PERFORMANCE OF A PHONETIC ENCODING SCHEME FOR SPEECH RECOGNITION USING NEURAL NETWORKS

V. Rodellar, P. Gómez, C. García, F. Naharro, M. Pérez, y C. Gonzalo

Departamento de Arquitectura y Tecnología de Sistemas Informáticos
Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo, s/n
Boadilla del Monte
28660 Madrid
Tfno: +34.1.336.73.84
Fax: +34.1.336.74.12
E-mail: pedro@pino.datsi.fi.upm.es

Abstract

Through the present work, an Encoding Scheme for the Identification of the Phonetic Features of Speech, introduced and developed in a preliminary research [1, 2], is implemented on a Time-Delay Back-Propagation Neural Network (TDBPNN), and its most relevant features are analized. The Encoding Scheme is based on an 8-bit Hamming code, and can be represented by an 8-dimensional hypercube. A separate Phonetic Subgraph, when the relevant nodes and edges are considered, is presented, showing the Minimum Distance Pairs for Spanish. The problem of using a Time Delay Neural Network to support this Encoding Scheme is addressed, pointing to the methods train the Network. For such, a Fragmentation and Labeling Technique based in the PARCORgram Correlation Matrix of Speech is presented. Results show that using a 48:9:8 Back Propagation Time-Delay Network with fragments of Speech of the kind VCV or CV from the densest subset of sounds in the Phonetic Subgraph, the percentage of failures produced during the recognition process may be substantially reduced. The technique shown may be used in Computer Aided Speech Learning.

Key words:

Speech Recognition, Neural Nets, Phonetic Coding, Spectral Estimation, Correlation

Introduction

The present paper describes an Encoding Scheme for the Phonetic Features of Spanish, which may be usefully employed in the automatic determination of the phonetic features of a given segment of Speech. The Encoding Scheme is based on an 8-bit Hamming Distance Code related to the Phonetic Features of Spanish as shown in Table 1, and may be expressed by an 8-dimensional hypercube as a fully connected graph [3], in which a given node is associated to a given sound in the Encoding Scheme.

Bit #	Feature	Comment
0	Oral/Nasal	b0 =1 oral; b0=0 nasal
1	Voiced/Unv.	b1=1 voiced; b1=0 unvoiced
2 3	Degree of closure	b2=0, b3=0; plosive b2=0, b3=1; fricative b2=1, b3=0; vowel b2=1, b3=1; semicons., glide
4 5	Articulatory place	b4=0, b5=0; palatal b4=0, b5=1; dental b4=1, b5=0; velar b4=1, b5=1; labial
6	In vowels: Rnd/Oval In consonants: Frontal/Lateral	b6=1 rounded; b6=0 oval b6=1 frontal; b6=0 lateral
7	Multiple/Simple	b7=1 multiple; b7=0 simple

Table 1. Feature Encoding Scheme

Not every node in the related graph is attached to a sound, most of them being empty. The subgraph obtained removing the empty nodes and edges may be seen in Fig. 1.

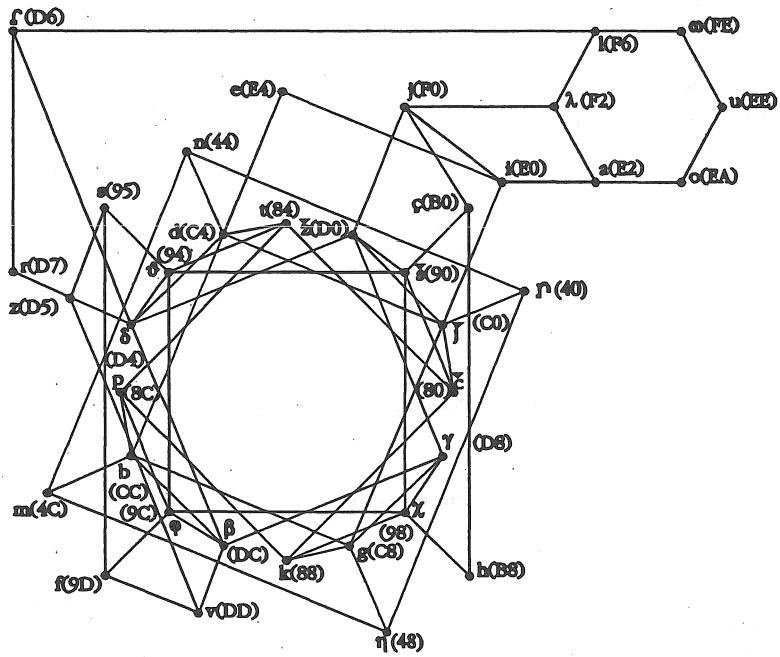

Fig. 1. Subgraph corresponding to the Encoding Scheme being proposed.

A given phoneme (shown in boldface), and its corresponding hexadecimal code, appear close to the corresponding node. The hexadecimal code is introduced as an abreviation of the related binary code, with bit #0 and bit #7 respectively taken as the most and least significant ones. In this subgraph, the edges relate sounds which may be considered familiar to others in the sense that they form a Minimum Hamming Distance Pair. This fact can be checked with sounds /t/ and /d/. Their corresponding codes are the hexadecimal numbers (H84) and (HC4). This implies a one-bit difference between them, corresponding to the bit coding the Phonetic Feature of Voicing (bit #1).

Phonetic Encoding by Neural Networks

This Encoding Scheme is specially devised to be supported by a Time-Delay Back-Propagation Neural network (TDBPNN) [4], which assigns an 8-bit code to each fragment of LPC vectorized Speech [1]. LPC extraction may be done using Blind Overlapping Fragmentation (BOF), or Phonetic Sensitive Fragmentation (PSF). Blind Fragmentation produces equal-size fragments, although is not careful with the changes in the spectral characteristics of the signal. This last fact may be attenuated using overlaping fragments. PSF is very careful with the spectral characteristics of Speech, produces a more compact set of LPC vectors, and may compare to Viterbi's Algorithms [5]. A typical BOF spectrogram obtained applying LPC extraction and spectral reconstruction may be seen in Fig. 2 for the utterance /aγa/.

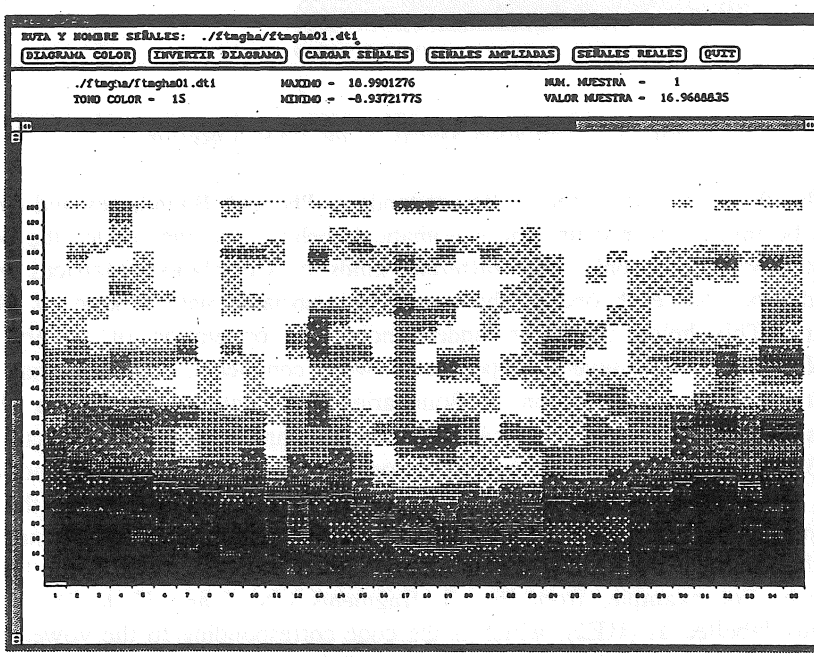

Fig. 2. Spectrogram obtained by Blind Overlaping Fragmentation of /aγa/.

Speech Labeling is a very complicated, sensitive and error-prone task, in the sense that an Expert Phonetician has to assign a tentative code to every LPC vector, which in many cases don't show clearly established Phonetic Features. To help the Expert Phonetician in this task, the present research introduces a new technique, based in using the Correlation Matrices of the PARCORgram, considered as the Time-Ordered Set of LPC vectors associated with a Speech fragment. A typical PARCORgram Correlation Matrix may be seen in Fig. 3.

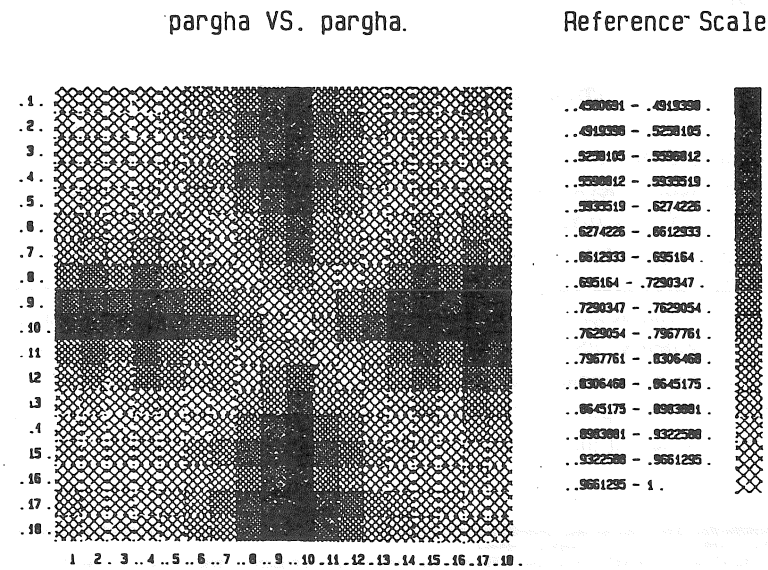

Fig. 3. Correlation Matrix of the PARCORgram.

It may be shown that only four different kinds of Phonetic Boundaries and Contours have to be taken into account for Automatic Labeling, and that under the point of view of Correlation, only four different kinds of Correlation Matrices may be expected, and that each one of them may be associated with a kind of Phonetic Boundary. This helps greatly in understanding the opperation of a very simple TDBPNN, for which several experiments were conducted, showing that in the simplest case, the identification of Boundaries and Contours by such a Network requires the use of a minimal 3:4:2 architecture, and demonstrating that Single-Layered Neural Networks, such as Perceptrons or ADALINES [6, 7] are not able of solving the problem, which can be considered as a generalization of the XOR-Problem. The study of the Correlation Matrices gives the hints to the Phonetician of where the Phonetic Boundaries should be established, to correctly label the fragments of Speech. For example, from Fig. 3, fragments 1 to 7 and 13 to 18 should be apparently labelled as (HE2), which is the code corresponding to the vowel /a/, and fragments from 8 to 12 should be labelled as (HD8), which is the code assigned for /γ/. A special Network may be devised for this purpose, relieving the Phonetician of such a cumbersome task. The second part of the paper is devoted to describe the behaviour of an Encoding TDBPNN, showing that an architecture of 48:9:8 as the

one in Fig. 4, and a Correlation-Matrix-Based Labeling, produce important improvements over preliminary approaches [2].

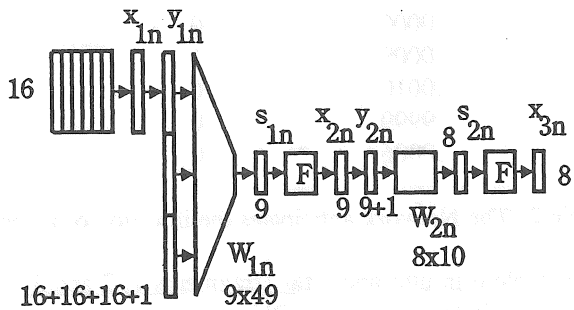

Fig. 4. Architecture of the TDBPNN being used

In Fig. 4, W are the weight matrices, x are the normalized inputs to each layer, and s are the unnormalized outputs. F is the normalizing nonlinear function. The dimensions of the matrices and vectors is stated in the lower part of the figure.

Results

The performance of such structure is checked using a core of 16 "cardinal" consonants, from the densest part of the Encoding Graph shown in Fig. 1, plus a vowel, in groups with structure VCV or CV. The use of the cardinal set of consonants, which is in itself a 4-dimensional hypercube, allows to test the performance of the Network to differentiate between Minimum Distance Pairs, in the sense of Hamming Distance, and as such allow a very strict check of the behaviour of the Network under the worst conditions. According to these considerations, a set of experiments was conducted using the cardinal set formed by the consonants /p/t/c/k/b/d/j/g/φ/ϑ/s/χ/ coarticulated with the vowel /a/, in structures of the kind CV, and the consonants /β/δ/z/γ/ coarticulated with the vowel /a/ in structures of the kind VCV. Each structure was fragmented using PSF techniques into 10 to 20 fragments, and LPC extracted. Each 16-element LPC vector was assigned an 8 bit Phonetic Code. With three consecutive of these groups a sample block to be fed to the TDNN was built. Thus, a total number of 224 of these samples were used during the training process. The 48:9:8 TDBPNN was trained using this set of samples during 935 training steps. When the process was completed, the set of samples associated to a given sound, was presented to the Network for its recognition, and the Network's response was compared to its corresponding Phonetic Label. Then, it was found that only 5 bits were erroneously coded, out of the total amount of 3584 checked, which yielded a percentage of 97,7% Phonemes correctly encoded. Besides, most of the observed mistakes were not fatal, in the sense that three of them appeared in the boundary between consonant and vowel, as exposed in Table 2. The Network "anticipated" the insertion of the vowel in disagreement with its preassigned label.

Sample 4 in utterance /ka/ (fragments 5, 6 and 7):

Bit #	Output value	Objective value	Agreement
0	.9994	1	Yes
1	.9957	0	No
2	.9999	0	No
3	.0000	0	Yes
4	.0009	1	No
5	.0010	0	Yes
6	.9999	0	No
7	.0006	0	Yes

Table 2. The Network anticipates the insertion of a vowel.

Sample 0 in utterance /ta/ (fragments 1, 2 and 3):

Bit #	Output value	Objective value	Agreement
0	.9997	1	Yes
1	.0244	0	Yes
2	.0009	0	Yes
3	.0000	0	Yes
4	.0249	0	Yes
5	.0000	1	No
6	.0008	0	Yes
7	.0002	0	Yes

Sample 1 in utterance /ta/ (fragments 2, 3 and 4):

Bit #	Output value	Objective value	Agreement
0	.9997	1	Yes
1	.0213	0	Yes
2	.0107	0	Yes
3	.0000	0	Yes
4	.0206	0	Yes
5	.0000	1	No
6	.0107	0	Yes
7	.0002	0	Yes

Table 3. Failures present in the detection of /ta/ coded as /ca/

In Table 2, an utterance of /ka/, divided into 13 samples, each one containing 3 time-consecutive fragments according to the suggestion of the PARCORgram of the Correlation Matrix, was being recognized. The Network's response for each bit in the Phonetic Code was listed under **Output value**, and the desired response (objective) was listed under **Objective value**. The response for the first four samples, from 0 to 3, not presented in the Table, showed a complete agreement between the desired response (**H88**), corresponding to /k/, and the actual response. For sample no. 4, the actual and desired responses were listed in Table 2, showing that there was a disagreement in four of the code bits, corresponding to numbers 1, 2, 4 and 6. The response produced by the Network was (**HE2**), this last code corresponding to /a/. Having into account that samples 5 to 12 were associated to that last sound, it seems

that the apparent failure was due to an anticipation in the detection of the boundary between consonant and vowel, rather than to a malfunctioning. In Table 3, the responses to the first two samples out of a set of 10, are shown. It can be seen that the actual responses yield code (H80) corresponding to the consonantal sound /c/ when (H84) corresponding to /t/, was expected. These mistakes could be produced by an excess of aspiration when uttering the original consonant, and in this case the Network's suggestion should be taken literally for the reconsideration of the assignment done during the labeling process. Finally, Table 4 shows the behaviour of the network in a typical case of error-free processing, involving the recognition of the utterance /aɣa/. It can be seen, that the detection of the vowel and the approximant, are quite complete, even in the boundaries between them, as is the case with samples 10 and 11, which share the spectral information of fragments 12 and 13.

Sample 4 in utterance /aɣa/ (fragments 5, 6 and 7):

Bit #	Output value	Objective value	Agreement
0	.9995	1	Yes
1	.9998	1	Yes
2	.9998	1	Yes
3	.0000	0	Yes
4	.0012	0	Yes
5	.0000	0	Yes
6	.9998	1	Yes
7	.0004	0	Yes

Sample 10 in utterance /aɣa/ (fragments 11, 12 and 13):

Bit #	Output value	Objective value	Agreement
0	.9977	1	Yes
1	.9898	1	Yes
2	.0009	0	Yes
3	.9987	1	Yes
4	.9997	1	Yes
5	.0077	0	Yes
6	.0009	0	Yes
7	.0019	0	Yes

Sample 11 in utterance /aɣa/ (fragments 12, 13 and 14):

Bit #	Output value	Objective value	Agreement
0	.9997	1	Yes
1	.9999	1	Yes
2	.9938	1	Yes
3	.0001	0	Yes
4	.0015	0	Yes
5	.0000	0	Yes
6	.9938	1	Yes
7	.0002	0	Yes

Table 4. Three typical fragments being properly identified.

The results rendered by the present approach show a great accuracy in the task of Phonetic Feature Encoding, and allow us to conclude that the research in progress is highly interesting because may be efficiently applied to Speech Processing and Recognition. Other approaches incorporating more sophisticated Encoding Schemes, and more powerful Network Architectures are also being considered. The applications of the present research are oriented to Computer Aided Speech Training, such as in Language Disorder Care, or in Language Schools.

Acknowledgements

The present research is being developed under Grants Nos. C072/90 from the Plan Regional de Investigación de la Comunidad de Madrid and Acción Concertada UPM A91-0020-02-78.

References

[1] V. Rodellar, P. Gómez, M. Hermida, A. Díaz and R. W. Newcomb, "A VLSI Architecture for the Support of an Auditory Model for Hearing and Speech Processing", **Proc. of the 33rd Midwest Symposium on Circuits and Systems**, Calgary, Alberta, Canada, August 12-15, 1990, pp. 787-790

[2] V. Rodellar, F. Naharro, C. García, S. Martín, M. L. Muñoz and P. Gómez, "A Neural Network for the Extraction and Characterization of the Phonetic Features of Speech", **Proc. of the Fourth Int. Conf. on Neural Networks and Applications, NEURO-NIMES'91**, Nimes, France, November 4-8, 1991, pp. 203-212

[3] A. Gibbons, **Algorithmic Graph Theory**, Cambridge University Press, Cambridge, England, 1985

[4] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano and K. J. Lang, "Phoneme Recognition using Time-Delay Neural Networks", **IEEE Trans. on ASSP**, Vol. 37, No. 3, March 1989, pp. 328-339

[5] T. Parsons, **Voice and Speech Processing**, McGraw-Hill, New York, 1987

[6] T. Khanna, **Foundations of Neural Networks**, Addison-Wesley Pub. Co., Reading, Massachusetts, 1990

[7] B. Widrow and M. A. Lehr, "30 Years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation", **Proc. of the IEEE**, Vol. 78, No. 9, Sept. 1990, pp. 1415-1442