ISCA Archive
http://www.isca-speech.org/archive

6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

GeNeSys: A NEURAL NETWORK MODEL FOR SPEAKER IDENTIFICATION

B.Ruiz-Mezcua(*), R.Rodríguez-Galán(*), Luis A. Hernandez-Gómez (**), Paloma Domingo-García(*) and Enrique Bailly-Baillière Gutiérrez(*)

(*)IRIS: Laboratory of Systems Integration, Computer Science Department,
Universidad Carlos III, c/ Butarque, 15 - 28911 Leganés, Madrid (SPAIN).
Phone: (34-1) 624 91 04. Fax: (34-1) 624 94 30. Email: bruiz@inf.uc3m.es
(**) Signals Systems and Radiocommunications Department.
ETSI Telecomunicación. Universidad Politecnica de Madrid.

10.21437/Eurospeech.1999-246

ABSTRACT

Mathematical models have been extensively used to shape living organism behaviour. These models are based on the N-dimensional space classification for those in which the patterns may have been defined.
GeNeSys neural network family has been postulated as a global, comprehensive solution that shapes an individual behaviour. This article describes the GeNeSys family and presents some theoretical results of the researches in speaker recognition.

An identification/verification system voice based is proposed. This implementation can identify or verify a speaker from 30 speakers contained in a multisession database. In this paper, a speaker verification system is presented and the tasks related to the speaker verification through the speech are developed. This system is applied to multimedia database access, services and applications. To achieve this goal a previous learning process is necessary. After the training phase is finished, the speaker model is calculated and stored in a database. A speaker recognition task using the database M2VTS from ElRA is about 88%.

Keywords: Speaker Verification, Speaker Identification, Adaptive control, ART, real-time applications, decision theory.

1. INTRODUCTION

The use of intelligent systems to control mechanical devices has been extensively tested. However, the elaboration of such systems requires specific knowledge on the environment where the system is going to be placed, a detailed description of the actions to be carried out and the specification of the relationship between these actions and the events in the world. This has been the dominant view in Artificial Intelligence. It can be described as the physical symbol hypothesis: in order to give system processing capabilities, it is necessary to implement a symbolic world model. It implies that the system's designer has a prior knowledge about the problem and problem solving methods. The system will use this knowledge, implemented as a symbol-relation system, to reach its goals. In response to the outlined restrictions we have developed GeNeSys, using the concept of competitive learning mechanisms, where the neurons compete mutually and, based on some criteria given, the winner categorises the input pattern. The result of this work is a generic neural network architecture adapted to problems of pattern categorisation.

In the speaker verification or identification system (speaker recognition) the discrimination task is one of the most arduous problem in the system scheme. In this sense it is very difficult to find a system that would be able to identify a speaker in a definitive way. In speaker recognising systems, the achievement of a discriminative scheme capable of learning the speaker's characteristics is one of the main goals in the system's architecture design. This scheme should identify the speaker in such a way that the system could distinguish him from the rest. The first difficulty happens to be in the establishment of a stable threshold, able to decide if the speaker's characteristics are similar enough to those of the desired user, and different enough to those of the rest of speakers. Therefore, GeNeSys is an efficient tool due to its strong capability of discrimination. Another intrinsic characteristic of GeNeSys is its learning skills, which absorbs changes in voice that speakers suffer along time.

The next dilemma is choosing the characteristics that are going to be used to feed such tool. Regarding system applications activated by voice there are two principal problems: first, voice is a time variant system; second, speaker's characteristics change due to physical and emotional reasons, the environment and pass of time. In text-dependent systems the message established in model generation (system training) is the same as in their own test. The choice has been a text-dependent application. The autocovariance matrices, obtained from a sequence of spectral characteristics vectors, have been selected out of the experiments carried out.

2. ACOUSTIC ANALISYS

In order to obtain a features vector useful to GeNeSys pre-processing, it is necessary that this vectors have the characteristics that the neural network needs. These characteristics are:

- Time stability
- Speaker representation

- Speakers' discrimination.

In general, the steps to obtain a set that identifies in a unique way the speaker's features and distinguishes from the rest are:

- Voice signal spectrum. The spectrum is defined as the spectral envelope, described by the cepstral coefficients (defined in the Mel-scale in presence of noise as seen further on). It is assumed that the description given by the first ten cepstral coefficients in the frequency series development has an acceptable error.
- Spectrum evolution. As mentioned, the voice signal is variable with time, thus, it is important to know how the spectrum evolution defined by the cepstral coefficients is described, calculating the differential cepstra in an analysis window width of three. In addition, the spectral evolution is defined with the first ten differential cepstra.
- Energy and incremental energy of the voice signal.

Once the vectors of features sequence for the analysis frame have been obtained, as in figure 1, they are useful for training the generation of the autocovariance matrices. These matrices have been chosen because they represent in a unique way each one of the speakers stored in the data base, for a training sequence given.

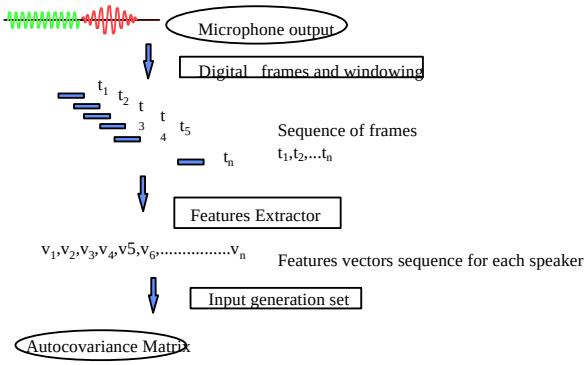

Figure 1: Autocovariance matrix generation

3. A BASIC OVERVIEW OF GeNeSys

Next, we are going to detail the equations that define the neural model GeNeSys. We consider N as the number of nodes in Input Layer I, and M as the number of nodes in the Output Layer O.

1. Model Initialisation: The parametrization of the constants is made according to the following restrictions:

   a,b > 0; $0 \le c,d \le 1$; e << 1;
   $K \ge 1$; $0 \le \rho \le 1$; $0 \le \theta \le 1$;

The rising and dropping weights receive initial values different from zero:

$$w_{ji}(0) = 1 \qquad w_{ij}(0) < \frac{1}{\sqrt{N}}$$

The model is initialised with a null vector, and the cycle counter is initialised to zero.

2. The counter value is incremented by 1. Six processes are defined, p_i, and they are applied to the input pattern, P:

   - Process 1 (p_1): Addition of the input pattern P to p_4 expanded by the constant *a*.

   $$p_{1i} = P_i + a p_{4i}$$

   - Process 2 (p_2): Normalisation of p_1, adjusted by the normalisation constant *e*.

   $$p_{2i} = \frac{p_{1i}}{e + \|p_1\|}$$

   - Process 3 (p_3): Functional addition of p_2 and p_6 expanded by the constant *b*.

   $$p_{3i} = f(p_{2i}) + bf(p_{6i})$$

   where the shape of the function *f(x)* determines the improvement of contrast in I. The logical choice for this function could be a sigmoid, but the most elementary option is the step function:

   $$f(x) = \begin{cases} 0 & 0 \le x \le \theta \\ x & x > \theta \end{cases}$$

   where θ is a positive constant less or equal to 1. Another possibility is this sigmoidal function:

   $$f(x) = \begin{cases} \dfrac{2\theta^2}{(x^2 + \theta^2)} & 0 \le x \le \theta \\ x & x > \theta \end{cases}$$

   - Process 4 (p_4): Normalisation of p_3, adjusted to the normalisation constant *e*.

   $$p_{4i} = \frac{p_{3i}}{e + \|p_3\|}$$

   - Process 5 (p_5): Functional addition of p_4 and functional expansion of the pattern winner of the layer O.

   $$p_{5i} = \begin{cases} p_{4i} + \sum_j g(y_j) w_{Ji} & \text{if } O \text{ is active} \\ p_{4i} & \text{if } O \text{ is inactive} \end{cases}$$

   - Process 6 (p_6): Normalisation of p_5, adjusted to the normalisation constant *e*.

   $$p_{6i} = \frac{p_{5i}}{e + \|p_5\|}$$

3. The values of the process p_4 propagate to layer O. The inputs of O are calculated, denominated I_O.

$$I_{Oj} = \sum_{i=1}^{M} p_{4i} w_{ij}$$

These allows a measure of likelihood between the Input Pattern P and the classes of layer O, without reaching such classes.

4. Only one node of O has a non-zero output, called winner node. This node is defined by the function:

$$g(y_J) = \begin{cases} dI_{OJ} = \max_k \{I_{Ok}\} & \forall k \\ 0 & In \quad any \quad other \quad case \end{cases}$$

5. If the counter value is set to 1, the previous steps are repeated in order to give values to all the crossed processes of the layer I.
6. Calculate the output of layer I, called O_I, according to the following normalised resemblance functions:
   - Euclidean: $O_{Ii} = \frac{\sqrt{\sum_i (P_i - w_{Ji})^2}}{N}$
   - Manhattan: $O_{Ii} = \frac{\sum_i |P_i - w_{Ji}|}{N}$
   - Hamming: $O_{Ii} = \frac{\sum_i \frac{P_i w_{Ji}}{max\{P_i\} - min\{P_i\}}}{\sum_i w_{Ji}^2}$

Considering the input pattern we pretend to categorise. We will select the function depending on the well known characteristics of each of them.

7. Without reaching the class previously learnt, it will be determined weather a learning process will take place or not. If the *surveillance factor*, ρ / (e + O_I) > 1, is not exceeded, a restoring signal is sent to O and all of the possible active nodes of O are marked as not valid. The cycle counter is set to one, and turn back to step 2. If there is no restoring and the cycle counter is set to 1, the counter is incremented and you pass to step 1. If there is no restoring, the next step follows.
8. The rising weights of the winning entity of O are modified.

$$w_{iJ} = \frac{p_{4i}}{(1-d)}$$

9. The descending weights of the winning entity of O are modified.

$$w_{Ji} = \begin{cases} P_i(t) & J \ new \ category \\ \frac{K P_i(t) + w_{Ji}(t)}{K+1} & J \ new \ category \end{cases}$$

10. The input pattern, P; is eliminated. All of the inactive entities of O are restored. You get back to step 2 with a new input pattern P.

4. DESCRIPTION OF THE EXPERIMENT

Once the classification scheme and the parameters that will be used in its input have been selected, it is necessary to establish the steps to be followed in order to implement the system:

- First of all, a previous training stage, in which the models associated to each speaker will be generated, will be carried out, admitting a reference base that will contain the models generated with the data presented in the training stage.
- After generating the models, we go on with the recognition itself. In which the voice features of the user that tries to enter the system, are compared to the existing models in the reference base or data base.

The sequence of steps necessary to generate the models, applicable to both technologies, is resumed in figure 2. The first two boxes correspond to the pre-process of the voice signal. This pre-process is carried out with the features extraction and the covariance matrices. In the models based on GeNeSys there have been several attempts to introduce some pre-models in the net. The most efficient are the ones based on measures of autocovariance matrices of features' vectors (developed with spectral measure elements and their speed, as well as the energy measure and its differential) obtained from speech. Up to now, these are the characteristics that turn to be most interesting as pre-models to train the neural network and categorise the patterns obtained from each speaker.

Figure 2: Training Task. Database Generation

In the recognition phase, as in figure 3, the models are compared with user's voice features, and, depending on a likelihood measure previously fixed, the user will be accepted or rejected in the verification mode.

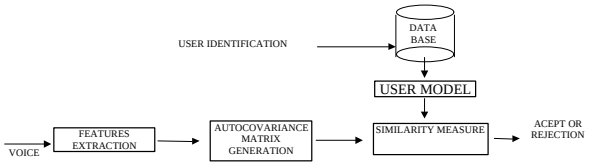

Figure 3: Speaker verification Task

In order to achieve the best set-up configuration able to obtain some models stables in the time is necessary to train the system using speech recording in different sessions. In this way the changes along time of the speech characteristics for each user are learnt by the system improving the its final performance. To develop this task is necessary to have a multi-session database to train and test the system, after that the training phase and recognition phases are accomplished.

The database used to attain this training and test a database recorded in the Carlos III University is used. This database has the following features:

- 40 speakers: male and females.

- Office environment recorded with an oriented microphone.
- Speech and digits recorded in 12 different sessions separated in media one week each one.
- Several items has been recorded in each session: name, address date and place of birth, digits from zero to 9 spoken several times, identification number and a free text different for each session.

In the training phase the recorded speech is isolated from noise, filtered and then sampled at 8 kHz. The signal is pre-processed and the spectral features have been extracted. After that the autocovariance matrix has been obtained from the features vector. The network GeNeSys is initialised with 30 input patterns (one autocovariance matrix by speaker). And, in order to obtain one category per speaker the vigilance factor is adjusted to 100% of similarity.

The training implementation is accomplished in two steps:

- Firstly, the autocovariance matrix feeds GeNeSys and the models are stored in the descendent weights. The matrices are obtained from the five first sessions. The matrices are introduced to GeNeSys and, when the surveillance factor is fixed to 100% the model corresponding to the speaker introduced is stored in the descendent weights. This task is performed for each speaker in a separately way.
- Second, when the pre-models are stored in the descendent weights all pre-models are introduced together and the surveillance factor is fixed to 95%. In the rising weights the discrimination factors are stored for each speaker, taking into account the speaker introduced. It is necessary to perform this phase when a new speaker is added to the database, and will also be necessary to perform only the first step for the new speaker.
- In the recognition phase the autocovariance matrix of test users is obtained. When the identity of speaker is claimed, the system compares the input patterns with the stored categories in the network. The distance between both is measured and the user is accepted or rejected depending on the threshold value (ρ).

The best results have been obtained in the recognition phase when the surveillance factor is equal to 99%. In this case the FR is equal to 0% and the FA is equal to 12%.

5. SUMMARY

GeNeSys is a new technology that stands out as a very useful tool in the speech speaker classification task, obtaining on adverse conditions a good system performance. In this sense, as has been shown in the previous paragraph, the speaker recognition rate is equal to 88%, the FR is equal to 0% and the FA equal to 12% using the M2VTS database contains in ELRA.

GeNeSys is a very simple classifier and could be implemented in a PC-486 environment with a SoundBlaster card able to record the speech that will be categorised.

It is necessary only one second of speech to recognise and about 15 second to train the system.

The results will be improved using a features vector more adequate to speech characteristics, and using more speech to train the system.

6. REFERENCES

[1] Cáceres-Alonso, P., Rodríguez-Galán, R., and García-Tejedor. A., *Non-Supervised Neural Categorisation of Near-Infrared Spectra. Proceeding of NIR-95*. 7th International Conference on Near-Infrared Spectroscopy. Montreal, Canada. August 6-11, 1995.

[2] Escrihuela, Gerboles y Ruiz (89). *Algoritmica del sistema de reconocimiento de grandes vocabularios.* Documento interno de Alcatel.

[3] Furui & Sondhi *Advances in Speech Signal Processing*. Ed. Marcel Dekker, Inc. 1989.

[4] Lobo, S., García-Tejedor, A.J., Rodríguez-Galán, R., López, L. and García-Crespo, A., *A Simplification of the Theory of Neural Groups Selection for Adaptive Control*. Proceeding de ECAL'95. Granada, Junio 1995.

[5] Rodríguez-Galán, R. and García-Tejedor, A., *ART: an implementation of the new direct access condition*. Proceeding de la International Neural Network Conference. Paris, 1990.

[6] Rodríguez-Galán, R. *Modificaciones al Mecanismo de Aprendizaje de Modelos Neuronales No Supervisados basados en la Teoría de Resonancia Adaptativa. Aplicación al Reconocimiento de Patrones Complejos en Entornos de Producción*. Tesis Doctoral. Dpto. de Matemática Aplicada a la Tecnología de la Información. E.T.S.I.T. Univ. Politécnica de Madrid, 1995.

[7] Ruiz-Mezcua, Gerbolés-Espina, Escrihuela-Langa, Gomez Mena, Veiga. (92) *Reconocimiento de grandes vocabularios independiente del locutor*. URSI92.

[8] Ruiz-Mezcua, Hernadez, Domingo, Rodriguez. *Acceso a servicios multimedia a traves de la voz*. URSI96 Conference.

[9] Ruiz-Mezcua, B. *Modelado estadístico y conexionista para reconocimiento de locutores con aprecndizaje de la variabilidad temporal del Habla.* Ph.D. E.T.S.I.T. – UPM, 1998.