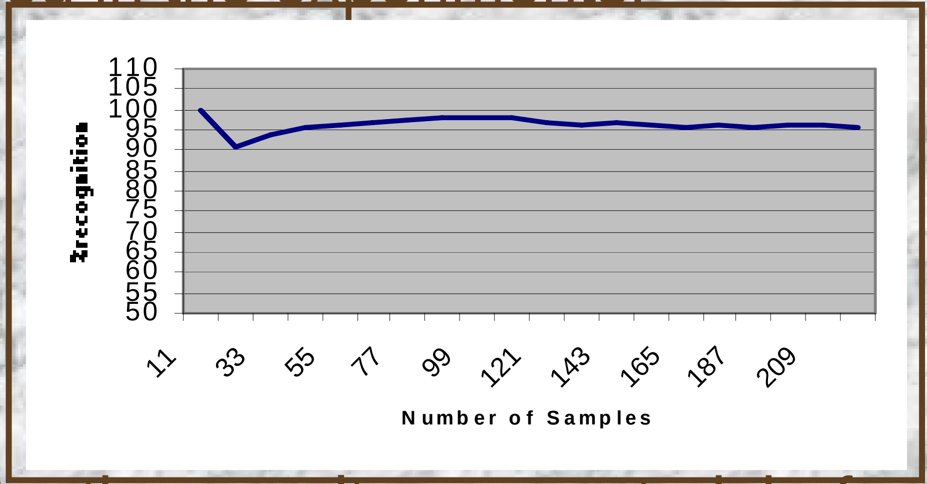
% of noise	% accuracy
0	100
10	100
20	100
30	100
40	95
50	100
60	100
70	95
80	100

Proposed Architecture and Results cont.

- The % of noise was added in a random fashion to the testing data set, that consisted of the original photographs, plus 200 additional images. We show in the following Figure some sample images with noise.

Proposed Architecture and Results cont.

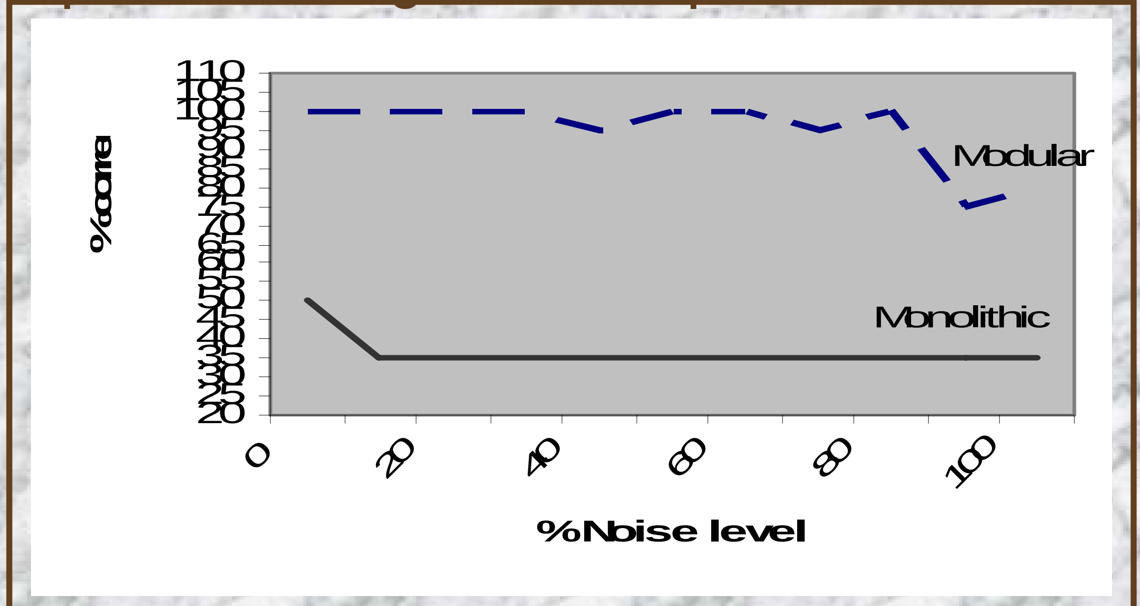
- We show in the following Figure a plot relating the percentage of recognition against the number of examples used in the experiments.



- In addition to the results presented before, we also compared the performance of the modular approach, against the performance of a monolithic neural network approach.

Proposed Architecture and Results cont.

- The conclusion of this comparison was that for this type of input data, the monolithic approach is not feasible, since not only training time is larger, also the recognition is too small for real-world use. We show in Figure a plot showing this comparison.



Summary of results

- We showed in this work the experimental results obtained with the proposed hybrid modular approach.
- In fact, we did achieve a 98.9% recognition rate on the testing data, even with an 80% level of applied noise.
- We also have to notice that it was achieved a 96.7 % of average reliability with our modular approach.
- This percentage values was obtained by averaging

Optimization of Modular Neural Networks using Genetic Algorithms Applied to Voice Recognition

Optimization of MNN using HGA

- We describe in this section the evolution of Modular Neural Networks using Hierarchical Genetic Algorithms.
- Modular Neural Networks(MNN) have shown significant learning improvement over single Neural Networks(NN).
- For this reason, the use of MNN for Pattern Recognition is well justified.
- However, network topology design of MNN is at least an order of magnitude more difficult than for classical NNs.

Optimization of MNN using HGA

- We describe the use of a Hierarchical Genetic Algorithm(HGA) for optimizing the topology of each of the Neural Network Modules of the MNN.
- The HGA is clearly needed due to the fact that topology optimization requires that we are able to manage both the layer and node information for each of the MNN modules.

Optimization of MNN using HGA

- Usually the MNN implementations are based on the “Divide and Conquer” principle. This principle consist first in breaking down a task into smaller and less complex subtasks, to make learn each task by different experts and then, to reuse the learning of each subtask to solve the whole problem.
- To make an efficient use of MNN, however, we need to optimize their topology for a specific problem. We propose the use of a HGA for evolving the topology design of the complete modular network, which means optimizing each and all of the modules of the MNN.

Evolving Neural Network Architectures

- Neural Network researchers know all too well that the particular architecture chosen can determine the success or failure of the application, so they would like very much to be able to automatically optimize the procedure of designing an architecture for a particular application.
- Recently, some efforts have been made to use genetic algorithms to evolve aspects of neural networks.
- Many believe that genetic algorithms are well suited for this task.

Hierarchical Genetic Algorithms for Neural Networks

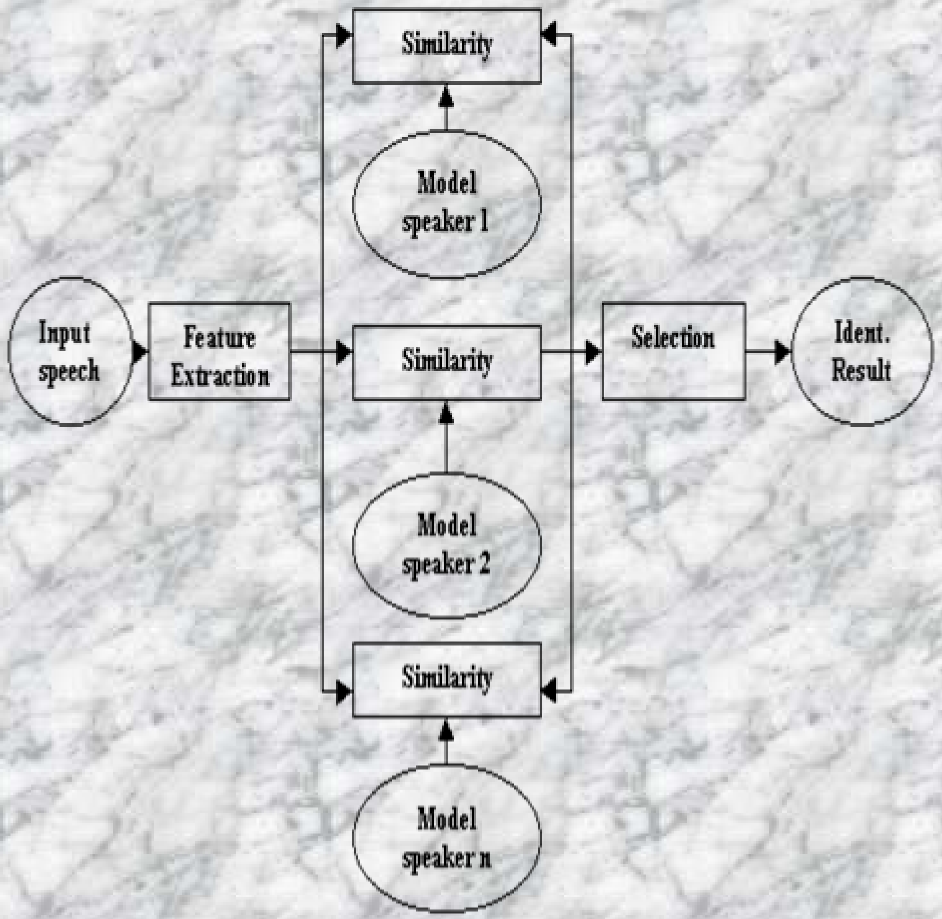
- The HGA differs from the standard GA with a hierarchy structure in that each chromosome consists of multilevel genes. Each chromosome consist of two types of genes: control genes and connection genes.
- The control genes in the form of bits, are the genes for layers and neurons for activation.
- The connection genes, a real value representation, are the genes for connection weightings and neuron bias.

The Problem of Speech Recognition

- Speaker recognition, which can be classified into identification and verification, is the process of automatically recognizing who is speaking on the basis of individual information included in speech waves.
- This technique makes it possible to use the speaker's voice to verify their identity and control access to services such as voice dialing, banking by telephone, information services, security control for confidential information areas and remote access to computers.

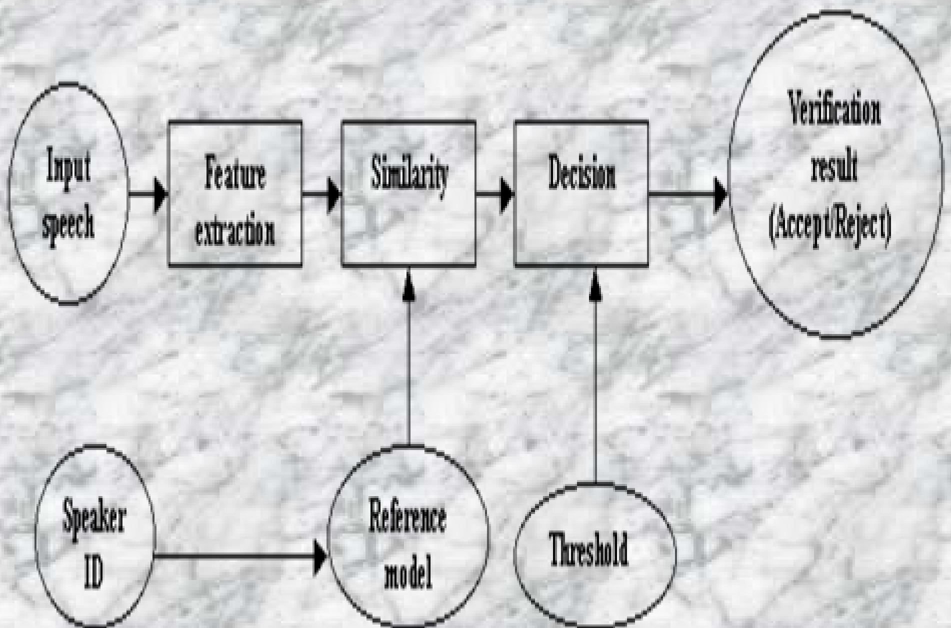
The Problem of Speech Recognition

- Speaker identification is the process of determining which registered speaker provides a given utterance.

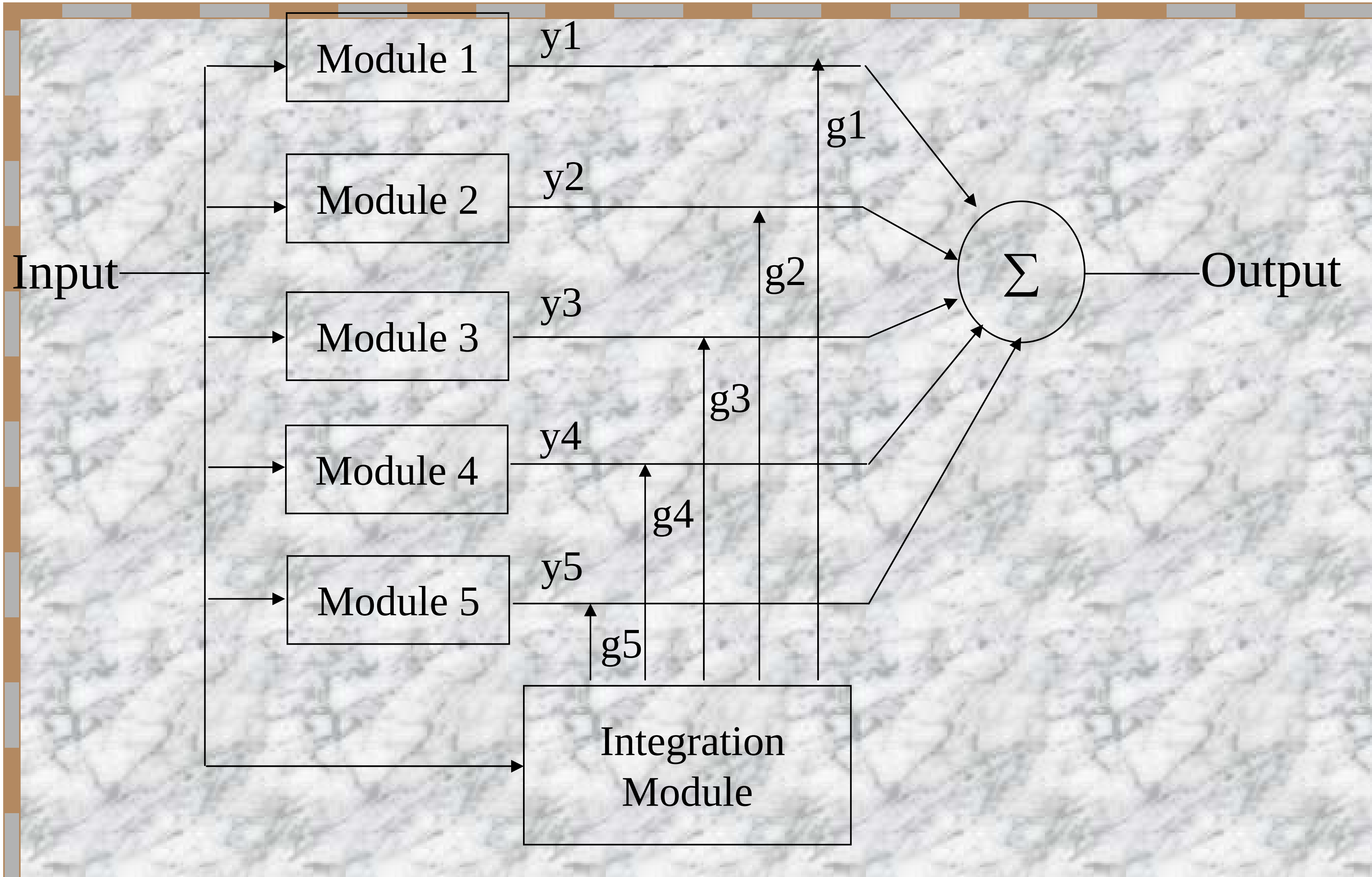


The Problem of Speech Recognition

- Speaker verification, on the other hand, is the process of accepting or rejecting the identity claim of a speaker.

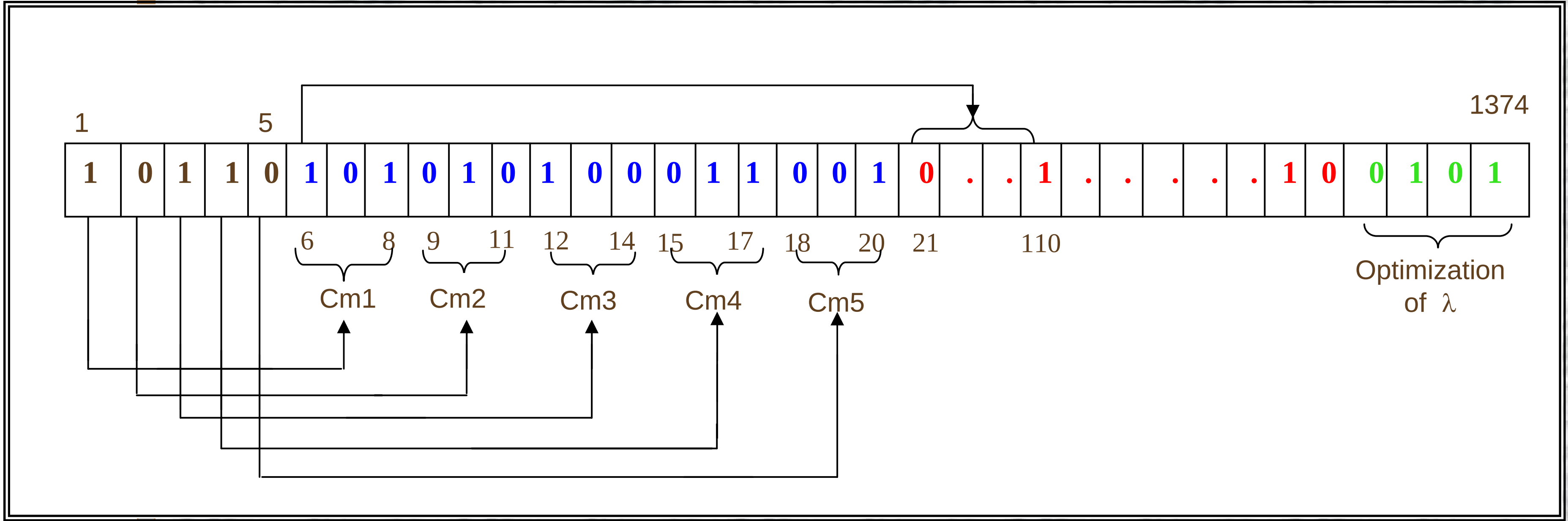


Initial Proposed Architecture



Proposed Structure of the Chromosome

- With respect to the genetic algorithm, a binary representation was used for respresenting the modular neural network



Basic Structure of the Chromosome

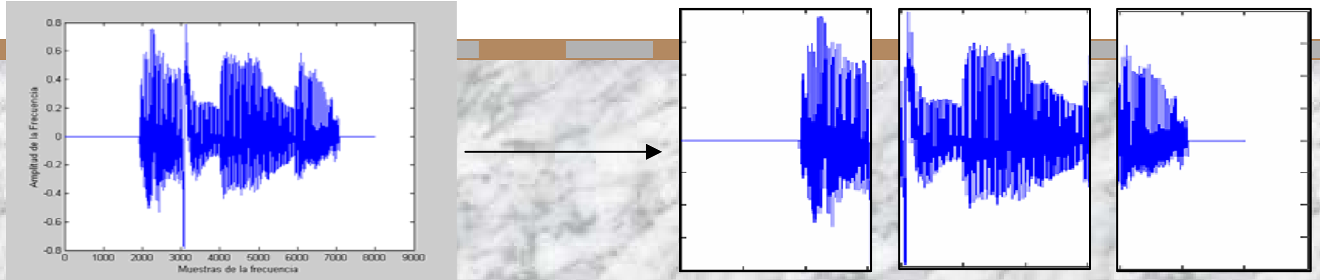
Proposed Architecture

Optimization of parameter λ

Bits	Value
0 0 0 0	-1
0 0 0 1	-0.875
0 0 1 0	-0.75
0 0 1 1	-0.625
0 1 0 0	-0.5
0 1 0 1	-0.375
0 1 1 0	-0.25
0 1 1 1	-0.125

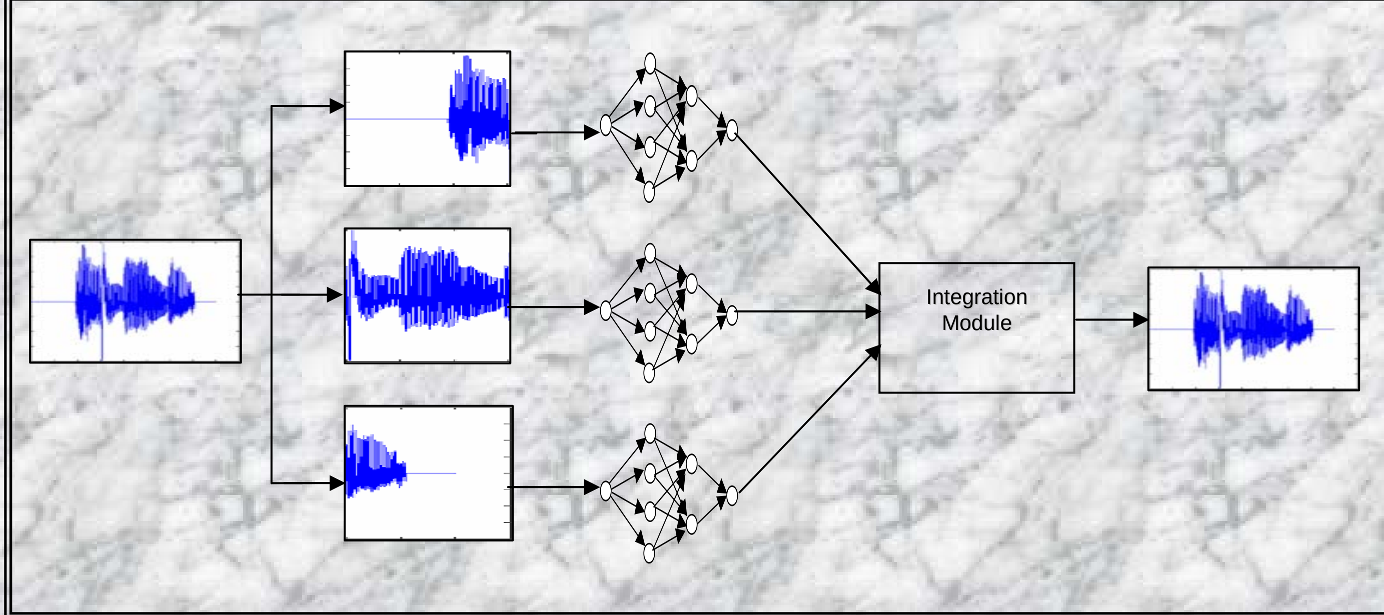
Bits	Value
1 0 0 0	0.125
1 0 0 1	0.25
1 0 1 0	0.375
1 0 1 1	0.5
1 1 0 0	0.625
1 1 0 1	0.75
1 1 1 0	0.875
1 1 1 1	1

Proposed Architecture cont.



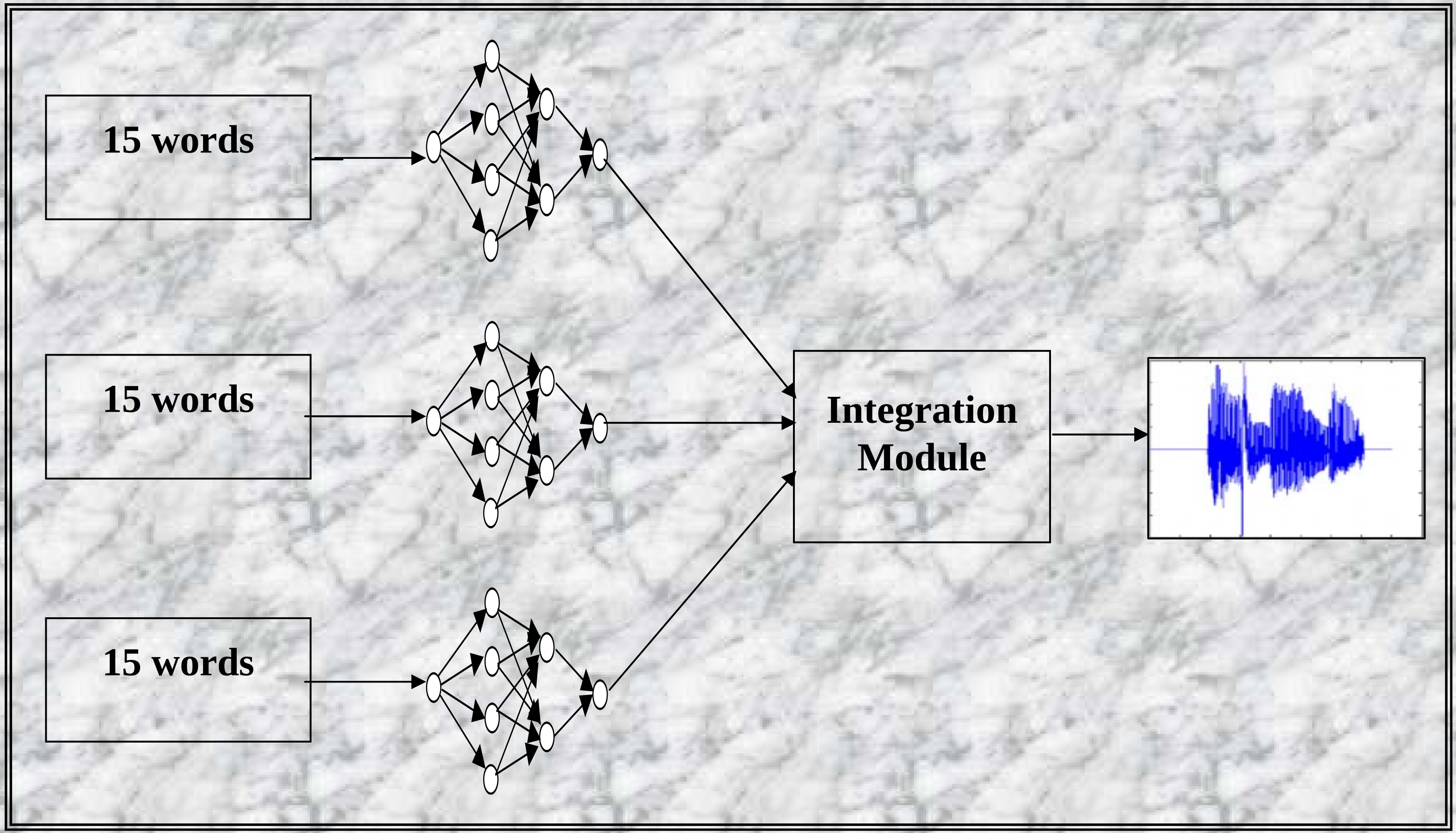
a) Word “camino” b) Division of word “camino”

Division of Word “camino” in three parts



Modular neural network architecture with word division

Proposed Architecture cont.



Modular Neural Network without word division

Results

Computer Time of HGA	Number of Modules	Proposed Architecture	Recognized Words
29 min	2	**3 layers (37-44-36)** **1 layer (43)**	9
30 min	3	**2 layers (40-48)** **2 layers (40-40)** **1 layer (47)**	9
29 min	3	**2 layers (70-85)** **2 layers (74-68)** **2 layers (72-75)**	10

Value of Parameter α

Number of Modules	Value of Parameter α
1 module	271
2 modules	541
3 modules	811
4 modules	1081
5 modules	1351

Results with monolithic NN

Number of Modules	Layers and Neurons	Training Time
1	(35 - 41)	53 hours
1	(38 – 44)	46 hours
1	(38 – 41)	50 hours
1	(35 – 25)	36 hours
1	(34 – 31)	41 hours
1	(38 – 37)	39 hours

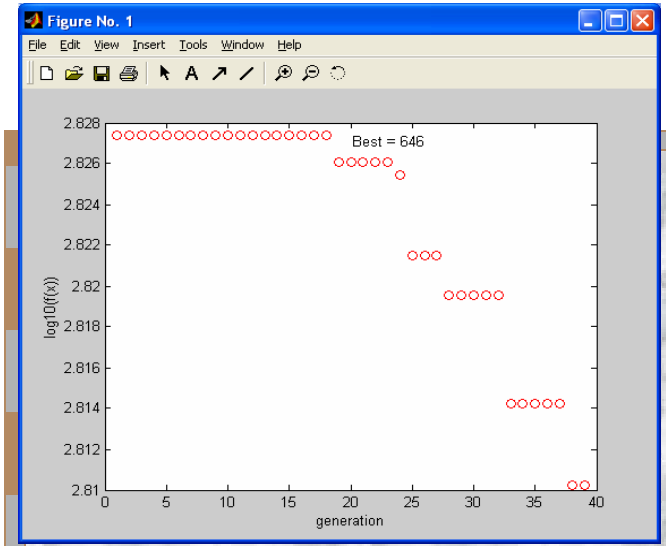
Results with MNN

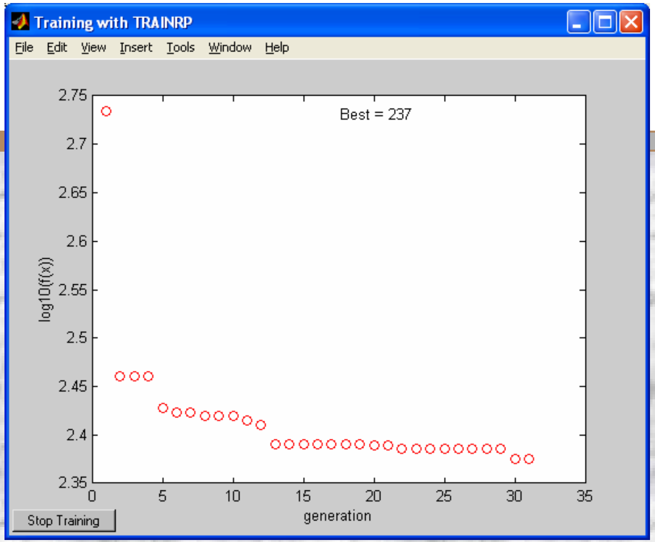
Number of Test	Number of Modules	Architecture	Parameter λ	Recognized Words	
				With Word Division	Without Word Div.
1	3	(40-48) (40-40) (47)	0.5	22	22
2	3	(46-43) (44-42) (43-47)	1	21	22
3	3	(70-45-55) (50-45-65) (50-45-75)	0.125	33	28
4	3	(40-46-50) (36-44-43) (48-41)	0.875	15	16
5	3	(49) (42-45) (45)	1	11	12

Results with MNN cont.

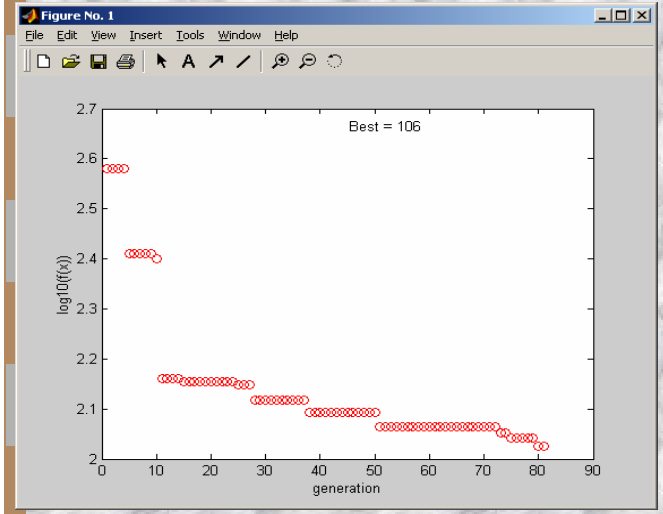
Number of Test	Number of Modules	Architecture	Parameter λ	Recognized words	
				With Word Division	Without Word Div.
6	3	(49-43-37) (37-55-45) (42-42)	1	12	22
7	3	(40-43) (38-47) (43)	0.625	20	13
8	3	(44-44-45) (40-39) (47-40)	1	19	28

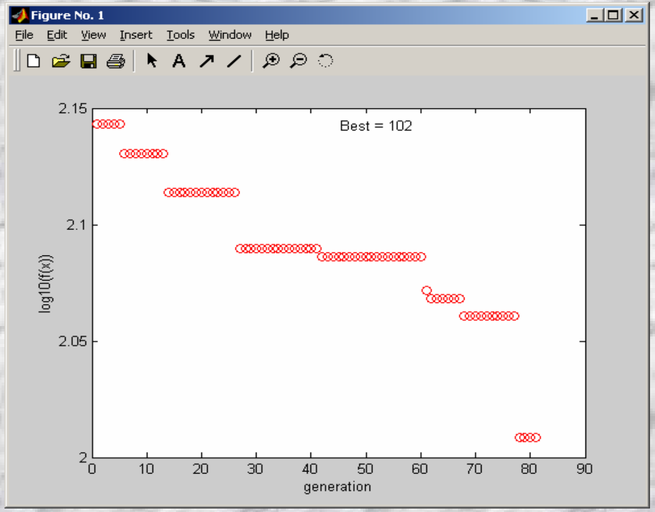
Results with MNN using HGA





Plots showing convergence of HGA with 38 and 30 generations





Plots showing convergence of HGA with 80 generations

Sample Words

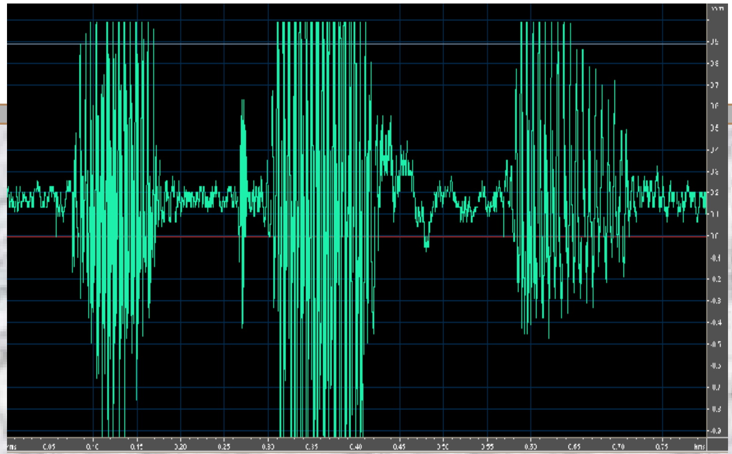


Figure 1. Input signal of word "Acosta" with noise

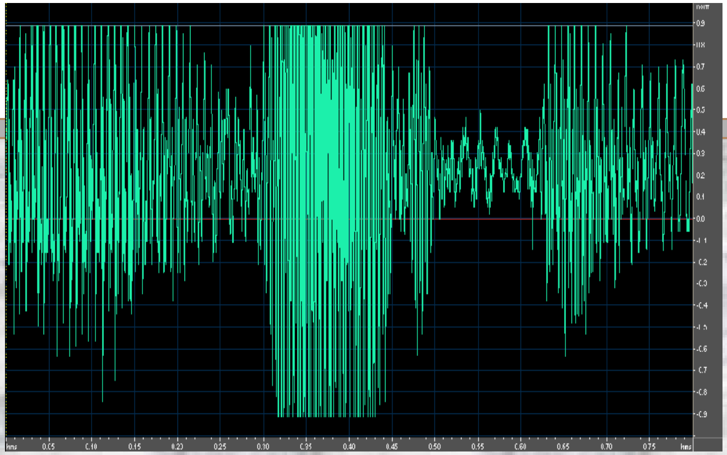


Figure 3. Input signal of word "Alberto" with noise.

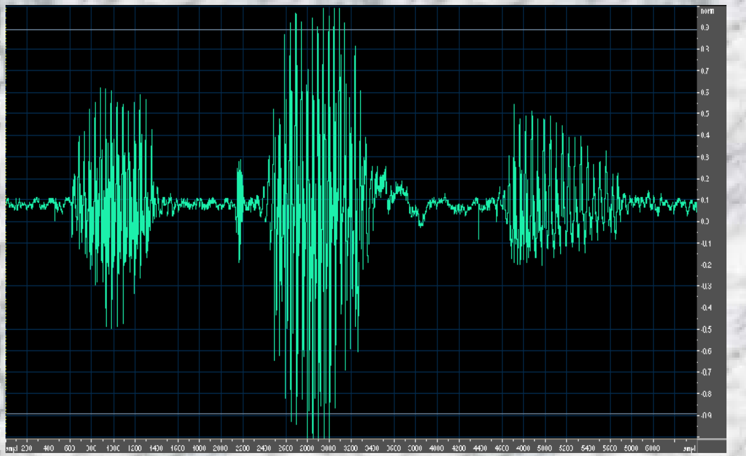


Figure 2. Identification of word "Acosta".

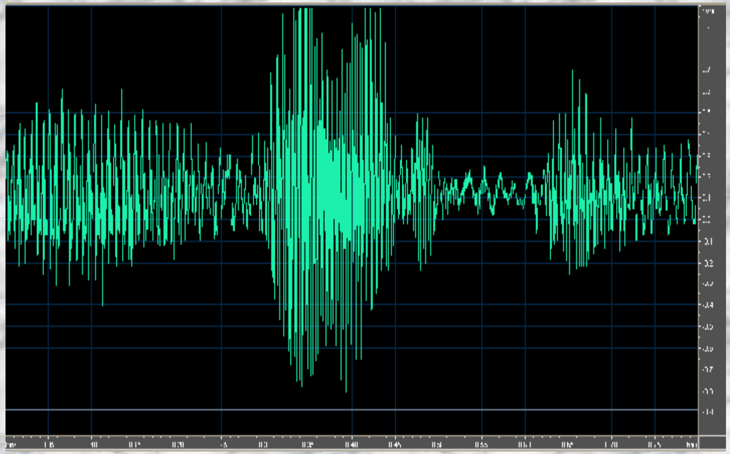


Figure 4. Identification of word "Alberto".

Summary of results

- We described in this work our HGA approach for MNN topology design and optimization.
- The proposed approach was illustrated with a specific problem of Pattern Recognition.
- The best MNN is obtained by evolving the modules according to the error of identification and also the complexity of the modules.

Human Recognition

- Our proposed approach for human recognition consists in integrating the information of the three main biometric parts of the person: the voice, the face, and the fingerprint.
- Basically, we have an independent system for recognizing a person from each of its biometric information (voice, face, and fingerprint), and at the end we have an integration unit to make a final decision based on the results from each of the modules.
- In the next Figure we show the general hybrid architecture of our approach in which it is clearly seen that we have one module for voice, one module for face recognition, and one module for fingerprint recognition. At the top, we have the decision unit integrating the results from the three modules.

Final Architecture

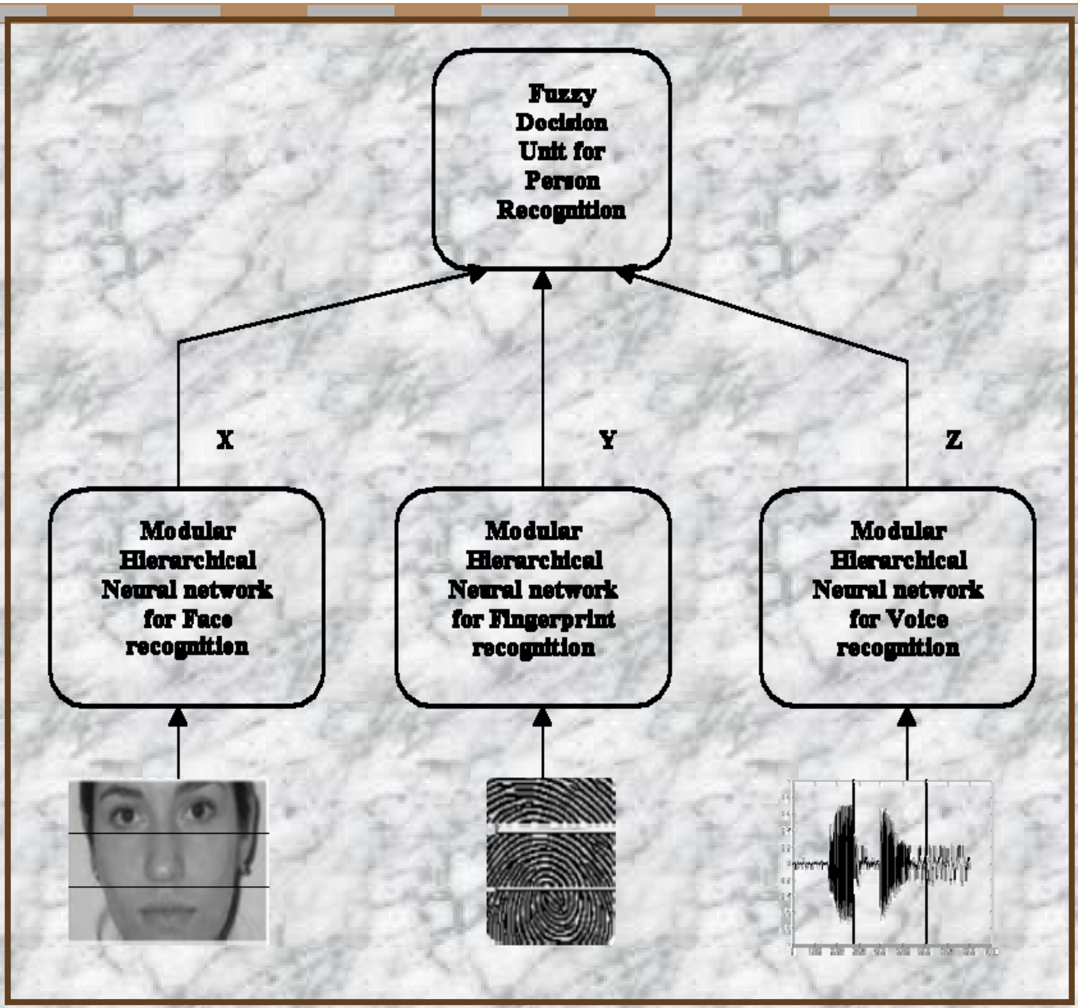


Figure: Architecture of the proposed approach.

Modular Neural Networks and Fuzzy Sugeno Integrals for Time Series Prediction

Introduction

- We describe in this part of the presentation the use of several neural network architectures to the problem of simulating and predicting the dynamic behavior of complex economic time series
- We use several neural network models and training algorithms to compare the results and decide at the end, which one is best for this application.
- In this case, we use real time series of prices of consumer goods to test our models.
- Real prices of tomato and green onion in the U.S. show complex fluctuations in time and are very complicated to predict with traditional statistical approaches.

Introduction cont. …

- The financial markets are well known for wide variations in prices over short and long terms.
- These fluctuations are due to a large number of deals produced by agents that act independently from each other.
- However, even in the middle of the apparently chaotic world, there are opportunities for making good predictions.
- Traditionally, brokers have relied on technical analysis, based mainly on looking at trends, moving averages, and certain graphical patterns, for performing predictions and subsequently making deals

Introduction cont. …

- More recently, soft computing methodologies, such as neural networks, fuzzy logic, and genetic algorithms, have been applied to the problem of forecasting complex time series.
- These methods have shown some advantages over the traditional statistical ones. The main advantage of soft computing methodologies is that, we do not need to specify the structure of a model a-priori, which is clearly needed in the classical regression analysis.
- Also, soft computing models are non-linear in nature and they can approximate more easily complex dynamical systems, than simple linear statistical models.

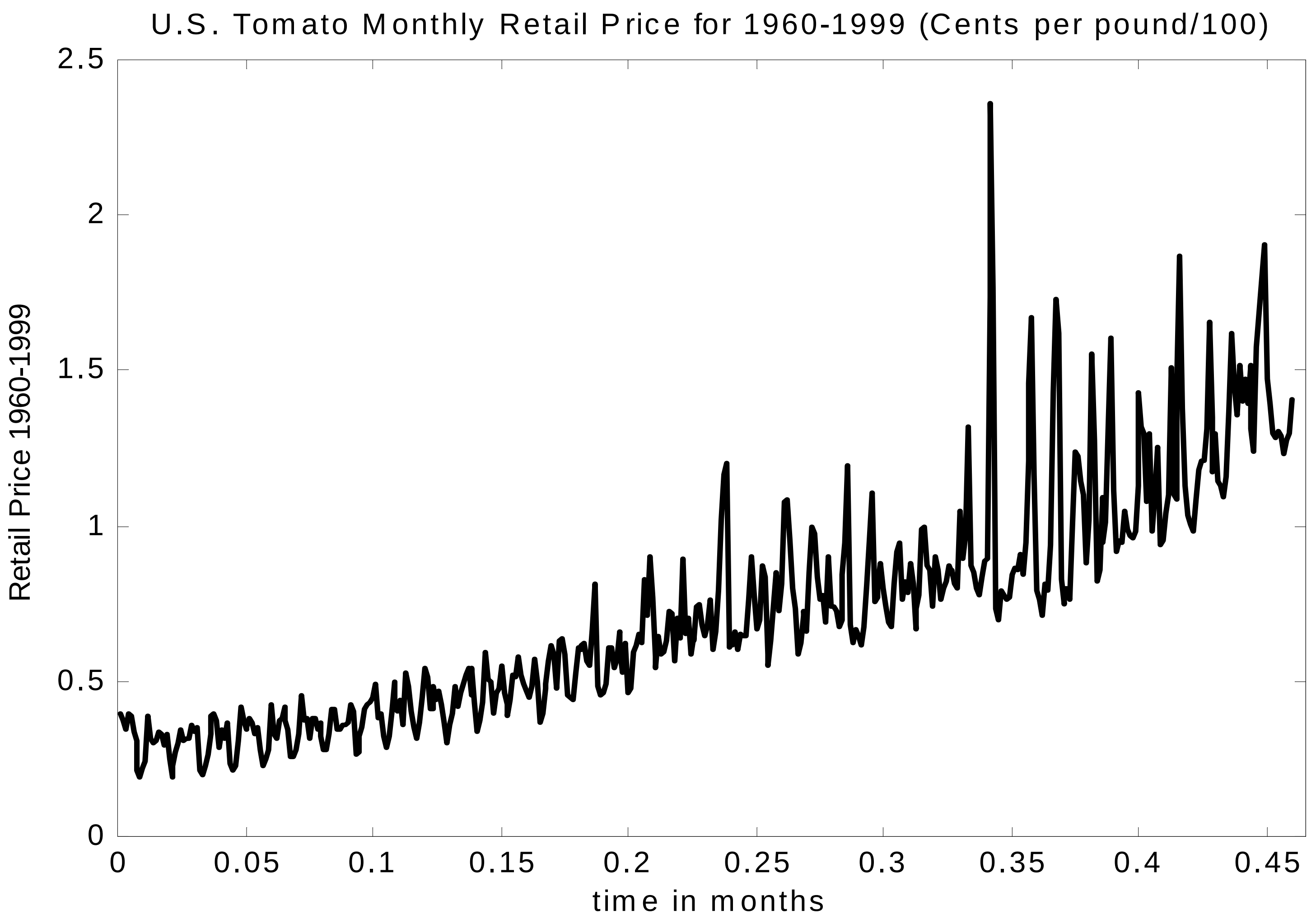
Introduction cont. …

- Of course, there are also disadvantages in using soft computing models instead of statistical ones. In classical regression models, we can use the information given by the parameters to understand the process, i.e. the coefficients of the model can represent the elasticity of price for a certain good in the market.
- However, if the main objective is to forecast as closely as possible the time series, then the use of soft computing methodologies for prediction is clearly justified

Simulation and Forecasting Prices of Consumer Goods in the U.S. Market

- We will consider the problem of forecasting the prices of tomato in the U.S. market. The time series for the prices of this consumer good shows very complicated dynamic behavior, and for this reason it is interesting to analyze and predict the future prices for this good.
- We show in the following Figure the time series of monthly tomato prices in the period of 1960 to 1999, to give an idea of the complex dynamic behavior of this time series
- We will apply both the modular and monolithic neural network approach and also the linear regression method to the problem of forecasting the time series of tomato prices. Then, we will compare the results of these approaches to select the best one for forecasting.

Figure: Prices in US Dollars of tomato from January 1960 to December 1999



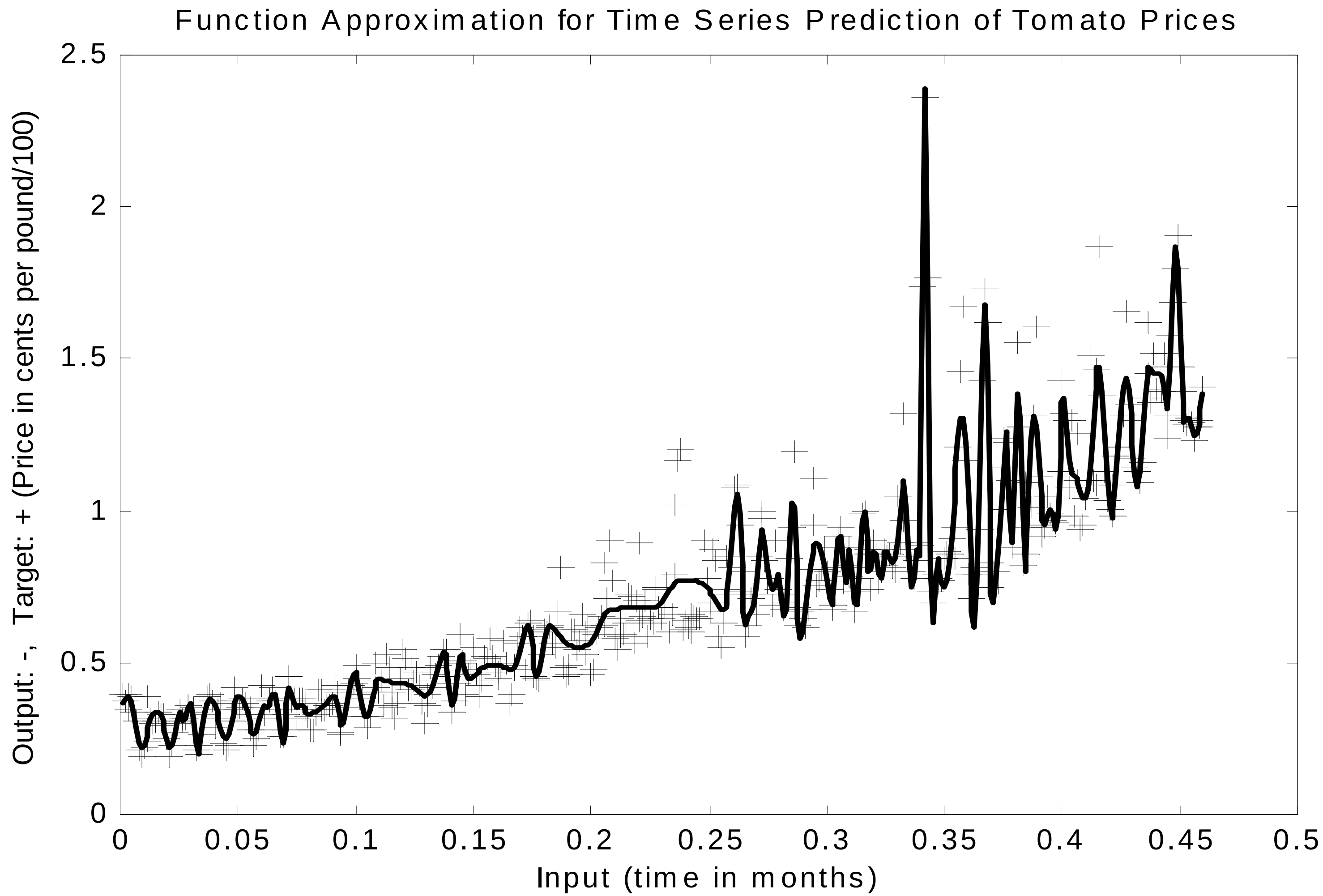
Experimental Results

- We describe, in this section, the experimental results obtained by using neural networks to the problem of forecasting tomato prices in the U.S. Market.
- We show results of the application of several architectures and different learning algorithms to decide on the best one for this problem.
- We also compare at the end the results of the neural network approach with the results of linear regression models, to measure the difference in forecasting power of both methodologies

First experiment.

- First, we will describe the results of applying modular neural networks to the time series of tomato prices. We used the monthly data from 1960 to 1999 for training a Modular Neural Network with four Modules, each of the modules with 80 neurons and one hidden layer.
- We show in the following Figure the result of training the modular neural network with this data.
- In this Figure, we can appreciate how the modular neural network approximates very well the real time series of tomato prices over the relevant period of time.

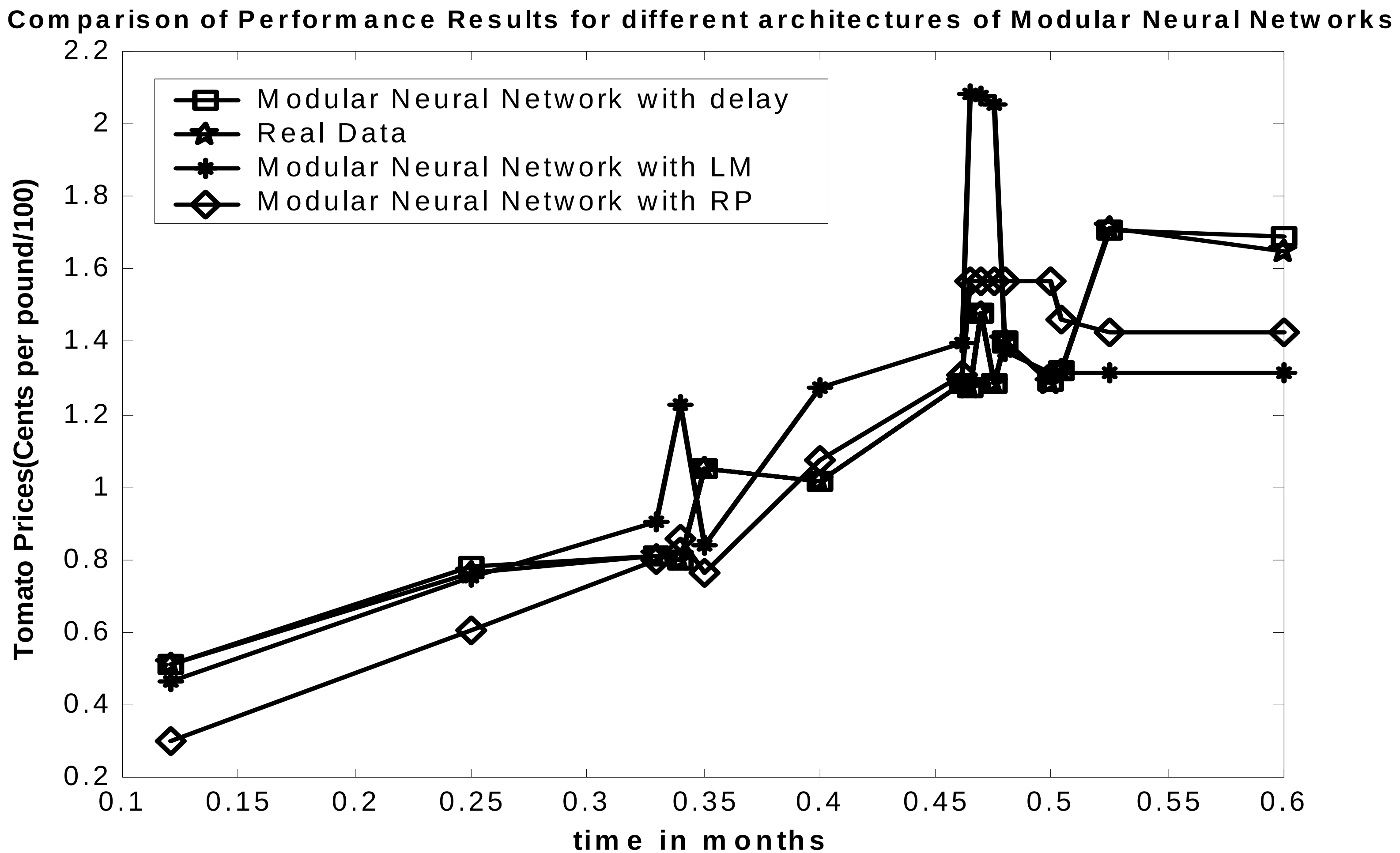
Figure: Modular Neural Network (MNN) for tomato prices with the Levenberg-Marquardt training Algorithm



Results

- We have to mention that the results shown in the previous Figure are for the best modular neural network that we were able to find for this problem. We show in the following Figure the comparison between several of the modular neural networks that we tried in our experiments.
- From this Figure we can appreciate that the modular neural network with one time delay and Leverberg-Marquardt (LM) training algorithm is the one that fits best the data and for this reason is the one selected

Figure: Comparison of performance results for several modular neural networks



Comparison Monolithic vs Modular

- We show in this Figure the comparison of the best monolithic network against the best modular neural network.
- The modular network clearly fits better the real data of the problem

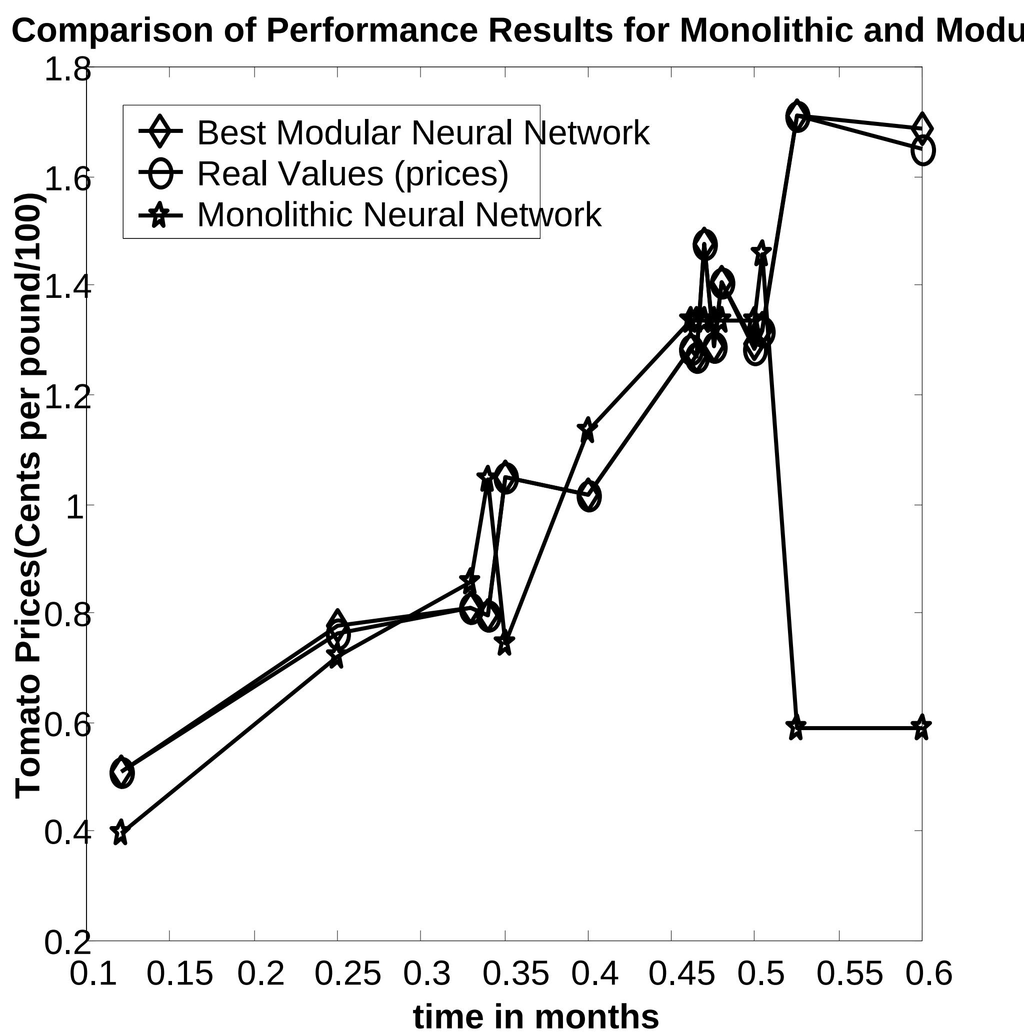


Figure 8.- Tomato prices of US and Mexico by weekly periods, using a MNN of 6 modules.

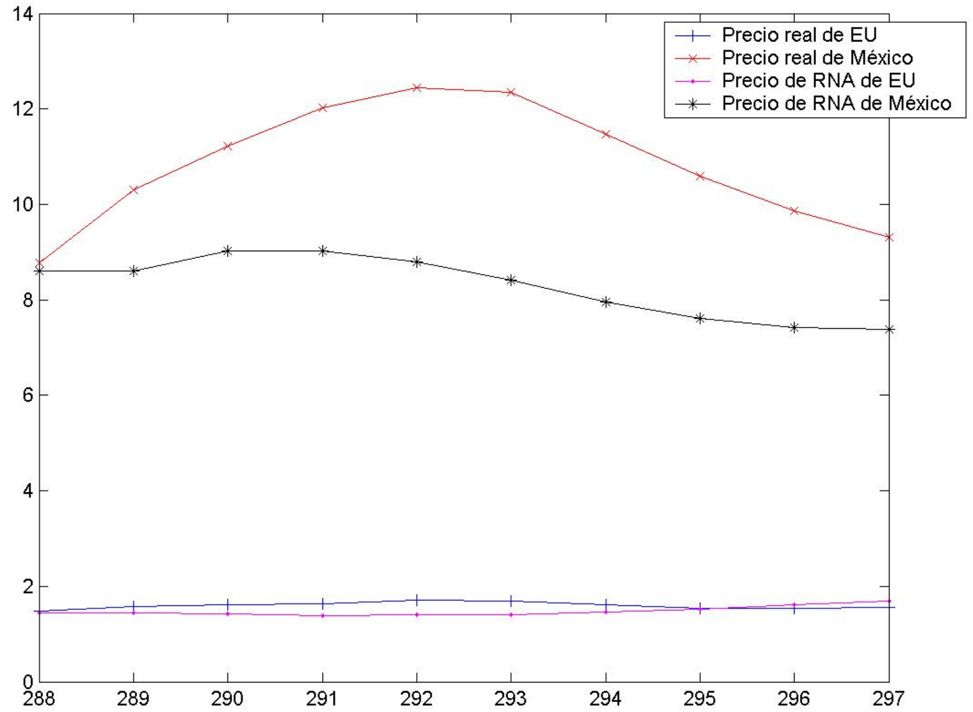


Table1.- MNN Architecture with 6 modules for Forecasting

Modules	Method	Layer 1	Layer 2	Epochs	Error	Gradient	µ
1	Trainlm	11	10	500	0.000001	0.0001	0.001
2	Trainlm	11	10	500	0.000001	0.0001	0.001
3	Trainlm	10	10	500	0.000001	0.0001	0.001
4	Trainlm	10	9	500	0.000001	0.0001	0.001
5	Trainlm	8	10	500	0.000001	0.0001	0.001
6	Trainlm	10	8	500	0.000001	0.0001	0.001

Figure : Tomato prices of US and Mexico by weekly periods, using MNN of 8 modules.

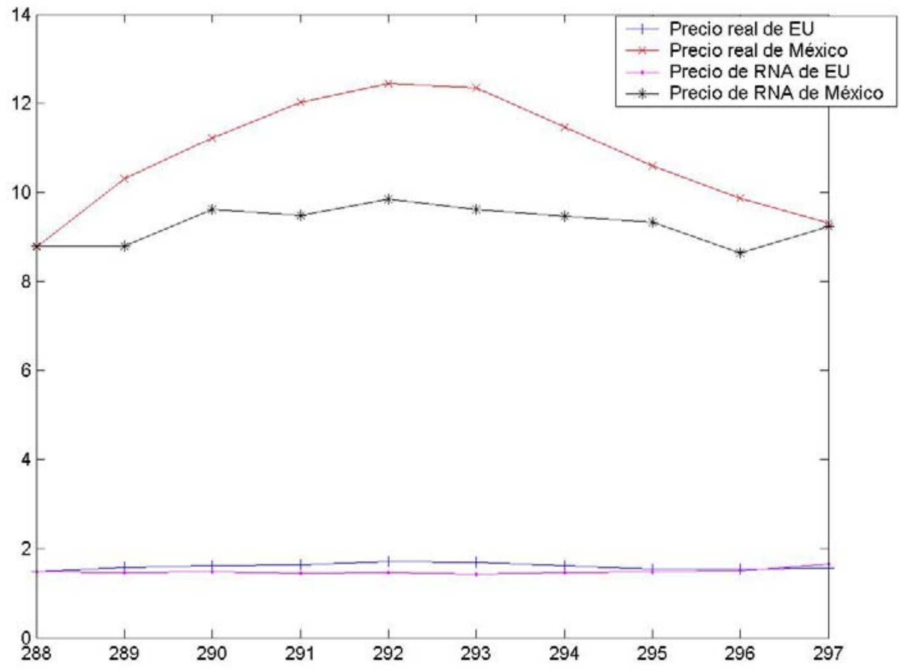
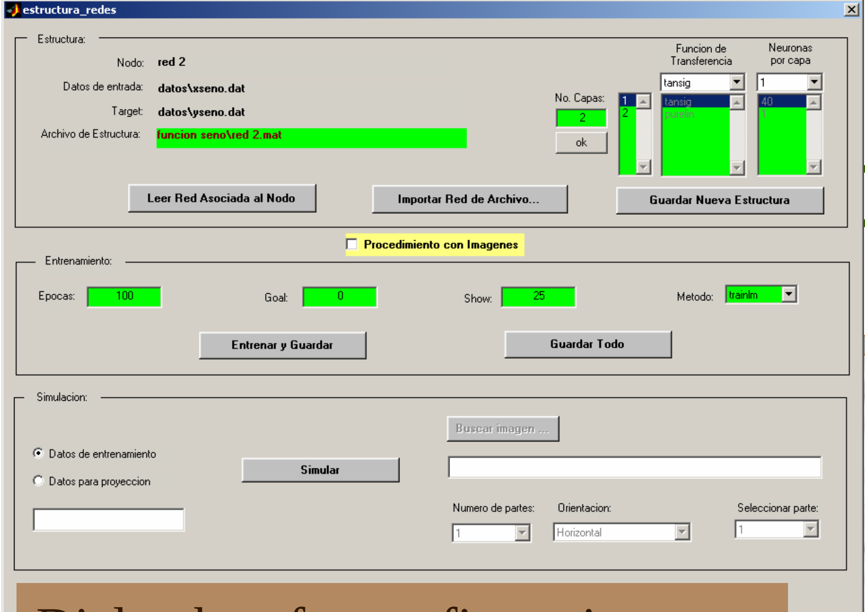


Table 2.- MNN Architecture with 8 modules for Forecasting

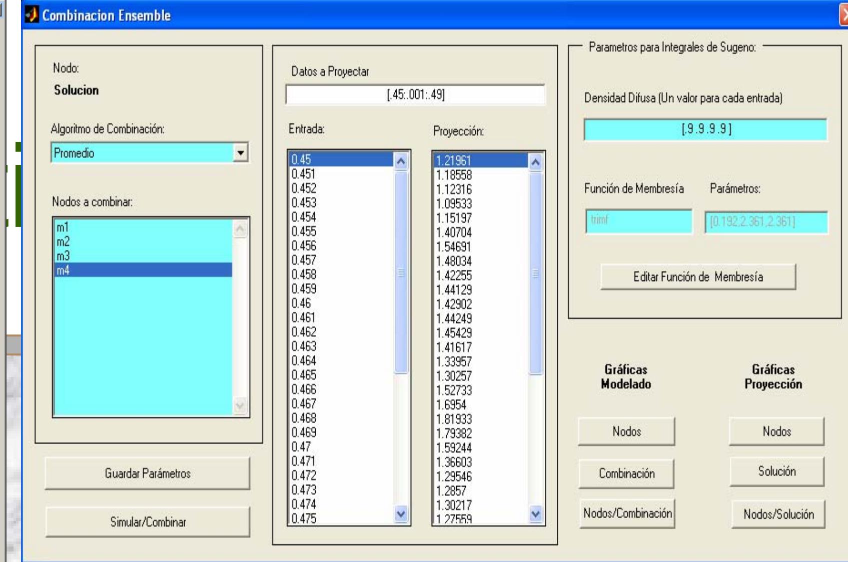
Modules	Method	Layer 1	Layer 2	Epochs	Error	Gradient	µ
1	Trainlm	10	9	700	0.000001	0.0001	0.001
2	Trainlm	11	9	700	0.000001	0.0001	0.001
3	Trainlm	10	10	700	0.000001	0.0001	0.001
4	Trainlm	10	9	700	0.000001	0.0001	0.001
5	Trainlm	8	10	700	0.000001	0.0001	0.001
6	Trainlm	10	8	700	0.000001	0.0001	0.001
7	Trainlm	11	9	700	0.000001	0.0001	0.001
8	Trainlm	9	10	700	0.000001	0.0001	0.001

Software for testing the approach

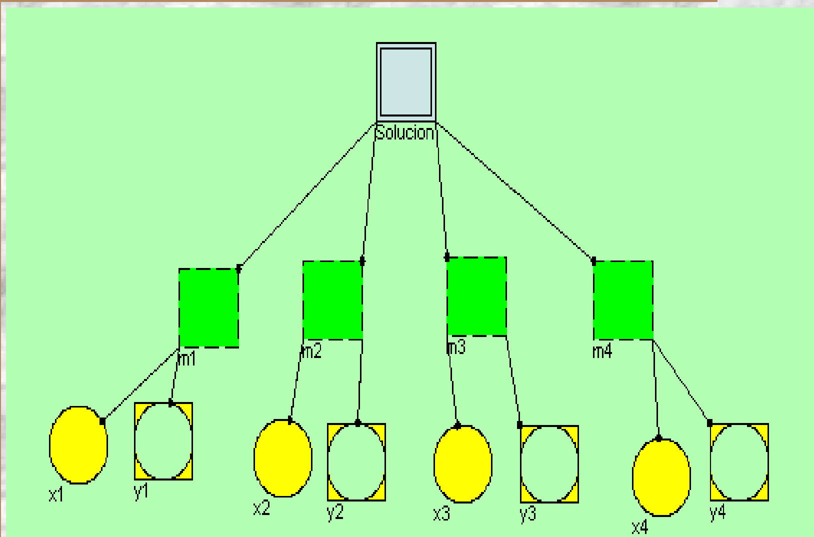
- *We developed a software tool (HMR) to experiment with new neural multi-net structures, including ensemble and modular approaches. This tool allow us to draw models, set parameters, save as a project and generate files with results, always in a user friendly graphic environment. The tool was developed in MATLAB and C programming languages.*
- *Other feature is the implementation of the Sugeno Integral formulas as modular integrated as well of other integrated methods, this program was developed to allow the combination of any number of elements.*



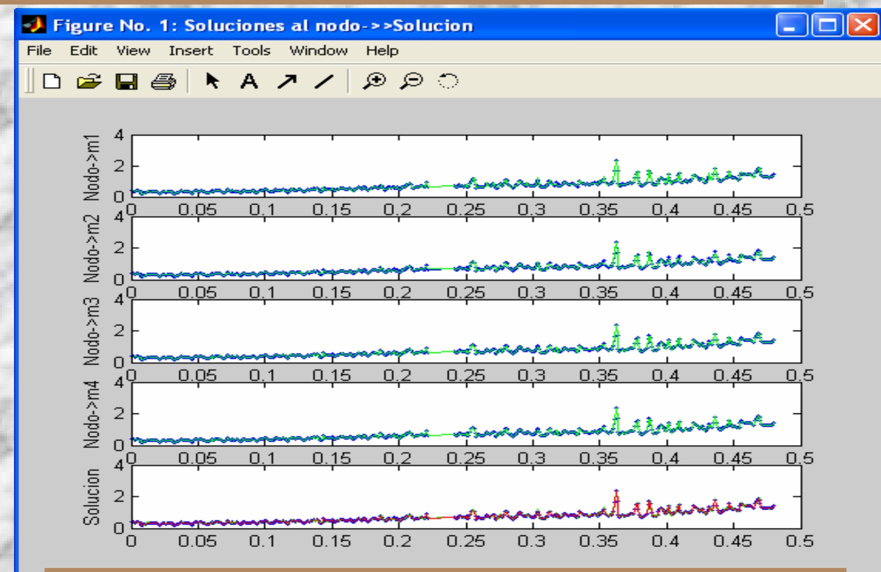
Dialog box for configuration, train and test each neural net



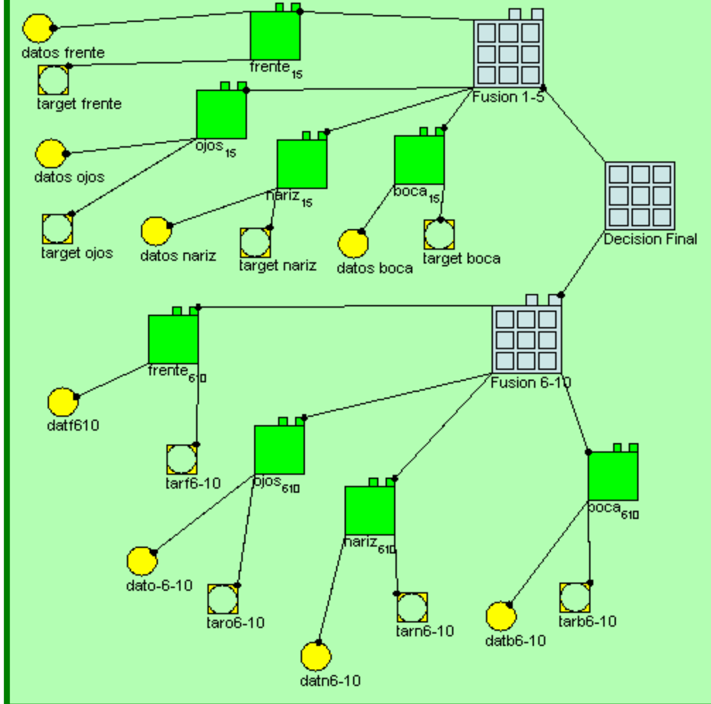
Dialog box to combine nodes in ensembles.



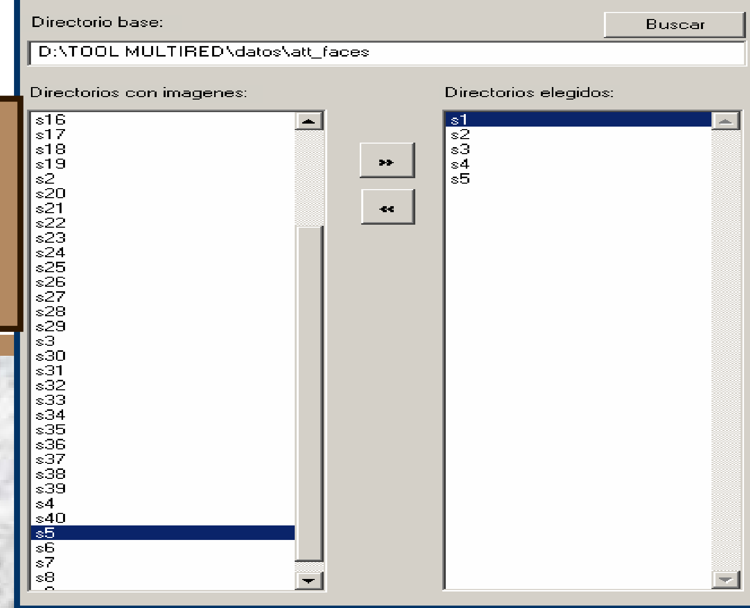
Model: Tomates1

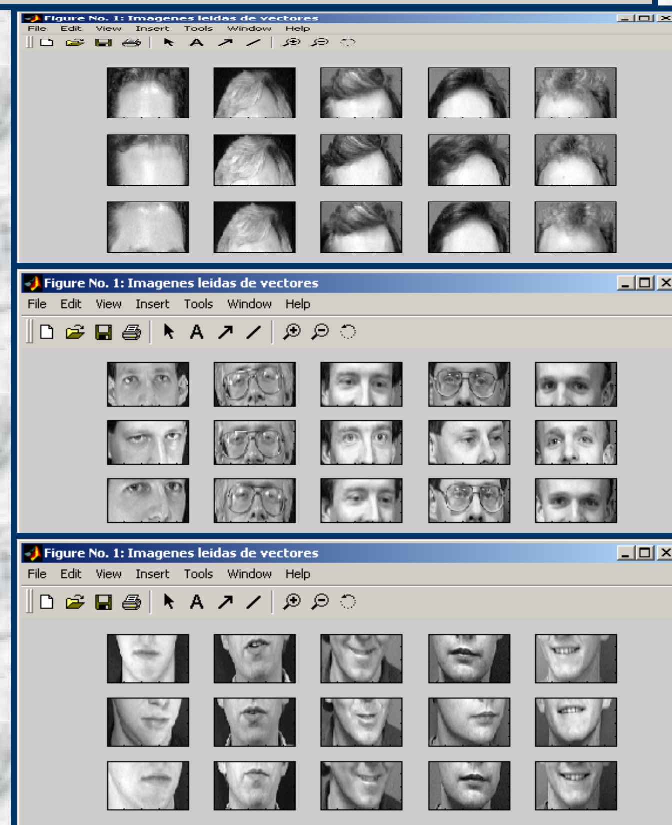


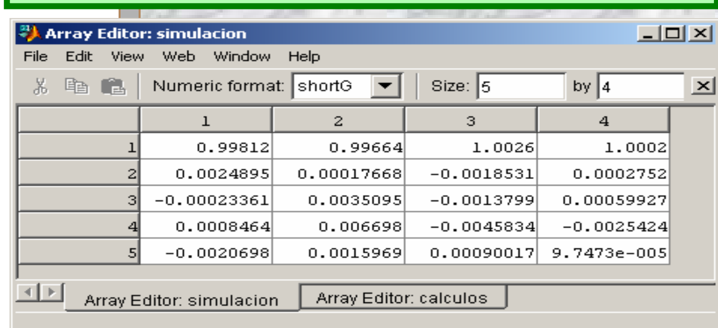
Graphic of each module and the combination results



Model for face recognition

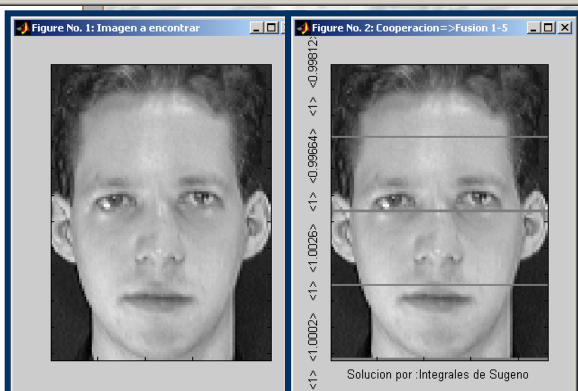






	1	2	3	4
1	0.99812	0.99664	1.0026	1.0002
2	0.0024895	0.00017668	-0.0018531	0.0002752
3	-0.00023361	0.0035095	-0.0013799	0.00059927
4	0.0008464	0.006698	-0.0045834	-0.0025424
5	-0.0020698	0.0015969	0.00090017	9.7473e-005

The value of the Sugeno Integral in each module



Results of the face recognition

Conclusions 1

- In light of the results of our proposed hybrid modular approach, we have to notice that using the modular approach for human face, fingerprint and voice recognition is a good alternative with respect to existing methods, in particular, monolithic, gating or voting methods.
- Modularity may imply reduction of number of parameters, which will allow and increase in computing speed and better generalization capabilities
- If there are changes in the environment, modular networks enable changes in an easier way, since there is no need to modify the whole system, only the modules that are affected by this change

Conclusions 2

- The results for the problem of voice recognition are very good and show the feasibility of the HGA approach for MNN topology optimization.

- The future work will consist in the application of HGA to MNN optimization in more complex problems of pattern Recognition.

Conclusions 3

- We also used modular neural networks for simulation and forecasting time series of consumer goods in the U.S. and Mexico Markets. We have considered a real case to test our approach, which is the problem of time series prediction of tomato prices in the U.S and Mexico markets.
- The performance of the modular neural networks was also compared with monolithic neural networks. The forecasting ability of modular neural networks was clearly superior than the one of monolithic neural networks

Conclusions

- Finally, we can conclude that the modular neural network approach with fuzzy integration of responses is a method that can be applied to problems of pattern recognition and time series prediction.
- We think that it is possible to apply this approach to problems in other areas, like control and monitoring of complex plants.

Future Work

- As future research work, we propose the study of methods for pre-processing, feature extraction, and compression of the data, like principal components analysis, eigenfaces, wavelets, or any other method that may improve the performance of the system.
- Other future work include considering different methods of fuzzy response integration.