11

Voice Recognition with Neural Networks, Fuzzy Logic and Genetic Algorithms

We describe in this chapter the use of neural networks, fuzzy logic and genetic algorithms for voice recognition. In particular, we consider the case of speaker recognition by analyzing the sound signals with the help of intelligent techniques, such as the neural networks and fuzzy systems. We use the neural networks for analyzing the sound signal of an unknown speaker, and after this first step, a set of type-2 fuzzy rules is used for decision making. We need to use fuzzy logic due to the uncertainty of the decision process. We also use genetic algorithms to optimize the architecture of the neural networks. We illustrate our approach with a sample of sound signals from real speakers in our institution.

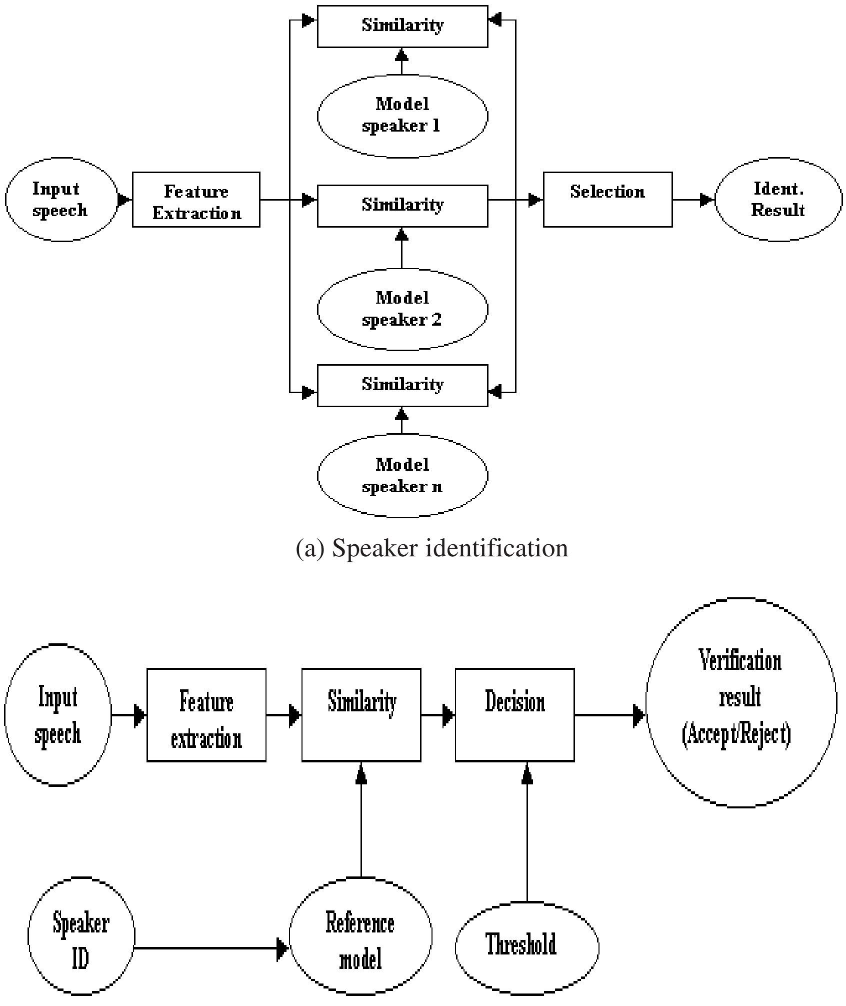
11.1 Introduction

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, telephone shopping, database access services, information services, voice mail, security control for confidential information areas, and remote access to computers.

Figure 11.1 shows the basic components of speaker identification and verification systems. Speaker identification is the process of determining which registered speaker provides a given utterance. Speaker verification, on the other hand, is the process of accepting or rejecting the identity claim of a speaker. Most applications in which a voice is used as the key to confirm the identity of a speaker are classified as speaker verification.

Speaker recognition methods can also be divided into text-dependent and text-independent methods. The former require the speaker to say key words or sentences having the same text for both training and recognition trials, whereas the latter do not rely on a specific text being spoken.

Patricia Melin and Oscar Castillo: *Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing*, StudFuzz **172**, 223–240 (2005)
www.springerlink.com



(a) Speaker identification

(b) Speaker Verification

Fig. 11.1. Basic structure of speaker recognition systems

Both text-dependent and independent methods share a problem however. These systems can be easily deceived because someone who plays back the recorded voice of a registered speaker saying the key words or sentences can be accepted as the registered speaker. To cope with this problem, there are methods in which a small set of words, such as digits, are used as key words and each user is prompted to utter a given sequence of key words that is randomly chosen every time the system is used. Yet even this method is not completely reliable, since it can be deceived with advanced electronic recording equipment that can reproduce key words in a requested order. Therefore, a text-prompted speaker recognition method has recently been proposed by (Matsui and Furui, 1993).

11.2 Traditional Methods for Speaker Recognition

Speaker identity is correlated with the physiological and behavioral characteristics of the speaker. These characteristics exist both in the spectral envelope (vocal tract characteristics) and in the supra-segmental features (voice source characteristics and dynamic features spanning several segments).

The most common short-term spectral measurements currently used are Linear Predictive Coding (LPC)-derived cepstral coefficients and their regression coefficients. A spectral envelope reconstructed from a truncated set of cepstral coefficients is much smoother than one reconstructed from LPC coefficients. Therefore it provides a stabler representation from one repetition to another of a particular speaker's utterances. As for the regression coefficients, typically the first- and second-order coefficients are extracted at every frame period to represent the spectral dynamics. These coefficients are derivatives of the time functions of the cepstral coefficients and are respectively called the delta- and delta-delta-cepstral coefficients.

11.2.1 Normalization Techniques

The most significant factor affecting automatic speaker recognition performance is variation in the signal characteristics from trial to trial (inter-session variability and variability over time). Variations arise from the speaker themselves, from differences in recording and transmission conditions, and from background noise. Speakers cannot repeat an utterance precisely the same way from trial to trial. It is well known that samples of the same utterance recorded in one session are much more highly correlated than samples recorded in separate sessions. There are also long-term changes in voices.

It is important for speaker recognition systems to accommodate to these variations. Two types of normalization techniques have been tried; one in the parameter domain, and the other in the distance/similarity domain.

11.2.2 Parameter-Domain Normalization

Spectral equalization, the so-called *blind equalization* method, is a typical normalization technique in the parameter domain that has been confirmed to be effective in reducing linear channel effects and long-term spectral variation (Furui, 1981). This method is especially effective for text-dependent speaker recognition applications that use sufficiently long utterances. Cepstral coefficients are averaged over the duration of an entire utterance and the averaged values subtracted from the cepstral coefficients of each frame. Additive variation in the log spectral domain can be compensated for fairly well by this method. However, it unavoidably removes some text-dependent and speaker specific features; therefore it is inappropriate for short utterances in speaker recognition applications.

11.2.3 Distance/Similarity-Domain Normalization

A normalization method for distance (similarity, likelihood) values using a likelihood ratio has been proposed by (Higgins et al., 1991). The likelihood ratio is defined as the ratio of two conditional probabilities of the observed measurements of the utterance: the first probability is the likelihood of the acoustic data given the claimed identity of the speaker, and the second is the likelihood given that the speaker is an imposter. The likelihood ratio normalization approximates optimal scoring in the Bayes sense.

A normalization method based on a posteriori probability has also been proposed by (Matsui and Furui, 1994). The difference between the normalization method based on the likelihood ratio and the method based on a posteriori probability is whether or not the claimed speaker is included in the speaker set for normalization; the speaker set used in the method based on the likelihood ratio does not include the claimed speaker, whereas the normalization term for the method based on a posteriori probability is calculated by using all the reference speakers, including the claimed speaker.

Experimental results indicate that the two normalization methods are almost equally effective (Matsui and Furui, 1994). They both improve speaker separability and reduce the need for speaker-dependent or text-dependent thresholding, as compared with scoring using only a model of the claimed speaker.

A new method in which the normalization term is approximated by the likelihood of a single mixture model representing the parameter distribution for all the reference speakers has recently been proposed. An advantage of this method is that the computational cost of calculating the normalization term is very small, and this method has been confirmed to give much better results than either of the above-mentioned normalization methods.

11.2.4 Text-Dependent Speaker Recognition Methods

Text-dependent methods are usually based on template-matching techniques. In this approach, the input utterance is represented by a sequence of feature vectors, generally short-term spectral feature vectors. The time axes of the input utterance and each reference template or reference model of the registered speakers are aligned using a dynamic time warping (DTW) algorithm and the degree of similarity between them, accumulated from the beginning to the end of the utterance, is calculated.

The hidden Markov model (HMM) can efficiently model statistical variation in spectral features. Therefore, HMM-based methods were introduced as extensions of the DTW-based methods, and have achieved significantly better recognition accuracies (Naik et al., 1989).

11.2.5 Text-Independent Speaker Recognition Methods

One of the most successful text-independent recognition methods is based on vector quantization (VQ). In this method, VQ code-books consisting of a small number of representative feature vectors are used as an efficient means of characterizing speaker-specific features. A speaker-specific code-book is generated by clustering the training feature vectors of each speaker. In the recognition stage, an input utterance is vector-quantized using the code-book of each reference speaker and the VQ distortion accumulated over the entire input utterance is used to make the recognition decision.

Temporal variation in speech signal parameters over the long term can be represented by stochastic Markovian transitions between states. Therefore, methods using an ergodic HMM, where all possible transitions between states are allowed, have been proposed. Speech segments are classified into one of the broad phonetic categories corresponding to the HMM states. After the classification, appropriate features are selected.

In the training phase, reference templates are generated and verification thresholds are computed for each phonetic category. In the verification phase, after the phonetic categorization, a comparison with the reference template for each particular category provides a verification score for that category. The final verification score is a weighted linear combination of the scores from each category.

This method was extended to the richer class of mixture autoregressive (AR) HMMs. In these models, the states are described as a linear combination (mixture) of AR sources. It can be shown that mixture models are equivalent to a larger HMM with simple states, with additional constraints on the possible transitions between states.

It has been shown that a continuous ergodic HMM method is far superior to a discrete ergodic HMM method and that a continuous ergodic HMM method is as robust as a VQ-based method when enough training data is available. However, when little data is available, the VQ-based method is more robust than a continuous HMM method (Matsui and Furui, 1993).

A method using statistical dynamic features has recently been proposed. In this method, a multivariate auto-regression (MAR) model is applied to the time series of cepstral vectors and used to characterize speakers. It was reported that identification and verification rates were almost the same as obtained by a HMM-based method.

11.2.6 Text-Prompted Speaker Recognition Method

In the text-prompted speaker recognition method, the recognition system prompts each user with a new key sentence every time the system is used and accepts the input utterance only when it decides that it was the registered speaker who repeated the prompted sentence. The sentence can be displayed as characters or spoken by a synthesized voice. Because the vocabulary is unlimited, prospective impostors cannot know in advance what sentence will be requested. Not only can this method accurately recognize speakers, but it can also reject utterances whose text differs from the prompted text, even if it is spoken by the registered speaker. A recorded voice can thus be correctly rejected.

This method is facilitated by using speaker-specific phoneme models, as basic acoustic units. One of the major issues in applying this method is how to properly create these speaker-specific phoneme models from training utterances of a limited size. The phoneme models are represented by Gaussian-mixture continuous HMMs or tied-mixture HMMs, and they are made by adapting speaker-independent phoneme models to each speaker's voice. In order, to properly adapt the models of phonemes that are not included in the training utterances, a new adaptation method based on tied-mixture HMMs was recently proposed by (Matsui and Furui, 1994).

In the recognition stage, the system concatenates the phoneme models of each registered speaker to create a sentence HMM, according to the prompted text. Then the likelihood of the input speech matching the sentence model is calculated and used for the speaker recognition decision. If the likelihood is high enough, the speaker is accepted as the claimed speaker.

Although many recent advances and successes in speaker recognition have been achieved, there are still many problems for which good solutions remain to be found. Most of these problems arise from variability, including speaker-generated variability and variability in channel and recording conditions. It is very important to investigate feature parameters that are stable over time, insensitive to the variation of speaking manner, including the speaking rate and level, and robust against variations in voice quality due to causes such as voice disguise or colds. It is also important to develop a method to cope with the problem of distortion due to telephone sets and channels, and background and channel noises.

From the human-interface point of view, it is important to consider how the users should be prompted, and how recognition errors should be handled. Studies on ways to automatically extract the speech periods of each person separately from a dialogue involving more than two people have recently appeared as an extension of speaker recognition technology.

This section was not intended to be a comprehensive review of speaker recognition technology. Rather, it was intended to give an overview of recent advances and the problems, which must be solved in the future (Furui, 1991).

11.2.7 Speaker Verification

The speaker-specific characteristics of speech are due to differences in physiological and behavioral aspects of the speech production system in humans. The main physiological aspect of the human speech production system is the vocal tract shape. The vocal tract modifies the spectral content of an acoustic wave as it passes through it, thereby producing speech. Hence, it is common in speaker verification systems to make use of features derived only from the vocal tract.

The acoustic wave is produced when the airflow, from the lungs, is carried by the trachea through the vocal folds. This source of excitation can be characterized as phonation, whispering, frication, compression, vibration, or a combination of these. Phonated excitation occurs when the airflow is modulated by the vocal folds. Whispered excitation is produced by airflow rushing through a small triangular opening between the arytenoid cartilage at the rear of the nearly closed vocal folds. Frication excitation is produced by constrictions in the vocal tract. Compression excitation results from releasing a completely closed and pressurized vocal tract. Vibration excitation is caused by air being forced through a closure other than the vocal folds, especially at the tongue. Speech produced by phonated excitation is called voiced, that produced by phonated excitation plus frication is called mixed voiced, and that produced by other types of excitation is called unvoiced.

Using cepstral analysis as described in the previous section, an utterance may be represented as a sequence of feature vectors. Utterances spoken by the same person but at different times result in similar yet a different sequence of feature vectors. The purpose of voice modeling is to build a model that captures these variations in the extracted set of features. There are two types of models that have been used extensively in speaker verification and speech recognition systems: stochastic models and template models. The stochastic model treats the speech production process as a parametric random process and assumes that the parameters of the underlying stochastic process can be estimated in a precise, well-defined manner. The template model attempts to model the speech production process in a non-parametric manner by retaining a number of sequences of feature vectors derived from multiple utterances of the same word by the same person. Template models dominated early work in speaker verification and speech recognition because the template model is intuitively more reasonable. However, recent work in stochastic models has demonstrated that these models are more flexible and hence allow for better modeling of the speech production process. A very popular stochastic model for modeling the speech production process is the Hidden Markov Model (HMM). HMMs are extensions to the conventional Markov models, wherein the observations are a probabilistic function of the state, i.e., the model is a doubly embedded stochastic process where the underlying stochastic process is not directly observable (it is hidden). The HMM can only be viewed through another set of stochastic processes that produce the sequence of observations.

The pattern matching process involves the comparison of a given set of input feature vectors against the speaker model for the claimed identity and computing a matching score. For the Hidden Markov models discussed above, the matching score is the probability that a given set of feature vectors was generated by a specific model. We show in Fig. 11.2 a schematic diagram of a typical speaker recognition system.

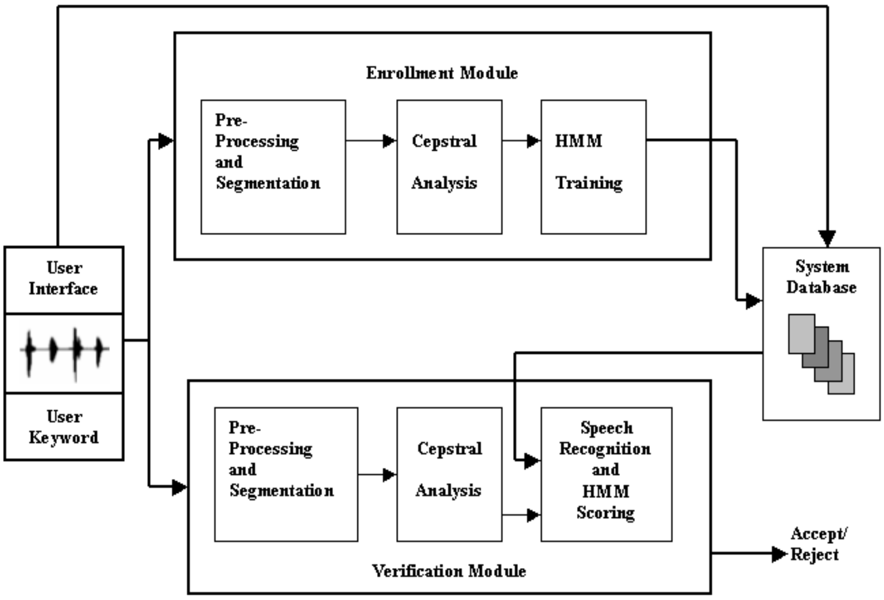
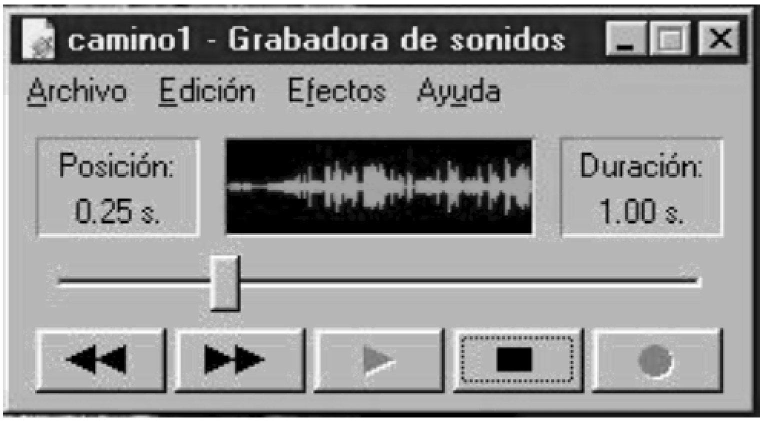


Fig. 11.2. Blocks diagram of a typical speaker recognition system

11.3 Voice Capturing and Processing

The first step for achieving voice recognition is to capture the sound signal of the voice. We use a standard microphone for capturing the voice signal. After this, we use the sound recorder of the Windows operating system to record the sounds that belong to the database for the voices of different persons. A fixed time of recording is established to have homogeneity in the signals. We show in Fig. 11.3 the sound signal recorder used in the experiments.

After capturing the sound signals, these voice signals are digitized at a frequency of 8 Khz, and as consequence we obtain a signal with 8008 sample points. This information is the one used for analyzing the voice.



Posición: 0.25 s.

Duración: 1.00 s.

Fig. 11.3. Sound recorder used in the experiments

We also used the Sound Forge 6.0 computer program for processing the sound signal. This program allows us to cancel noise in the signal, which may have come from environment noise or sensitivity of the microphones. After using this computer program, we obtain a sound signal that is as pure as possible. The program also can use fast Fourier transform for voice filtering. We show in Fig. 11.4 the use of the computer program for a particular sound signal.

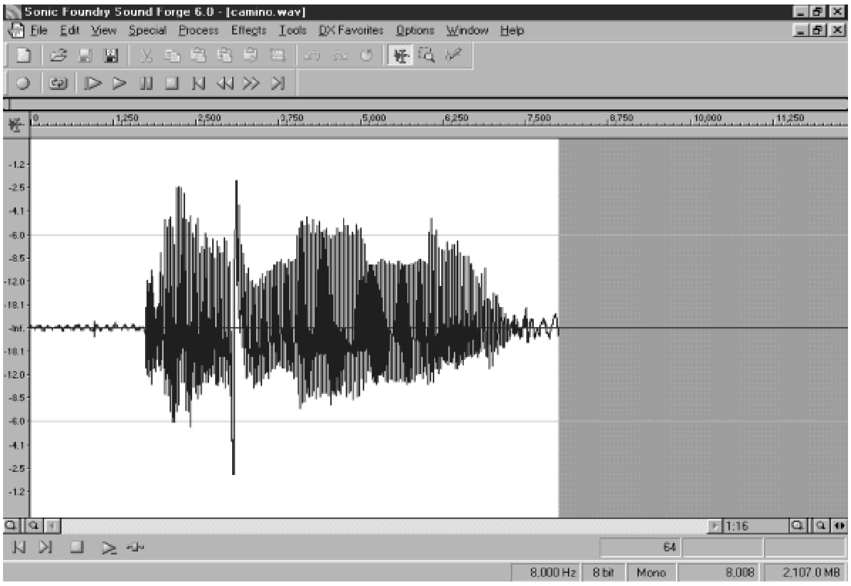
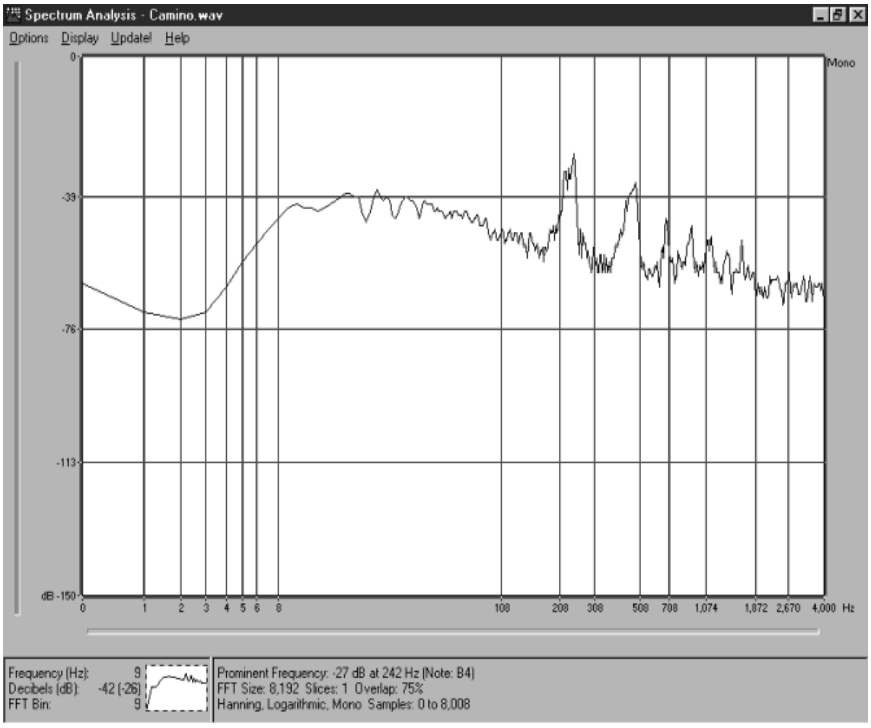


Fig. 11.4. Main window of the computer program for processing the signals



Frequency (Hz): 9
Decibels (dB): -42 (-26)
FFT Bin: 9

Prominent Frequency: -27 dB at 242 Hz (Note: B4)
FFT Size: 8,192 Slices: 1 Overlap: 75%
Hanning, Logarithmic, Mono Samples: 0 to 8,008

Fig. 11.5. Spectral analysis of a specific word using the FFT

We also show in Fig. 11.5 the use of the Fast Fourier Transform (FFT) to obtain the spectral analysis of the word "way" in Spanish.

11.4 Neural Networks for Voice Recognition

We used the sound signals of 20 words in Spanish as training data for a supervised feedforward neural network with one hidden layer. The training algorithm used was the Resilient Backpropagation (trainrp). We show in Table 11.1 the results for the experiments with this type of neural network.

The results of Table 11.1 are for the Resilient Backpropagation training algorithm because this was the fastest learning algorithm found in all the experiment (required only 7% of the total time in the experiments). The comparison of the time performance with other training methods is shown in Fig. 11.6.

We now show in Table 11.2 a comparison of the recognition ability achieved with the different training algorithms for the supervised neural networks. We are showing average values of experiments performed with all the training algorithms. We can appreciate from this table that the resilient backpropagation

Table 11.1. Results of feedforward neural networks for 20 words in Spanish

Stage	Time (min)	Num. of Words	No. Neurons	Words Recognized	% Recognition
1a.	11	20	50	17	85%
2a.	04	20	50	19	95%
1a.	04	20	70	16	80%
2a.	04	20	70	16	80%
3a.	02	20	25	20	100%
1a.	04	20	25	18	90%
1a.	03	20	50	18	90%
2a.	04	20	70	20	100%
2a.	04	20	50	18	90%
1a.	07	20	100	19	95%
2a.	06	20	100	20	100%
1a.	09	20	50	10	50%
1a.	07	20	75	19	95%
1a.	07	20	50	19	95%
2a.	06	20	50	20	100%
1a.	29	20	50	16	80%
1a.	43	20	100	17	85%
2a.	10	20	40	16	80%
3a.	10	20	80	16	80%
1a.	45	20	50	11	55%
2^a	30	20	50	15	75%
3^a.	35	20	70	16	80%

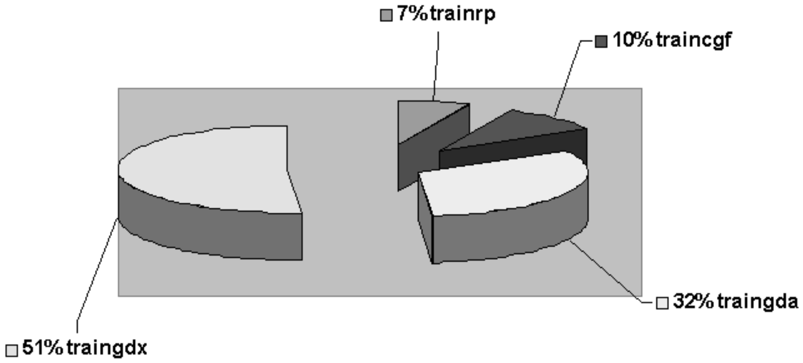


Fig. 11.6. Comparison of the time performance of several training algorithms

algorithm is also the most accurate method, with a 92% average recognition rate.

We describe below some simulation results of our approach for speaker recognition using neural networks. First, in Fig. 11.7 we have the sound signal

Table 11.2. Comparison of average recognition of four training algorithms

Method	Average Recognition
trainrp	92%
TRAINCGF-srchcha	85%
traingda	81%
traingdx	70%

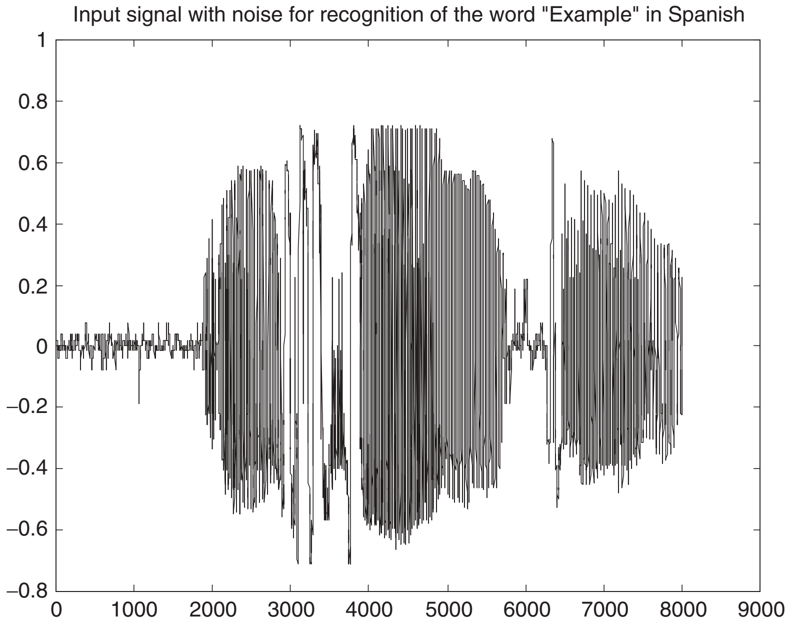


Fig. 11.7. Input signal of the word “example” in Spanish with noise

of the word “example” in Spanish with noise. Next, in Fig. 11.8 we have the identification of the word “example” without noise. We also show in Fig. 11.9 the word “layer” in Spanish with noise. In Fig. 11.10, we show the identification of the correct word “layer” without noise.

From the Figs. 11.7 to 11.10 it is clear that simple monolithic neural networks can be useful in voice recognition with a small number of words. It is obvious that words even with noise added can be identified, with at leat 92% recognition rate (for 20 words). Of course, for a larger set of words the recognition rate goes down and also computation time increases. For these reasons it is necessary to consider better methods for voice recognition.

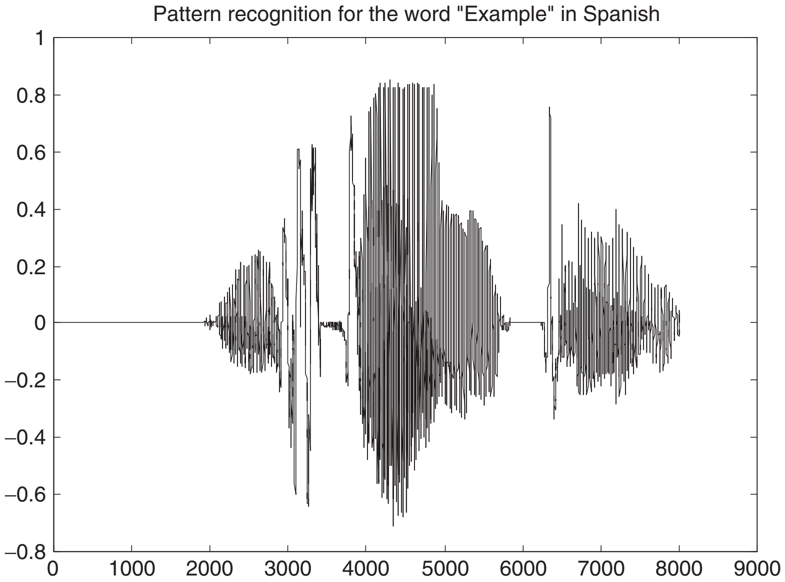


Fig. 11.8. Identification of the word "example"

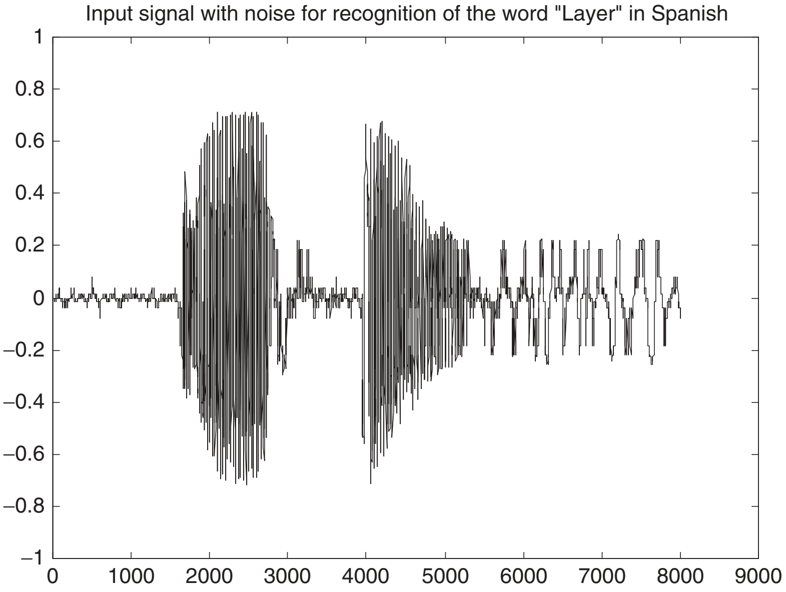


Fig. 11.9. Input signal of the word "layer" in Spanish with noise added

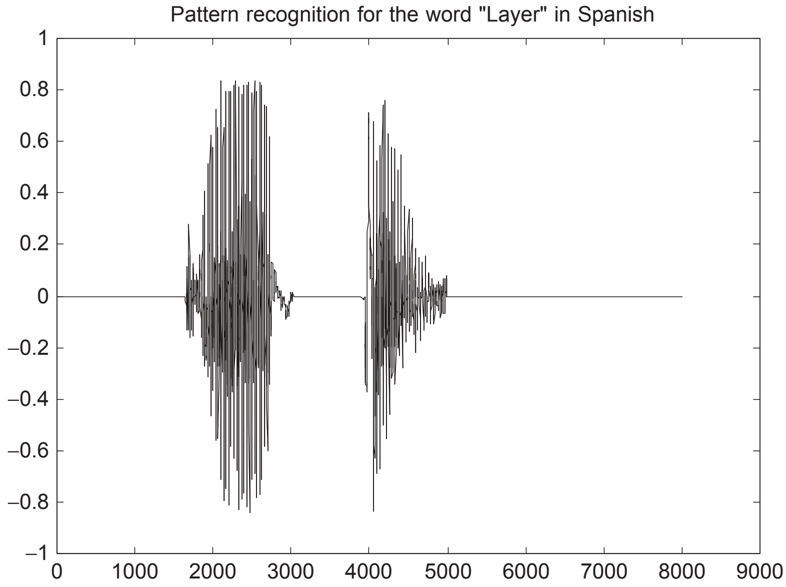


Fig. 11.10. Identification of the word “layer”

11.5 Voice Recognition with Modular Neural Networks and Type-2 Fuzzy Logic

We can improve on the results obtained in the previous section by using modular neural networks because modularity enables us to divide the problem of recognition in simpler sub-problems, which can be more easily solved. We also use type-2 fuzzy logic to model the uncertainty in the results given by the neural networks from the same training data. We describe in this section our modular neural network approach with the use of type-2 fuzzy logic in the integration of results.

We now show some examples to illustrate the hybrid approach. We use two modules with one neural network each in this modular architecture. Each module is trained with the same data, but results are somewhat different due to the uncertainty involved in the learning process. In all cases, we use neural networks with one hidden layer of 50 nodes and “trainrp” as learning algorithm. The difference in the results is then used to create a type-2 interval fuzzy set that represents the uncertainty in the classification of the word. The first example is of the word “example” in Spanish, which is shown in Fig. 11.11.

Considering for now only 10 words in the training, we have that the first neural network will give the following results:

SSE = 4.17649e-005 (Sum of squared errors)
Output = [0.0023, 0.0001, 0.0000, 0.0020, 0.0113, 0.0053, 0.0065, 0.9901, 0.0007, 0.0001]

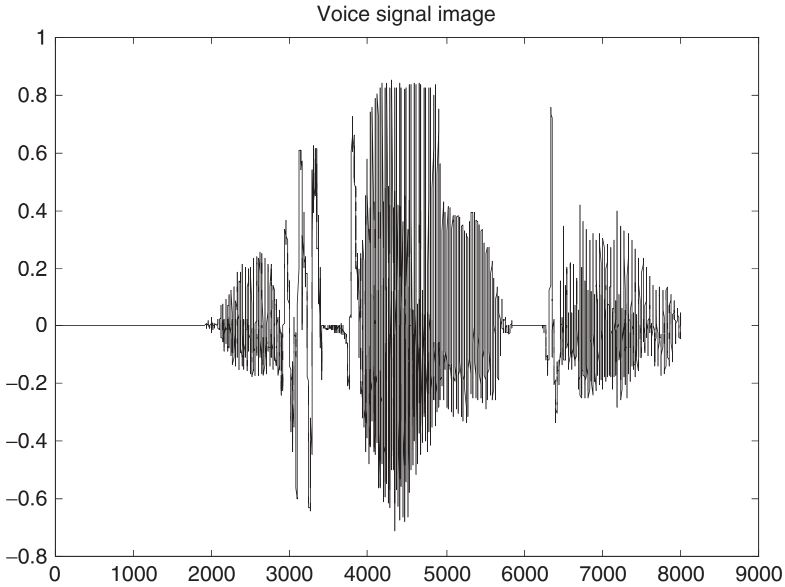


Fig. 11.11. Sound signal of the word "example" in Spanish

The output can be interpreted as giving us the membership values of the given sound signal to each of the 10 different words in the database. In this case, we can appreciate that the value of 0.9901 is the membership value to the word "example", which is very close to 1. But, if we now train a second neural network with the same architecture, due to the different random inicialization of the weights, the results will be different. We now give the results for the second neural network:

SSE = 0.0124899
Output = [0.0002, 0.0041, 0.0037, 0.0013, 0.0091, 0.0009, 0.0004, 0.9821, 0.0007, 0.0007]

We can note that now the membership value to the word "example" is of 0.9821. With the two different values of membership, we can define an interval [0.9821, 0.9901], which gives us the uncertainty in membership of the input signal belonging to the word "example" in the database. We have to use centroid deffuzification to obtain a single membership value. If we now repeat the same procedure for the whole database, we obtain the results shown in Table 11.3. In this table, we can see the results for a sample of 6 different words.

The same modular neural network approach was extended to the previous 20 words (mentioned in the previous section) and the recognition rate was improved to 100%, which shows the advantage of modularity and also the

Table 11.3. Summary of results for the two modules (M1 and M2) for a set of words in "Spanish"

Example		Daisy		Way	
M1	M2	M1	M2	M1	M2
0.0023	0.0002	0.0009	0.0124	0.0081	0.0000
0.0001	0.0041	0.9957	0.9528	0.0047	0.0240
0.0000	0.0037	0.0001	0.1141	0.0089	0.0003
0.0020	0.0013	0.0080	0.0352	0.9797	0.9397
0.0113	0.0091	0.0005	0.0014	0.0000	0.0126
0.0053	0.0009	0.0035	0.0000	0.0074	0.0002
0.0065	0.0004	0.0011	0.0001	0.0183	0.0000
0.9901	0.9821	0.0000	0.0021	0.0001	0.0069
0.0007	0.0007	0.0049	0.0012	0.0004	0.0010
0.0001	0.0007	0.0132	0.0448	0.0338	0.0007
Salina		Bed		Layer	
M1	M2	M1	M2	M1	M2
0.9894	0.9780	0.0028	0.0014	0.0009	0.0858
0.0031	0.0002	0.0104	0.0012	0.0032	0.0032
0.0019	0.0046	0.9949	0.9259	0.0000	0.0005
0.0024	0.0007	0.0221	0.0043	0.0001	0.0104
0.0001	0.0017	0.0003	0.0025	0.9820	0.9241
0.0000	0.0017	0.0003	0.0002	0.0017	0.0031
0.0006	0.0000	0.0032	0.0002	0.0070	0.0031
0.0001	0.0024	0.0003	0.0004	0.0132	0.0000
0.0067	0.0051	0.0094	0.0013	0.0003	0.0017
0.0040	0.0012	0.0051	0.0001	0.0010	0.0019

utilization of type-2 fuzzy logic. We also have to say that computation time was also reduced slightly due to the use of modularity.

We now describe the complete modular neural network architecture (Fig. 11.12) for voice recognition in which we now use three neural networks in each module. Also, each module only processes a part of the word, which is divided in three parts one for each module.

We have to say that the architecture shown in Fig. 11.12 is very similar to the ones shown in previous chapter for face and fingerprint recognition, but now the input is a voice sound signal. This signal is then divided in three parts to take advantage of the modularity, but to improve accuracy we use several simple neural networks in each module. At the end, the results of the three modules are integrated to give the final decision.

We have also experimented with using a genetic algorithm for optimizing the number of layers and nodes of the neural networks of the modules with very good results. The approach is very similar to the one described in the

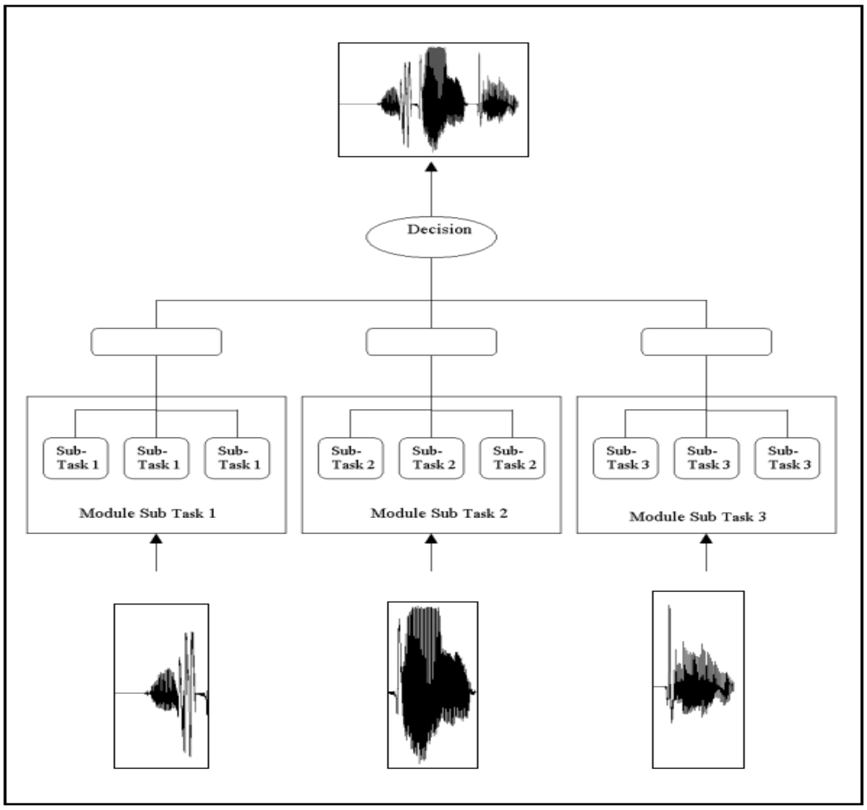


Fig. 11.12. Complete modular neural network architecture for voice recognition

previous chapter. We show in Fig. 11.13 an example of the use of a genetic algorithm for optimizing the number of layers and nodes of one of the neural networks in the modular architecture. In this figure we can appreciate the minimization of the fitness function, which takes into account two objectives: sum of squared errors and the complexity of the neural network.

11.6 Summary

We have described in this chapter an intelligent approach for pattern recognition for the case of speaker identification. We first described the use of monolithic neural networks for voice recognition. We then described a modular neural network approach with type-2 fuzzy logic. We have shown examples for words in Spanish in which a correct identification was achieved. We have performed tests with about 20 different words in Spanish, which were spoken by three different speakers. The results are very good for the monolithic

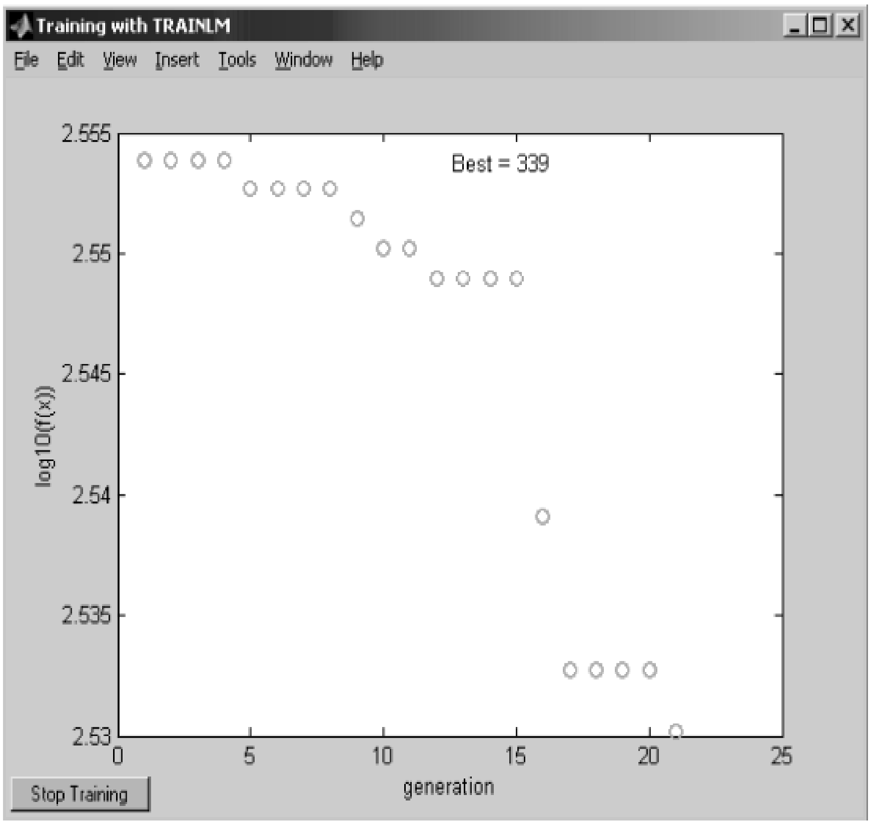


Fig. 11.13. Genetic algorithm showing the optimization of a neural network

neural network approach, and excellent for the modular neural network approach. We have considered increasing the database of words, and with the modular approach we have been able to achieve about 96% recognition rate on over 100 words. We still have to make more tests with different words and levels of noise.