10

Fingerprint Recognition with Modular Neural Networks and Fuzzy Measures

We describe in this chapter a new approach for fingerprint recognition using modular neural networks with a fuzzy logic method for response integration. We describe a new architecture for modular neural networks for achieving pattern recognition in the particular case of human fingerprints. Also, the method for achieving response integration is based on the fuzzy Sugeno integral. Response integration is required to combine the outputs of all the modules in the modular network. We have applied the new approach for fingerprint recognition with a real database of fingerprints obtained from students of our institution.

10.1 Introduction

Among all the biometric techniques, fingerprint-based identification is the oldest method, which has been successfully used in numerous applications. Everyone is known to have unique, immutable fingerprints. A fingerprint is made of a series of ridges and furrows on the surface of the finger. The uniqueness of a fingerprint can be determined by the pattern of ridges and furrows as well as the minutiae points. Minutiae points are local ridge characteristics that occur at either a ridge bifurcation or a ridge ending. Fingerprint matching techniques can be placed into two categories: minutiae-based and correlation based. Minutiae-based techniques first find minutiae points and then map their relative placement on the finger. However, there are some difficulties when using this approach. It is difficult to extract the minutiae points accurately when the fingerprint is of low quality. Also this method does not take into account the global pattern of ridges and furrows. The correlation-based method is able to overcome some of the difficulties of the minutiae-based approach. However, it has some of its own shortcomings. Correlation-based techniques require the precise location of a registration point and are affected by image translation and rotation.

Patricia Melin and Oscar Castillo: *Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing*, StudFuzz **172**, 207–221 (2005)
www.springerlink.com

Fingerprint matching based on minutiae has problems in matching different sized (unregistered) minutiae patterns. Local ridge structures can not be completely characterized by minutiae. We are trying an alternative representation of fingerprints, which will capture more local information and yield a fixed length code for the fingerprint. The matching will then hopefully become a relatively simple task of calculating the Euclidean distance between the two codes.

We describe algorithms, which are more robust to noise in fingerprint images and deliver increased accuracy in real-time. This is very difficult to achieve with any other technique. We are investigating methods to pool evidence from various matching techniques to increase the overall accuracy of the system. In a real application, the sensor, the acquisition system and the variation in performance of the system over time is very critical. We are also testing our system on a limited number of users to evaluate the system performance over a period of time.

The basic idea of the new approach is to divide a human fingerprint in to three different regions: the top, the middle and the bottom. Each of these regions is assigned to one module of the neural network. In this way, the modular neural network has three different modules, one for each of the regions of the human fingerprint. At the end, the final decision of 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 top, middle and bottom of a human specific fingerprint. Other approaches in the literature use other types of integration modules, like voting methods, majority methods, and neural networks.

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 fingerprint recognition, methods of weighted statistical average do not work well due to the nature of the fingerprint recognition problem. For these reasons, a new approach for fingerprint recognition using modular neural networks and fuzzy integration of responses is described in this chapter.

The new approach for fingerprint recognition was tested with a database of students and professors from our institution. This database was collected at our institution using a special scanner. The results with our new approach for fingerprint recognition on this database were excellent.

10.2 Some Basic Concepts of Fingerprint Recognition

When we interact with others we are used to identifying them by their physical appearance, their voice, or other sensory data. When we need proof of identity beyond physical appearance we obtain a signature or we look at a photo identification card. In Cyberspace, where people need to interact with digital systems or with one another remotely, we do not have these tried and true means of identification available. In almost all cases we cannot see, hear, or obtain a signature from the person with whom we are interacting. Biometrics, the measurement of a unique physical characteristic, is an ideal solution to the problem of digital identification. Biometrics makes it possible to identify ourselves to digital systems, and through these systems identify ourselves to others in Cyberspace. With biometrics we create a digital persona that makes our transactions and interactions in Cyberspace convenient and secure. Of all the biometrics available, including face, iris and retina scanning, voice identification, and others, the fingerprint is one of the most convenient and foolproof. The advantages of fingerprint biometrics for the purpose of personal digital identification include:

1. Each and every one of our ten fingerprints is unique, different from one another and from those of every other person. Even identical twins have unique fingerprints!
2. Unlike passwords, PIN codes, and smartcards that we depend upon today for identification, our fingerprints are impossible to lose or forget, and they can never be stolen.
3. We have ten fingerprints as opposed to one voice, one face or two eyes.
4. Fingerprints have been used for centuries for identification, and we have a substantial body of real world data upon which to base our claim of the uniqueness of each fingerprint. Iris scanning, for instance, is an entirely new science for which there is little or no real world data.

The skin on the inside surfaces of our hands, fingers, feet, and toes is "ridged" or covered with concentric raised patterns. These ridges are called friction ridges and they serve the useful function of making it easier to grasp and hold onto objects and surfaces without slippage. It is the many differences in the way friction ridges are patterned, broken, and forked which make ridged skin areas, including fingerprints, unique.

Fingerprints are extremely complex. In order to "read" and classify them, certain defining characteristics are used, many of which have been established by law enforcement agencies as they have created and maintained larger and larger databases of prints. We usually have two types of fingerprint characteristics for use in identification of individuals: Global Features and Local Features. Global Features are those characteristics that you can see with the naked eye. Global Features include:

- Basic Ridge Patterns
- Pattern Area
- Core Area
- Delta

- Type Lines
- Ridge Count

The Local Features are also known as Minutia Points. They are the tiny, unique characteristics of fingerprint ridges that are used for positive identification. It is possible for two or more individuals to have identical global features but still have different and unique fingerprints because they have local features – minutia points – that are different from those of others.

Large volumes of fingerprints are collected and stored everyday in a wide range of applications including forensics, access control, and driver license registration. An automatic recognition of people based on fingerprints requires that the input fingerprint be matched with a large number of fingerprints in a database. To reduce the search time and computational complexity, it is desirable to classify these fingerprints in an accurate and consistent manner so that the input fingerprint is matched only with a subset of the fingerprints in the database.

Fingerprint classification is a technique to assign a fingerprint into one of the several pre-specified types already established in the literature, which can provide an indexing mechanism. Fingerprint classification can be viewed as a coarse level matching of the fingerprints. An input fingerprint is first matched at a coarse level to one of the pre-specified types and then, at a finer level, it is compared to the subset of the database containing that type of fingerprints only.

A critical step in automatic fingerprint matching is to automatically, and reliably extract minutiae from the input fingerprint images. However, the performance of a minutiae extraction algorithm relies heavily on the quality of the input fingerprint images. In order to ensure that the performance of an automatic fingerprint identification/verification system will be robust with respect to the quality of the fingerprint images, it is essential to incorporate a fingerprint enhancement algorithm in the minutiae extraction module.

10.3 Image Pre-Processing for the Fingerprints

To improve the performance of the fingerprint recognition system, we first need to pre-process the fingerprints. The pre-processing allows to extract the most important characteristics of the fingerprint. The raw images of the fingerprints are of 200 by 198 pixels and we have a database of 50 digital fingerprins from students of our institution. For achieving this pre-processing, we used a demo version (freely available) of the VeriFinger software. This computer program allows us to open our scanned fingerprint image and extract the most important points of the fingerprint. We show in Fig. 10.1 the original fingerprint image of a particular student. We also show in Fig. 10.2 the image after processing. This processed image is more clear (has less noise) and the important points are indicated.

Fig. 10.1. Original image of a fingerprint

Fig. 10.2. Image of a fingerprint after processing

The processed images are the ones used as input to the neural networks. We describe in the following section how these fingerprint images are used for achieving the goal of identification.

10.4 Architecture for Fingerprint Recognition

In the experiments performed in this research work, we used 50 fingerprints that were taken with a scanner from students and professors of our Institution (Quezada, 2004). The images were taken in such a way that they had 198 pixels wide and 200 pixels high, with a resolution of 300×300 ppi, and with a color representation of a gray scale, some of these images are shown in Fig. 10.3. In addition to the training data (50 fingerprints) we did use 10 images for each fingerprint that were obtained by applying noise in a random fashion, which was increased from 10 to 100%.

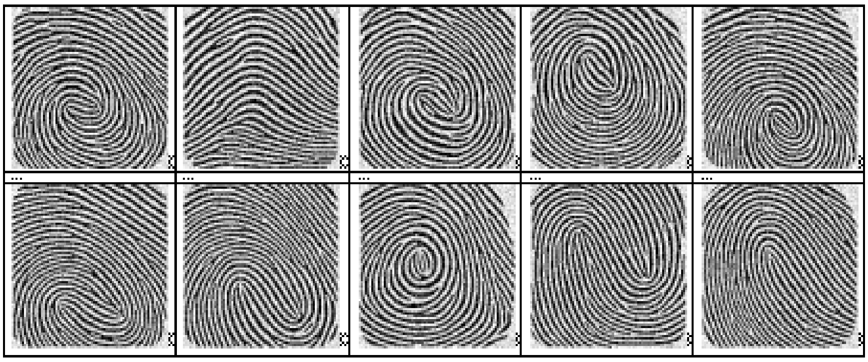

Fig. 10.3. Sample Images Used for Training

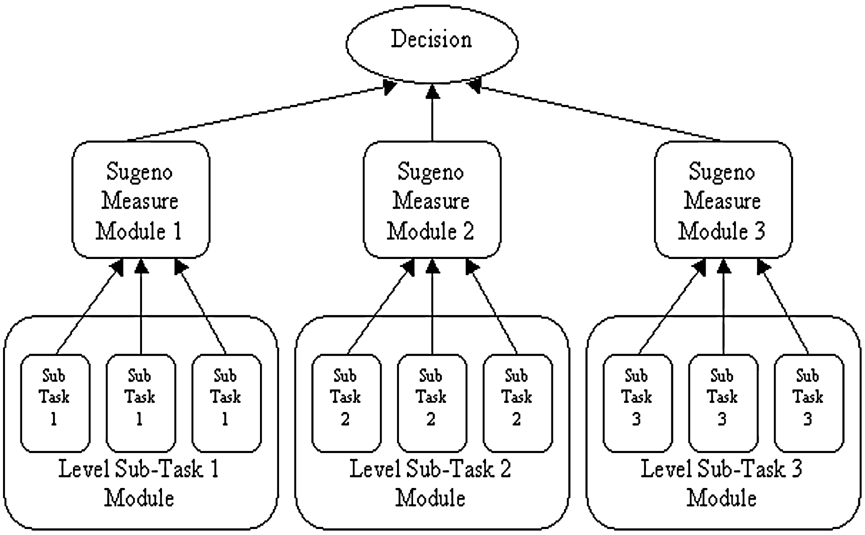

Fig. 10.4. Final Proposed Architecture

The architecture proposed for fingerprint recognition consists 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 Fig. 10.4.

The input to the modular system is a complete fingerprint image. For performing the neural network training, the images of the human fingerprints were divided in three different regions. The first region consists of the area on top, which corresponds to Sub Task 1. The second region consists of the area on the middle, which corresponds to Sub Task 2. The third region consists of the area on the bottom, which corresponds to Sub Task 3. An example of this image division is shown in Fig. 10.5.

Fig. 10.5. Example of Image Division

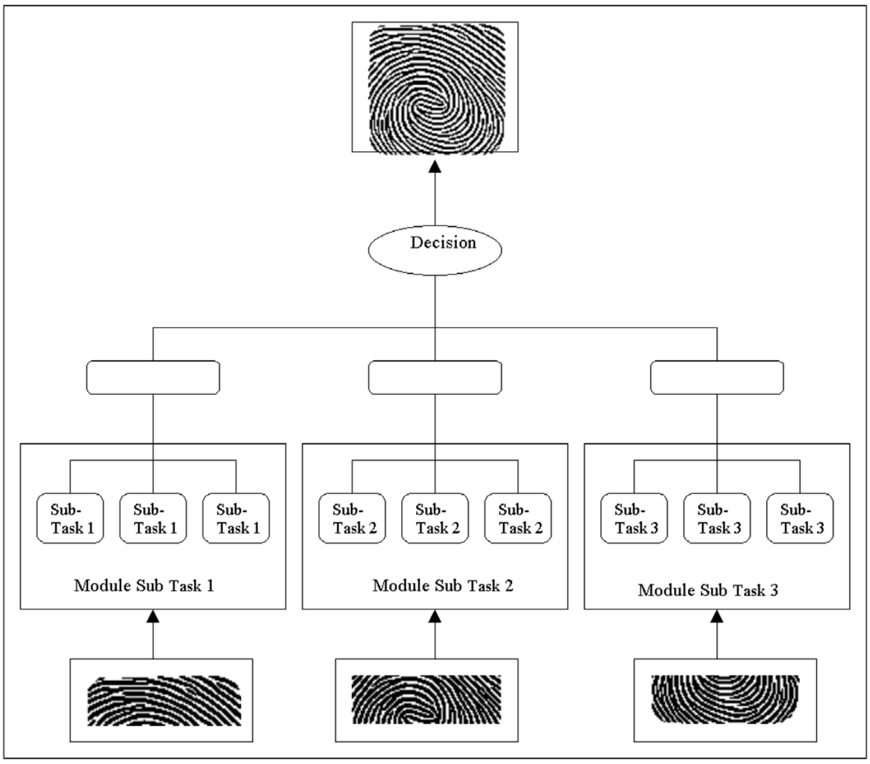

Fig. 10.6. Final architecture showing inputs and outputs

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 Fig. 10.6 an example of this.

The integration modules performs its task in two phases. In the first phase, it obtains two matrices. The first matrix, called h, of dimension 3×3, 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 3×3, stores the image number corresponding to the particular index.

Once the first phase is finished, the second phase is initiated, in which the decision is obtained. 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 to 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,

$$g(M_i) = h(A) + h(B) + \lambda h(A)h(B) \tag{10.1}$$

where λ is equal to 1. Once we have these measures, we select the largest one to show the corresponding image.

10.5 Genetic Algorithm for Optimization of the Modular Neural Network

To design the optimal modular neural network for fingerprint recognition, we need to optimize the architecture of each module (consisting of three neural networks) in the complete system. The genetic algorithm will simplify as much as possible each of the neural networks in the architecture. In other words, the number of nodes and layers will be reduced to the minimum necessary, in this way achieving the optimal modular architecture. The basic architecture of each neural network is shown in Fig. 10.7. We have for each neural network three hidden layers with a maximum of 200 nodes each, one input node and one output node. This information has to be represented in the genetic algorithm. The chromosome of the genetic algorithm will have the information about the number of layers and the number of nodes of each layer. This information is

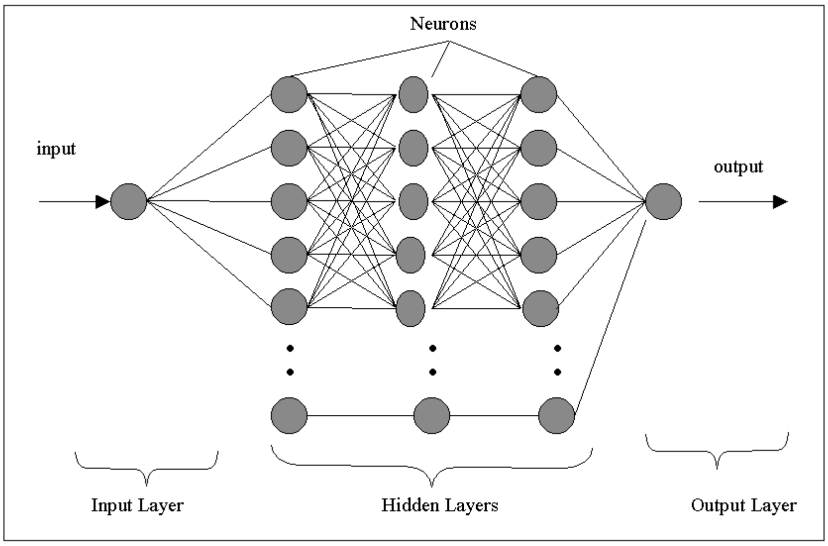

Fig. 10.7. Architecture of a neural network in the modular system

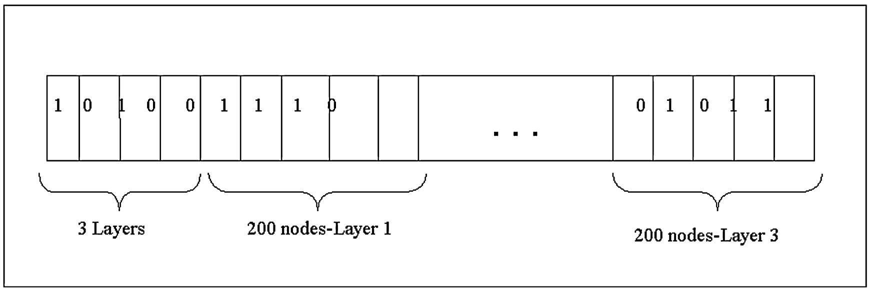

Fig. 10.8. Chromosome representation of a neural network

binary, since a 1 means the existence of a layer or a node, and a 0 means that no layer or node exists. In this way, a sequence of 0's and 1's will represent a specific architecture of a neural network. The genetic operators will be used to obtain better network architectures by the process of evolution, in which the fittest networks will survive and compete for producing new neural networks.

We show in Fig. 10.8 the chromosome representation used for an individual neural network in the genetic algorithm. We have to say that this chromosome contains hierarchical information, because the information about the layers has a higher hierarchy than the information about the nodes or neurons. This is the reason why this type of genetic algorithm is called a hierarchical genetic algorithm.

The chromosome contains the following data:

Total number of Bits = 613
Number of Layers = 3
Number of nodes in each layer = 200

The other information about the genetic algorithm is the following:

Size of the Population = 15 neural networks
Type of crossover = single-point crossover
Type of mutation: simple binary mutation
Type of selection: stochastic universal sampling

The objective function used for the genetic algorithm takes into account not only the error but also the number of nodes in the network (complexity). The basic idea is that we want good approximation, but it is also important that the size of the network is as small as possible. The following equation defines the objective function used:

$$F(z) = \alpha[\text{rank } (f_1(z))] + \beta \cdot f_2(z) \ . \tag{10.2}$$

Where:

$\alpha = 50$ (constant selected by the user)
$\beta = 1$ (constant selected by the user

$f1(z)$ = first objective (sum of squared errors produced by the neural network)
$f2(z)$ = second objective (number of nodes in the neural network)

10.6 Summary of Results for Fingerprint Recognition

We describe in this section the experimental results obtained with the proposed approach using the 50 images as training data. We show in Table 10.1 the relation between accuracy (measured as the percentage of correct results) and the percentage of noise in the figures.

Table 10.1. Relation between the % of noise and the % of correct results

% of Noise	% Accuracy
0	100
10	100
20	100
30	100
40	95
50	100
60	100
70	95
80	100
90	75
100	80

In Table 10.1 we show the relation that exists between the % of noise that was added in a random fashion to the testing data set, that consisted of the 50 original images, plus 500 additional images. We show in Fig. 10.9 sample images with noise.

In Table 10.2 we show the reliability results for the pattern recognition system. Reliability was calculated as shown in the following equation.

$$\text{Reliability} = \frac{\text{correct results} - \text{error}}{\text{correct results}} \tag{10.3}$$

We describe in more detail in the following lines the results of using the modular neural networks for fingerprint recognition. We first describe the results with a modular neural network that was manually designed. After that, we show the results of a modular neural network that was optimized using a hierarchical genetic algorithm.

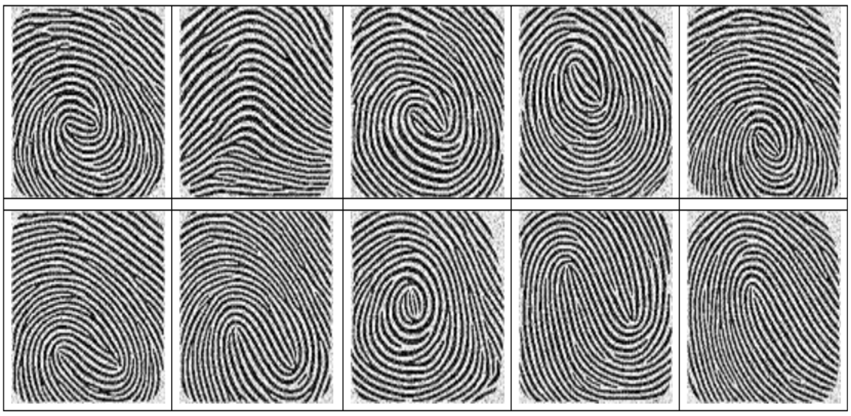

Fig. 10.9. Sample images with noise for testing

Table 10.2. Relation between reliability and accuracy

% Errors	% Reliability	% Correct Results
0	100	100.00
0	100	100.00
0	100	100.00
0	100	100.00
5	94.74	95.00
0	100	100.00
0	100	100.00
5	94.74	95.00
0	100	100.00
25	66.67	75.00
20	75	80.00

10.6.1 Recognition Results of the Modular Network without Genetic Algorithm Optimization

We first describe the results of using a modular neural network that was manually design for achieving fingerprint recognition. The number of nodes and layers in each neural network of the architecture was find through experimentation. We show in Table 10.3 the best parameters of the modular neural network, that were find through experimentation. These parameters are the result of trial and error in combination with some experience in designing neural networks for different problems. In all the experiments the number of epoch was 1000 and the error goal 0.01.

We show in Table 10.4 the detailed results for a sample of 20 fingerprints for different levels of noise. For example, fingerprint number 6 is recognized up to 60% of noise with a Sugeno measure of 0.615385. Another image that is not recognized at any level of noise is image 13.

Table 10.3. Parameters of the modular neural network manually found

Module	Type of Network	Training Function	Transfer Function	Number of Layers	Neuros in Hidden Layers	Training Error
	Backpro	Trainrp	Tansig	4	200	Mse
1	Backpro	Trainrp	Tansig	4	160	Mse
	Backpro	Trainrp	Tansig	4	160	Mse
	Backpro	Trainrp	Tansig	4	200	Mse
2	Backpro	Trainrp	Tansig	4	180	Mse
	Backpro	Trainrp	Tansig	4	180	Mse
	Backpro	Trainrp	Tansig	4	190	Mse
3	Backpro	Trainrp	Tansig	4	240	Mse
	Backpro	Trainrp	Tansig	4	190	Mse

Table 10.4. Results for each fingerprint with different levels (%) of noise applied

Finger-print	10% MS	20% MS	30% MS	40% MS	50% MS	60% MS	70% MS
1	3.043141	3.043141	3.043141	3.043141	3.043141	3.043141	3.043141
2	2.560881	2.560881	2.560881	2.560881	2.560881	2.560881	2.560881
3	2.999999	2.999999	2.999999	2.999999	2.999999	2.999999	2.999999
4	0.777779	0.777779	0.777779	0.777779	0.777779	0.777779	0.777779
5	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000
6	0.615385	0.615385	0.615385	0.615385	0.615385	0.615385	NO
7	2.999998	2.999998	NO	NO	NO	NO	NO
8	3.000002	3.000002	3.000002	3.000002	3.000002	3.000002	3.000002
9	3.000003	3.000003	3.000003	3.000003	3.000003	3.000003	3.000003
10	2.999999	2.999999	2.999999	2.999999	2.999999	2.719862	2.719862
11	NO	NO	NO	NO	NO	NO	NO
12	2.504059	2.504059	2.504059	2.504059	2.504059	2.504059	2.069608
13	NO	NO	NO	NO	NO	NO	NO
14	2.952918	2.952918	2.952918	2.952918	2.952918	2.952918	2.952918
15	2.999999	2.999999	2.999999	2.999999	2.999999	2.999999	2.999999
16	1.250000	1.250000	1.250000	1.250000	1.250000	1.250000	1.250000
17	2.384798	2.384798	2.384798	2.384798	2.384798	2.384798	2.384798
18	2.999999	2.999999	2.999999	2.999999	2.999999	2.999999	2.999999
19	2.232802	2.232802	2.232802	2.232802	2.232802	2.232802	2.232802
20	2.999999	2.999999	2.999999	2.999999	2.999999	2.999999	2.999999

We now show in Fig. 10.10 the relation between the recognition percentage and the level of noise applied. As we can appreciate from this figure the results are not as good as we expected. We have a 90% recognition rate up to a 20% of noise applied and then performance goes down to 80%. We will see in the next section that optimizing the modular network with a genetic algorithm improves these results tremendously.

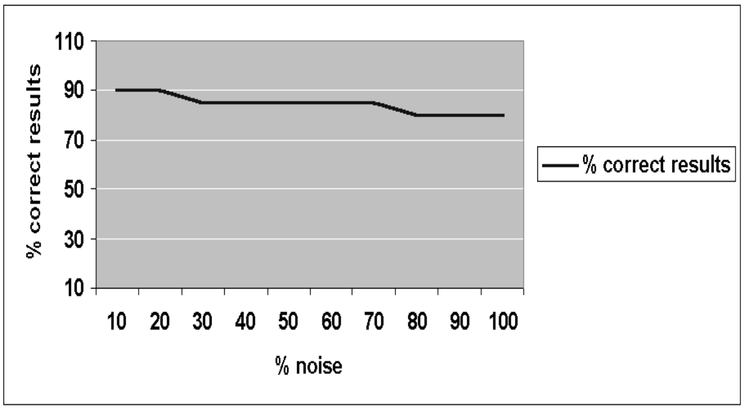

Fig. 10.10. Relation between the percentage of recognition and the level of noise applied

10.6.2 Recognition Results of the Modular Network with Genetic Algorithm Optimization

As we described previously, it is possible to use a hierarchical genetic algorithm for optimizing the architecture of the neural networks. For this reason, we applied the hierarchical genetic algorithm approach for minimizing the number of layers and nodes for all the networks in the respective modules that form the modular neural network. We show in Table 10.5 the parameters used in the genetic algorithm for optimizing the networks.

We now show in Table 10.6 the results obtained with the genetic algorithm in all of the experiments the number of generations was 1000. We can appreciate from this table that we now have the optimal number of layers and

Table 10.5. Parameters for optimizing the modular neural network

Module	Population	Number of Generations	Num. Bits	Best	Num. of Network	Num. Hidden Layers	Neurons in Hidden Layers	
	15	200	613	242	12	2	68	86
1	15	200	613	231	13	2	71	76
	15	200	613	239	3	2	72	74
	20	200	613	246	10	1	82	–
2	20	200	613	246	19	1	89	–
	20	200	613	234	14	1	79	–
	25	300	613	226	12	2	74	76
3	25	300	613	228	5	2	71	79
	25	300	613	226	12	2	74	76

Table 10.6. Results of the genetic algorithm application

Module	Type of Network	Training Function	Transfer Function	Num. of Layers	Neurons in Hidden Layers		Error Goal
	Backpro	Trainrp	Tansig	4	68	86	0.01
1	Backpro	Trainrp	Tansig	4	74	76	0.02
	Backpro	Trainrp	Tansig	3	82	–	0.01
	Backpro	Trainrp	Tansig	4	71	76	0.01
2	Backpro	Trainrp	Tansig	4	71	79	0.02
	Backpro	Trainrp	Tansig	3	89	–	0.01
	Backpro	Trainrp	Tansig	4	72	74	0.01
3	Backpro	Trainrp	Tansig	4	74	76	0.02
	Backpro	Trainrp	Tansig	3	79	–	0.01

nodes for each of the neural networks for the three modules of the complete architecture.

Now we show in Fig. 10.11 the relation between the percentage of recognition, achieved in this case, with respect to the level of noise applied. We can clearly appreciate from this figure the improvement achieved due to the application of the genetic algorithm for optimizing the modular neural network architecture. Now we have a 100% level of recognition up to 70% of noise applied, and then after this, performance goes down (see Table 10.7). It is clear that finding the optimal architecture of the networks in the modules, results in an improvement on the recognition rate of the modular neural network.

We also have to point out that due to the application of the genetic algorithm for optimizing the networks in the three modules, the training time of the neural networks was reduced considerably. Of course, this is due to the fact that the networks are smaller and as consequence require less computation time.

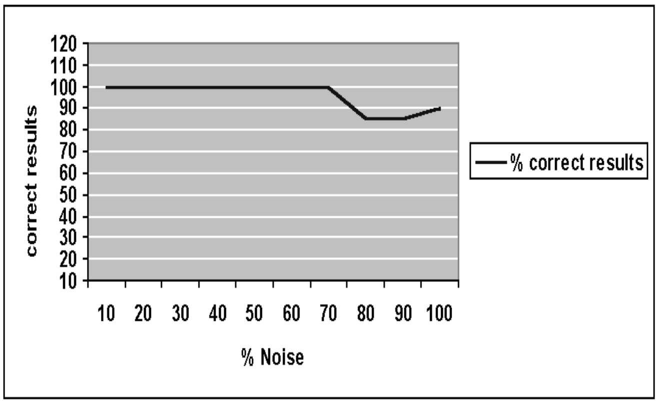

Fig. 10.11. Relation between the recognition rate and the level of noise

Table 10.7. Results for each fingerprint with different % of noise applied

Finger-print	10% MS	20% MS	30% MS	40% MS	50% MS	60% MS	70% MS
1	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000
2	2.771493	2.771493	2.771493	2.771493	2.771493	2.771493	2.461565
3	2.613652	2.613652	2.613652	2.613652	2.613652	2.613652	2.613652
4	2.907039	2.907039	2.907039	2.907039	2.907039	2.104775	2.104775
5	3.028653	3.028653	3.028653	3.028653	3.028653	3.028653	3.028653
6	3.000001	3.000001	3.000001	3.000001	3.000001	3.000001	3.000001
7	2.999997	2.999997	2.999997	2.999997	2.999997	2.999997	3.680908
8	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000
9	2.999999	2.999999	2.999999	2.999999	2.999999	2.999999	2.999999
10	2.999995	2.999995	2.999995	2.999995	2.999995	2.999995	2.999995
11	2.999998	2.999998	2.999998	2.999998	2.999998	2.999998	2.999998
12	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000
13	2.999997	2.999997	2.999997	2.999997	2.999997	2.999997	2.999997
14	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000
15	3.000001	3.000001	3.000001	3.000001	3.000001	3.000001	3.000001
16	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000
17	3.000001	3.000001	3.000001	3.000001	3.000001	3.000001	3.000001
18	3.348529	3.348529	3.348529	3.348529	3.348529	3.348529	3.348529
19	3.029244	3.029244	3.029244	3.029244	3.029244	3.029244	3.029244
20	3.000007	3.000007	3.000007	3.000007	3.000007	3.000007	3.000007

10.7 Summary

We described in this chapter the experimental results obtained with the proposed modular approach. In fact, we did achieve a 100% recognition rate on the testing data, even with an 70% level of applied noise. For the case of 100% level of applied noise, we did achieve a 90% recognition rate on the testing data. The testing data included 10 images for each fingerprint in the training data. These 10 images were obtained by applying noise in a random fashion, increasing the level of noise from 10 to 100%, to the training data. We also have to notice that it was achieved a 96.7% of average reliability with our modular approach. These percentage values were obtained by averaging. In light of the results of our proposed modular approach, we have to notice that using the modular approach for human fingerprint pattern recognition is a good alternative with respect to existing methods, in particular, monolithic, gating or voting methods. As future research work, we propose the study of methods for pre-processing the data, like principal components analysis, eigenvalues, or any other method that may improve the performance of the system. Other future work include considering different methods of fuzzy response integration, or considering evolving the number of layers and nodes of the neural network modules.