THE APPLICATION OF A FUZZY CLUSTERING NEURAL NETWORK TO ACOUSTIC-PHONETIC MAPPING FOR ASR

Attilio Erriquez, Rosa Caradonna & Jacques Koreman

Institute of Phonetics, University of the Saarland, Saarbrücken, Germany
{erriquez, roca, koreman}@coli.uni-sb.de, http://coli.uni-sb.de/~erriquez

Abstract

Most state-of-the-art automatic speech recognition (ASR) systems use a vector of acoustic (spectral) parameters to recognise phones. The acoustic parameters reflect linguistic properties encoded in the signal. As shown in Koreman et al. (1999), mapping acoustic parameters onto phonetic features helps to extract these linguistic properties and leads to an increase in phone identification rate. In the experiment presented here, the Kohonen map used in Koreman et al. (1999) is replaced by a fuzzy neural network sensitive to density (FNNSD).

This network has been applied successfully for clustering colour shades (Acciani et al., 1999), which are non-uniformly distributed data with overlaps. Overlaps between different phones typically occur in natural speech data, so that the fuzzy rules to compute cluster membership used in the FNNSD should be applicable to ASR. For each input, this kind of network attributes a continuous degree of membership of all phones. In this article we shall compare results of phoneme identification using a FNNSD with those using a Kohonen map.

1. Introduction

In this paper we present phone identification results for a hybrid automatic speech recognition (ASR) system. The system uses a neural network to map vectors of acoustic parameters onto phonetic feature vectors, which are then used as input for hidden Markov modelling (HMM). One model is created for each phone and for each vowel transition generalised for place of articulation (cf. Koreman et al., 1998a,b). Left-to-right hidden Markov models are used, consisting of three states (no states

Phonus 5, Institute of Phonetics, University of the Saarland, 2000, 35-46.

were allowed to be skipped) and one mixture per state. No lexicon or language model was used.

In previous experiments using a hybrid phone identification system (Koreman et al., 1999) it was shown that acoustic-phonetic mapping can increase the phoneme identification rate, because the HMM phone identification system uses relatively homogeneous phonetic features instead of more variable acoustic parameters as its input. In the experiments presented in Koreman et al. (1998a,b, 1999) acoustic-phonetic mapping was performed by means of Kohonen maps. In this paper another type of network is used: a fuzzy neural network sensitive to density (FNNSD). Acciani et al. (1999) have shown that a FNNSD can identify colour shades quite effectively. Colour shades consist of non-uniformely distributed, overlapping clusters. Speech data typically share these properties, since their distribution is usually non-uniform (see for example the distribution of the burst centre frequencies of [p, t, k] for different vowels in Liberman et al., 1952, reprinted as Fig. 6.14 in Borden et al., 1994) and it is known that there is considerable overlap between different phonemes, depending on dialect, speaker sex, speaker, accent, context, etc. (cf. also Peterson & Barney, 1952). Because speech data share these properties with colour shades, our expectation is that the application of a FNNSD should be useful for speech data, too.

2. Acoustic-phonetic mapping

FNNSD's belong to the class of clustering neural networks. Their goal is to divide an input space into different regions where data clusters are present, independently of their position, size and overlap with other regions; the only assumption a FNNSD makes about the regions concerns their shape: they must be hyperspherical, i.e. the mean distance between all points of a cluster and its centre is the same in all directions. Detection of each cluster in the input space is achieved by using a feedback neuron which is sensitive to density. At the end of the training phase only one neuron remains for each cluster in the input space. All other neurons are rejected from the network (see Section 2.1 for a more detailed description of the training process).

In contrast, a Kohonen map covers the complete input space, positioning more neurons where the density of input data is greater (many data points packed closely together). This aspect may lead to problems in training (as shown in Becker, 1991), because clusters with few input vectors (rare phones) may not be not modelled in the Kohonen map in the presence of more frequent phones. Differences in phone frequencies are large in our data. For example, /ɲ/ occurs only in Italian, and even in

that language not very often, while /n/ is very frequent. This can cause /ɲ/ not to be modelled by any neuron in the Kohonen map, while /n/ takes up a lot of space, i.e. is modelled by a large number of neurons. This dependency on frequency of occurrence does not exist for a FNNSD (as for some other types of clustering networks).

An additional, practical advantage of a FNNSD over a Kohonen network is that, since fewer neurons are trained in a FNNSD than in a Kohonen map, it also leads to a lower CPU load, which is a clear advantage if the system must be available on a computer beside other applications.

To apply the FNNSD for acoustic-phonetic mapping, two problems with the representation of the input space must be solved. A pilot experiment performed with 12 mel-frequency cepstral coefficients (MFCC's) failed, probably because the acoustic clusters are not hypospherical. Two possible reasons for this are 1) the representation of the input space that we used and 2) the fact that in speech, phones are not realised as clearly distinct clusters.

First, the "Lab" colour space developed by the Commission Internationale de l'Éclairage, which was used in Acciani et al. (1999), has the important advantage that the Euclidian distance between vectors corresponds to perceived *similarity* in human perception. This is not necessarily true for the acoustic parameter space used in our pilot experiment, which is represented by 12 MFCC's. We therefore also experimented with another acoustic parameter type, namely mel-frequency spectral coefficients (MFSC's), as well as with another distance measure (based on the inner product). These experiments are described in Section 3.

Second, since phones are not static signals which are sequenced like beads on a string, but rather show continuous change reflecting the movement from one articulatory position to the next, their acoustic parameters are distributed evenly over the entire input space. A Kohonen network is quite able to represent such an input space in the phonotopic map which is created (Kohonen, 1988). This is not true for a FNNSD, since the more or less even distribution of the data over the input space does not conform to the requirement of hypospherical clusters. For this reason, we have used only steady states of phones, under the assumption that by ignoring the transitions between phones, clearer clusters will build themselves in the acoustic input space. The transitions, which are defined as 35-ms sections of the signal at the beginning and at the end of each vowel (cf. Furui, 1986) – or half the vowel if its duration was less than 70 ms – were therefore not used to train the FNNSD. In the acoustic-phonetic mapping, they should be modelled by a change in the membership

function (see Section 2.3). All experiments with FNNSD's reported in this article are based on FNNSD's trained without transitions.

2.1. Training of the FNNSD

In a FNNSD, each neuron has four fundamental properties: position, radius, slope of its activity function and sensitivity. The first three are stored at the end of training and describe the neuron and the cluster which it represents, while the last property is only used in the training phase and represents the plasticity of the neuron, i.e. the extent to which the other three parameters are affected by an acoustic input.

The schema of a neuron in the neural network is shown in Figure 1.

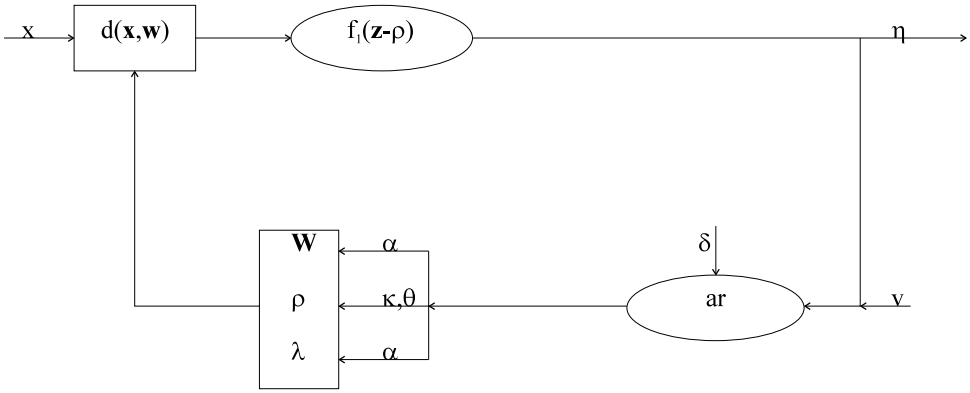

Figure 1. Topology of a neuron in a Fuzzy Neural Network Sensitive to Density

Before starting the training phase six parameters must be set by the user:

- NN_0 = initial number of neurons, which should be greater than the expected number of clusters
- ρ_0 = initial radius of each neuron
- β_0 = initial (and minimal) slope – see Figure 1
- α_0 = initial sensitivity for each neuron, to set its mobility and plasticity
- δ = degree of variation of the sensitivity (constant)
- θ = threshold used in feature calibration and projection

After the initialisation of the network, there are two sub-processes: *analysis* and *collision detection*. The input data are presented in random order. The analysis consists of the following steps for each input vector:

1. The response (η) of each neuron to the input vector is computed according to Figure 2 shown below, where d is distance between the neuron and the input vector. It is clear from the figure that the distance measure is very important for the behaviour of the network.

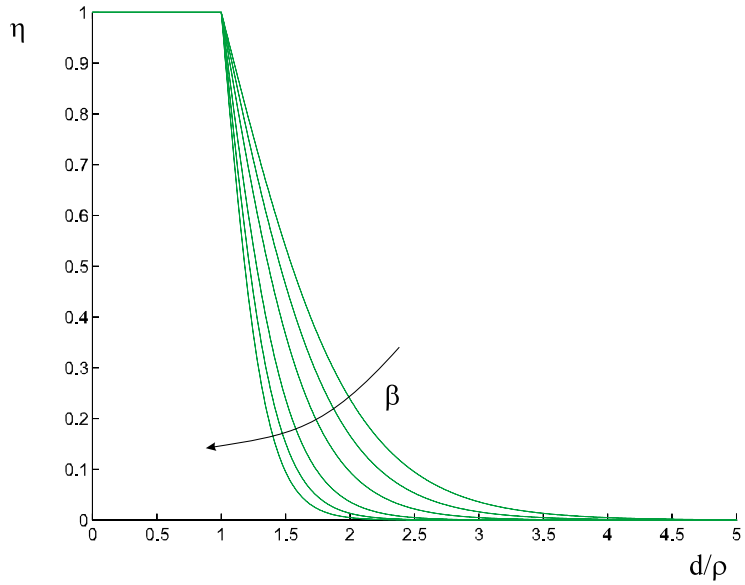

Figure 2. Response of a neuron dependent on its distance to the input vector

2. The most active neuron (with the greatest η) is then updated according to the current sensitivity and η value:
   - it is moved towards the current input
   - its radius is adjusted so as to cover the previously covered region plus the new input
   - its slope is decreased
   - its sensitivity is decreased on the basis of the δ value
3. The position of all other neurons remains the same, their radius and sensitivity are increased, slope is decreased.

In this way all neurons will try to cover a different region, although some may collide with other neurons and some others may never be activated. After the analysis process, collisions between the neurons are detected. If more neurons cover the same region (as shown in Acciani et al., 1999) the neurons with a larger radius and/or greater sensitivity are removed from the network. Each neuron tries to cover the region with as large a radius as possible, if input data are present and if other stable

neurons do not collide with it. In case of non-hyperspherical regions, more neurons will cover the same region, but will not collide because they identify different hyperspherical regions. Neurons which have not been activated are also removed. An example of a collision is presented in Figure 3, in which neuron B (which has the largest radius and the greatest sensitivity of the four neurons) will be removed.

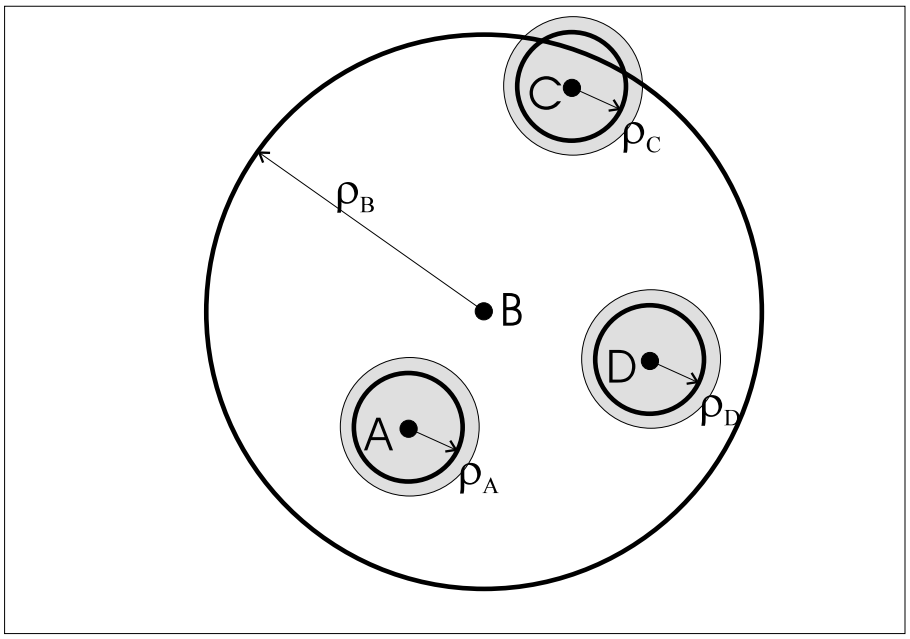

Figure 3. Example of a collision.

Note that the only connection between neurons is the collision detection, which does not permit more than one winning neuron. After analysis and collision detection, the training process repeats these two sub-processes until no neuron is removed from the network. The last analysis, in which the neurons in the network adjust their final position, radius and slope, is called *annealing*, in analogy to the use of the term in integrated circuit manufacturing. Each neuron will eventually cover regions in the input space which were previously covered by removed neurons. The training leads to a more stable structure of the FNNSD, matching the clusters in the input space. From this moment on, the response η of the neuron reflects the membership value of an input vector to a cluster. During the training, labels were not used, therefore the training is unsupervised.

Now the network is ready to distinguish different acoustic inputs, recognise or map them. The correspondence between phones and neurons is not necessarily one-to-one, i.e. one phone can be modelled by more neurons and vice versa. The presence of

different allophones in the input data can cause a phoneme to be modelled by more than one neuron. Since our input data consist of texts read by four speakers (2 male, 2 female) for each of four languages (English, German, Dutch, Italian) from the Eurom0 database, and since the phonemes occur in different contexts within a text, it is obvious that many phonemes will have several allophonic realisations. When one neuron models more than one phoneme, its feature specification is a combination of the phonological features that define each of the phonemes.

2.2. *Feature calibration of the FNNSD*

Because we want to use the FNNSD to map acoustic parameters onto phonetic features, a feature calibration is carried out for the trained network, in which an ouput vector of phonetic features is associated with each neuron. During the feature calibration, a vector of "ideal" phonetic (IPA) features is attached to each acoustic parameter vector. This phonetic feature vector is derived from the phoneme label (listed in the label file which corresponds to the signal file) to which the input frame, i.e. the vector of acoustic parameters, belongs. As before, each neuron reacts with a value of η to the input frame. If the value of η is greater than a threshold θ, the neuron updates its output vector on the basis of the attached phonetic feature value, proportionally to η. At the end of the feature calibration we obtain a map of phonetic features in the acoustic space. Table 1 shows the output vector values for two neurons. Not all neurons have saturated output values, because they have reacted to different phones. This is the case for neuron nr. 88, which has high values for the place-of-articulation feature [glo] (glottal), although it has also been activated by some labial, dental and alveolar phones. It was hardly ever activated by palatal, uvular and velar phones. Mainly fricatives, but also some plosives led to the neuron's activation. It was activated somewhat more often by voiceless than by voiced phones. Neuron nr. 184 shows completely saturated values for all phonetic features, i.e. it was always activated by the phone voiceless alveolar fricative [s].

Table 1. Phonetic feature vectors for neurons nr. 184 and 88 at the end of feature calibration

N_i	lab	den	alv	pal	vel	uvu	glo	plo	fri	nas	lat	appr	voi
184	-1	-1	1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1
88	0.26	-0.28	-0.26	-0.94	-0.94	-0.94	0.94	-0.40	0.40	-0.94	-0.94	-0.94	-0.22

2.3. *Mapping*

In the last step, the FNNSD performs the acoustic-phonetic mapping. For each input vector, a membership value η is computed for each neuron in the network. Each neuron with $\eta > \theta$ (the same θ as used in the feature calibration) contributes to the final value of the output vector of phonetic features, proportionally to η.

In Section 2 we explained that the FNNSD is trained on steady states only, i.e. transitions were discarded in its training and calibration. In acoustic-phonetic mapping, a phonetic feature vector is computed on the basis of η in exactly the same way as it is done for acoustic vectors belonging to the steady states (the non-transitional signal frames): ideally, a transition from phone A to phone B should lead to a decrease in the membership value for the neuron representing phone A and an increase in the membership value for the neuron representing phone B.

3. Representation of the input space

In a pilot experiment, a FNNSD was trained with 12 MFCC's and energy. The Euclidian distance was used. This representation of the input space was chosen for comparability with previous experiments in which Kohonen maps were used for the acoustic-phonetic mapping. These were trained with 12 MFCC's, energy and the corresponding delta parameters, and Euclidian distances were used to update the neurons in the training and to compute the output vector (which was the phonetic feature vector associated with the winning neuron). Unlike in the Kohonen network, we did not normalise the parameters for differences in range, because this would lead to a distortion of the input space – which would make hyperspherical clusters less likely. The training resulted in a network in which only one neuron survived the training process. A possible reason for this is that the clusters in the input space are not hyperspherical (see Section 2), so that the network was unable to distinguish them: the single neuron covers the complete input space as a unique large cluster. Since we want to identify different phones, a network consisting of a single neuron is obviously useless for ASR. In Section 2, it was pointed out that the input space used in previous experiments, in which a Kohonen map was used for acoustic-phonetic mapping, does not conform to the FNNSD's requirement that it can be divided into hyperspherical clusters. Therefore, we shall use another signal representation and two different distance algorithms in the experiments presented below.

3.1. *MFCC's versus MFSC's*

Many state-of-the-art ASR systems use MFCC's for their input parameters, because they are compact (normally 12 coefficients suffice to cover the relevant frequency range of speech), they take into account redundancy between different parts of the spectrum and the mel-frequency scale has been shown to appropriately model the sensitivity of human perception in different frequency ranges. Since the choice of this input space leads to a FNNSD which consists of only a single neuron, we decided to attempt building a FNNSD on the basis of another type of acoustic parameter.

Since MFCC's are difficult to interpret we decided to use a mel-frequency spectral representation (24 MFSC's), which is closer to the visual representation in a spectrogram and is therefore more readily accessible to the human observer.

3.2. *Euclidian distance and inner product distance*

We used two different distance criteria in the experiment using MFSC's: the Euclidian distance (as the baseline and most simple algorithm, which was also used with MFCC's in the Kohonen map) and the cosine of the angle between vectors, based on the inner product (therefore inner product distance). This distance has one important advantage over the Euclidian distance when applied to MFSC's. Since the Euclidian distance is based on absolute differences between corresponding vector elements, the comparison of two realisations of the same phone (i.e. with same distribution of energy over the frequency bands) with a different *overall* energy can result in a greater Euclidian distance than the comparison of two different low-energy phones. The inner product distance does not share this disadvantage and is more selective to differences in the distribution of energy over different frequency bands.

Looking at the left half of Figure 4, one might conclude that the Euclidian distance is more appropriate to represent the distance in the input space than the inner product distance. Comparison with the right two figures shows, however, that the angle distinguishes between /n/ and /z/ much better, despite the greater variability of the inner product distance for the realisations of /n/.

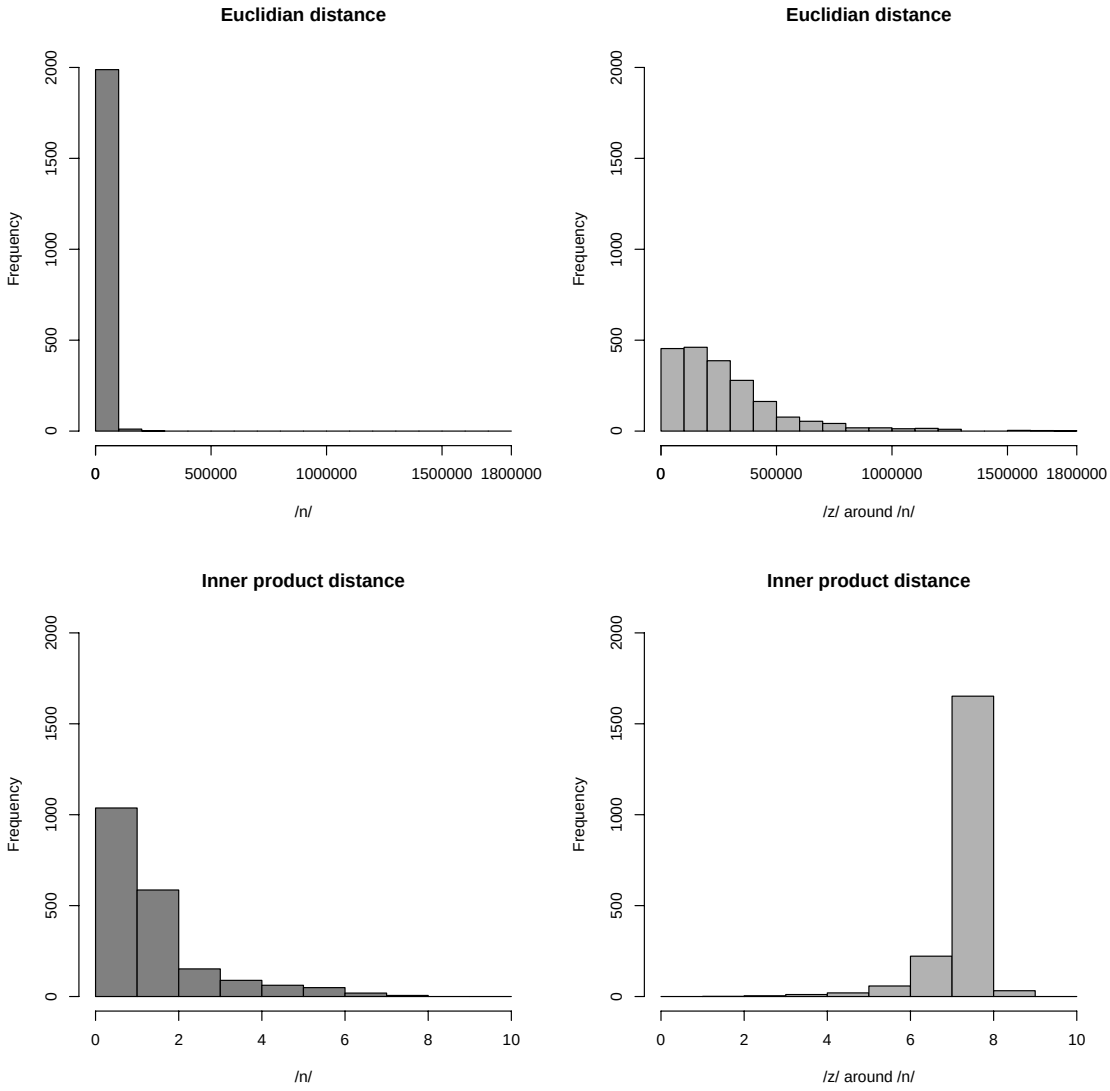

Figure 4. Euclidian distance (top half) and inner product distance (bottom half) for one realisation of /n/ compared to 2000 realisations of /n/ (left two figures) and for the same realisation of /n/ compared to 2000 realisations of /z/ (right two figures)

4. Results and discussion

Consonant identification rates for two hidden Markov modelling experiments based on the output of the acoustic-phonetic mapping are presented here. The acoustic-phonetic mapping is performed by a FNNSD, which is trained with 24 MFSC's. When the FNNSD uses a Euclidian distance measure, the consonant identification rate is as low as 12.6%. When the distance measure is based on the inner product consonant identification rises to 17.9%. This shows that a representation of the input space by MFSC's using the inner product as a distance measure is more appropriate.

At the same time, the consonant identification rate is still much lower compared to when a Kohonen network is used for acoustic-phonetic mapping (52.2% – see Koreman et al., 1998b). A clear advantage of the Kohonen map is that it can model any input space. FNNSD's make an important assumption about the input space, namely that the clusters in the input space are hyperspherical. Further experiments are needed to optimise the representation of the input space. This can be achieved by using different input parameters or by using a different distance measure. It was shown that using a MFSC representation for speech leads to distinct clusters in the network, whereas one cluster covering the whole input space is created when MFCC's are used (see Section 3). Using the inner product instead of the Euclidian distance led to a further improvement of the representation of the input space. In Section 3.2 we argued why the inner product is a more appropriate distance measure for MFSC's. Still, the representation of the input space remains a problem. Until this problem has been solved, we cannot hope to understand the relationship between phones and clusters in the input space.

5. References

Acciani, G., Caradonna, R. & Chiarantoni, E. (1999). A density based membership function for fuzzy clustering. *Proc. Int. Joint Conf. on Neural Networks '99*, Washington DC.

Becker, S. (1991). Unsupervised learning procedures for neural networks. *Intern. Journal for Neural Systems* **2**(1-2), 17-33.

Borden, G.J., Harris, K.S. & Raphael, L.J. (1994). *Speech Science Primer. Physiology, Acoustics, and Perception of Speech.* Baltimore: Williams & Wilkins.

Caradonna, R. (1998). *Riconoscimento delle sfumature di colore con rete neuro-fuzzy* (M.Sc. thesis, Politecnico di Bari).

Furui, S. (1986). On the role of spectral transitions for speech perception. *J. Acoust. Soc. Am.* **80**(4), 1016-1025.

Kohonen, T. (1988). The "Neural" Phonetic Typewriter. *IEEE Computer* **21**(3), 413-424.

Koreman, J., Andreeva, B. & Barry, W.J. (1998a). Do phonetic features help to improve consonant identification in ASR? *Proc. 5th Int. Conf. on Spoken Lang. Proc. '98*, Sydney.

Koreman, J., Barry, W.J. & Andreeva, B. (1998b). Exploiting transitions and focussing on linguistic properties for ASR. *Proc. 5th Int. Conf. on Spoken Lang. Proc. '98*, Sydney.

Koreman, J., Andreeva, B. & Strik, H. (1999). Acoustic parameters versus phonetic features in ASR. *Proc. 14th Int. Congress of Phonetic Sciences*, San Francisco, 719-722.

Liberman, A.M., Delattre, P.C. & Cooper, F.S. (1952). The role of selected stimulus-variables in the perception of the unvoiced stop consonants. *Am. J. Psychol.* **LXV**, 497-516.

Peterson, G.E. & Barney, H.L. (1952). Control methods used in a study of the vowels. *J. Acoust. Soc. Am.* **24**, 175-184.