From the Proc. *International Joint Conference on Neural Networks*, Seattle, 1991.

Global Optimization of a Neural Network - Hidden Markov Model Hybrid

Yoshua Bengio, Renato De Mori, Giovanni Flammia, Ralf Kompe
School of Computer Science, McGill University, 3480 University Str., H3A2A7, Montreal, Qc., Canada

Abstract

In this paper an original method for integrating Artificial Neural Networks (ANN) with Hidden Markov Models (HMM) is proposed. ANNs are suitable to perform phonetic classification, whereas HMMs have been proven successful at modeling the temporal structure of the speech signal. In the approach described here, the ANN outputs constitute the sequence of observation vectors for the HMM. An algorithm is proposed for *global* optimization of all the parameters. Results on speaker-independent recognition experiments using this integrated ANN-HMM system on the TIMIT continuous speech database are reported.

1 Introduction

In spite of the fact that speech exhibits features that cannot be represented by a first-order Markov model, Hidden Markov Models (HMMs) of speech units (*e.g.*, phonemes) have been used with a good degree of success in Automatic Speech Recognition (ASR) (Rabiner & Levinson 85; Lee & Hon 89). Artificial Neural Networks (ANNs) have proven to be useful for classifying speech properties and phonemes based on the analysis of a speech segment of limited duration (Bengio *et al* 89; see Lippman 89 for review). Various attempts have been made to interpret the time evolution of ANN outputs. Worth mentionning is the post-processor proposed by Robinson and Fallside (1990) which uses dynamic programming with duration and bigram constraints. Along a similar line, researchers have attempted to combine the classification power of ANNs with the time-domain modeling capability of HMMs (Bengio *et al* 90; Bourlard & Wellekens 88; Franzini, Lee & Waibel 90; Morgan & Bourlard 90) or to formalize HMMs in the framework of ANN theory (Bridle 90; Levin 90). In this paper, continuous densities HMMs (CDHMMs) are considered in conjunction with networks trained with the generalized delta rule (Rumelhart *et al* 86). It is shown how to perform a joint *global optimization* of both the ANN and the HMM parameter estimation. In the proposed algorithm, the gradient of the optimization criterion with respect to the transformed observations is computed for the HMM system. The HMM can be trained with traditional methods (Rabiner 89) with which the gradient of an optimization criterion can be computed. This gradient is sent to the ANN for the estimation of the weight associated to each connection of the network. No assumption need to be made or constraints imposed on the network outputs, except that the network output distribution should be modeled by a mixture of multivariate gaussians. Multiple ANNs are combined and an incremental design method is described in which specialized networks are integrated to the recognition system in order to improve its performance.

2 Related Work

Interesting papers have been published recently, describing attempts at combining ANNs with HMMs. In some of the proposed approaches (*e.g.*, Franzini, Lee & Waibel 90; Bridle 90) the activation value of each output node of the network corresponds to $P(observation \mid state)$, the observation probability conditional to the state of the HMM (that will be indicated later as $b_{i,t}$). The ANN is trained to compute these observation probabilities for the best sequence of states produced by the alignment. In (Franzini, Lee & Waibel 90) the input data are aligned with the model of the spoken utterance with the Viterbi algorithm. In this case, the observation probabilities are approximated by the network outputs. Another approach was proposed by Bridle (1990) and consists in computing the gradient of an optimization criterion with respect to all the observation probabilities and to use gradient descent to estimate network parameters (including the parameters of the HMM, which is viewed as a recurrent ANN). Other hybrid systems combining ANNs with HMMs (*e.g.*, Bourlard & Wellekens 88; Morgan & Bourlard 90) theoretically require that the ANN parameter estimation has converged to the global minimum in order to express the posterior probability $P(state \mid observation)$. Our previous work on hybrid models (Bengio *et al* 90) used ANNs merely to compute an additional set of symbols considered as observations for a discrete HMM. A vector-quantized codebook was generated for these parameters and added to codebooks obtained for other popular parameter sets. This did not require any assumption on the network outputs but had the disadvantage that the ANN and the HMM were optimized separately. The method described in the present paper allows to perform global parameter optimization by transmitting to the ANN a gradient computed for the HMM.

3 Gradient Computation in the hybrid ANN/HMM system

For this paper, only left-to-right HMMs with a single final state are assumed. Let Y_t be the vector of ANN outputs at time t. These outputs are considered as observations of a CDHMM used in the scheme shown in Figure 1. Let Y_1^T be the whole observation sequence for the HMM, T is the length of the observation sequence, and Y_t a particular observation, made when the HMM is in the state S_t at time t. Let a_{ij} be the transition probability from state i to state j. The probability that the HMM generates Y_t in state S_t at time t is denoted as $b_{i,t} = P(Y_t \mid S_t = i)$ Algorithms (Rabiner 89) allow one to efficiently compute the following probabilities for partial sequences (up to time t, from time $t+1$ on) and the posterior probabilities of state occupancy:

$$\begin{aligned} \alpha_{i,t} &= P(Y\,_1^t \text{ and } S_t = i \mid model) &&= b_{i,t} \sum_j a_{ji}\alpha_{j,t-1} \\ \beta_{i,t} &= P(Y\,_{t+1}^T \mid\ S_t = i \text{ and } model) &&= \sum_j a_{ij} b_{j,t+1}\beta_{j,t+1} \\ \gamma_{i,t} &= P(S_t = i \mid Y\,_1^t \text{ and } model) &&= \alpha_{i,t}\,\beta_{i,t} \end{aligned} \quad (1)$$

with appropriate boundary conditions. If the task is to model isolated units (*e.g.*, isolated words), there will be multiple models ω, one for each unit. For continuous speech recognition, unit models (*e.g.* phonemes) are concatenated to make word and sentence models. The likelihood that a HMM has generated the observation corresponding to the pronounciation of the unit ω is $L_\omega = \alpha_{F_\omega,T}$, where F_ω is the final state for model ω. HMM parameters can be estimated with different criteria. Two popular criteria are Maximum Likelihood (ML) and Maximum Mutual Information (MMI). Modeling with these two criteria is discussed in (Nadas, Nahamoo & Picheny 89). Maximum Likelihood Estimation (MLE) is based on the maximization of the criterion C expressed as $C_{MLE} = L_c$ where, for isolated unit modeling, c represents the pronounced unit. Let us define

$$H_{isolated} = \frac{L_c}{\sum_\omega L_\omega} \quad (2)$$

In the case of Maximum Mutual Information Estimation (MMIE) for isolated unit modeling, the following criterion can be used:

$$C_{MMIE} = \log(H_{isolated}) = \log(\frac{L_c}{\sum_\omega L_\omega}) \quad (3)$$

The mutual information between the correct model c and the observation Y_1^T is

$$I = \log(\frac{P(Y_1^T, model_c)}{P(Y_1^T)P(model_c)}) = \log(\frac{P(Y_1^T \mid model_c)}{\sum_\omega P(Y_1^T \mid model_\omega)P(model_\omega)}) \quad (4)$$

Assuming equal prior probabilities for each model, maximizing C_{MMIE} as in equation 3 also maximizes the mutual information I. For continuous speech, we assume that there is a single HMM built by concatenating unit models. During *training*, we consider a constrained model τ that is made of the concatenation of the units that form the training sentence. On the other hand, during *recognition* all the transitions from one unit to another one are possible and we use an unconstrained model ρ, for example a loop model (see Lee & Hon 89). Hence, for continuous speech, C_{MMIE} can be expressed as

$$C_{MMIE} = \log(H_{continuous}) = \log(\frac{L_\tau}{L_\rho}), \qquad \text{where} \qquad H_{continuous} = \frac{L_\tau}{L_\rho} \quad (5)$$

$L_\tau = \alpha_{F_\tau,T}$ denotes the likelihood of the training model and $L_\rho = \alpha_{F_\rho,T}$ denotes the likelihood of the recognition model. Assume $b_{i,t}$ can be represented by gaussian mixtures as follows:

$$b_{i,t} = \sum_k \frac{Z_k}{((2\pi)^n \mid \Sigma_k \mid)^{1/2}} \exp(-\frac{1}{2}(Y_t - \mu_k)\Sigma_k^{-1}(Y_t - \mu_k)^T) \quad (6)$$

where n is the number of observation features of the HMM. The transition probabilities a_{ij}, normal distribution mean vectors μ_k, covariance matrices Σ_k, and gains Z_k can be estimated as in (Rabiner 89). A derivative of the cost function with respect to $b_{i,t}$ can be computed and used for estimating the parameters of the ANN as it will be shown in the next section.

4 Estimation of ANN parameters

As the optimization criterion C depends on the parameters Y_1^T computed by the ANN, it is possible to express C as a function of them and derive the following equation, using the chain rule:

$$\frac{\partial C}{\partial Y_{jt}} = \sum_i \frac{\partial C}{\partial b_{i,t}} \frac{\partial b_{i,t}}{\partial Y_{jt}} \tag{7}$$

for all the ANN output units j (Y_{jt} is the j^{th} element of the network output vector Y_t). The negative of this gradient can be used with backpropagation [1] to estimate the ANN weights w_{mn}. In the case of MLE, the derivative of C_{MLE} with respect to $b_{i,t}$ is simply

$$\frac{\partial C_{MLE}}{\partial b_{i,t}} = \frac{\partial L_{model}}{\partial b_{i,t}} = \frac{\partial \alpha_{F_{model},T}}{\partial b_{i,t}} \tag{8}$$

where *model* is the training model (the correct word model, in the case of isolated units modeling). In the case of MMIE, the gradient of the optimization criterion C_{MMIE} with respect to the observation probabilities $b_{i,t}$ can be expressed as $\frac{\partial C}{\partial b_{i,t}} = \frac{1}{H}\frac{\partial H}{\partial b_{i,t}}$ where H is defined as in equations 2 and 5 for isolated and continuous speech modeling, respectively. In the case of isolated units modeling, for states i that are in a unit model ω:

$$\frac{\partial H_{isolated}}{\partial b_{i,t}} = \frac{(\delta_{c\omega} - H_c)}{\sum_\omega L_\omega} \tag{9}$$

For continuous speech, we have the following derivative:

$$\frac{\partial H_{continuous}}{\partial b_{i,t}} = \alpha_{F_\rho,T} \frac{\partial \alpha_{F_\tau,T}}{\partial b_{i,t}} - \alpha_{F_\tau,T} \frac{\partial \alpha_{F_\rho,T}}{\partial b_{i,t}} \tag{10}$$

In general, for every optimization criterion C that can be expressed as a differentiable function of the likelihood L, it is possible to compute $\frac{\partial C}{\partial L}$. By differentiating equation (6), $\frac{\partial b_{i,t}}{\partial Y_{jt}}$ can be expressed as follows:

$$\frac{\partial b_{i,t}}{\partial Y_{jt}} = \sum_k \frac{Z_k}{((2\pi)^n \mid \Sigma_k \mid)^{1/2}} (\sum_l d_{k,lj}(\mu_{kl} - Y_{lt})) \exp(-\frac{1}{2}(Y_t - \mu_k)\Sigma_k^{-1}(Y_t - \mu_k)^T) \tag{11}$$

where $d_{k,lj}$ is the element (l,j) of the inverse of the covariance matrix (Σ^{-1}) for the k^{th} gaussian distribution and μ_{kl} is the l^{th} element of the k^{th} gaussian mean vector μ_k. Then, following Bridle (1990), it is possible to compute using (1)

$$\begin{aligned} \frac{\partial \alpha_{F_{model},T}}{\partial b_{i,t}} &= \frac{\partial \alpha_{F_{model},T}}{\partial \alpha_{i,t}} \frac{\partial \alpha_{i,t}}{\partial b_{i,t}} = (\sum_j \frac{\partial \alpha_{j,t+1}}{\partial \alpha_{i,t}} \frac{\partial L_{model}}{\partial \alpha_{j,t+1}})(\sum_j a_{ji}\alpha_{j,t-1}) \\ &= (\sum_j b_{j,t+1}a_{ji} \frac{\partial \alpha_{F_{model},T}}{\partial \alpha_{j,t+1}})(\sum_j a_{ji}\alpha_{j,t-1}) = \beta_{i,t} \frac{\alpha_{i,t}}{b_{i,t}} = \frac{\gamma_{i,t}}{b_{i,t}} \end{aligned} \tag{12}$$

for any hidden Markov *model*, where *model* is ω for isolated units modeling, or ρ (recognition model) or τ (training model) for continous speech modeling.

5 Experimental Results

A preliminary experiment has been performed using a prototype system based on the integration of ANNs with HMMs. The task is the recognition of plosive sounds in every context and pronounced by a large speaker population. The TIMIT continuous speech database (Zue, Seneff & Glass 90) has been used for this purpose. SI and SX sentences from regions 2, 3 and 6 were used, with 1080 training sentences and 224

[1] It replaces the usual $\partial E_p/\partial Y_{jt} = (Y_{jt} - target_{jt})$ for output units, for a particular pattern p, as used in (Rumelhart *et al* 86), where $target_{jt}$ would be the desired output at time t for unit j.

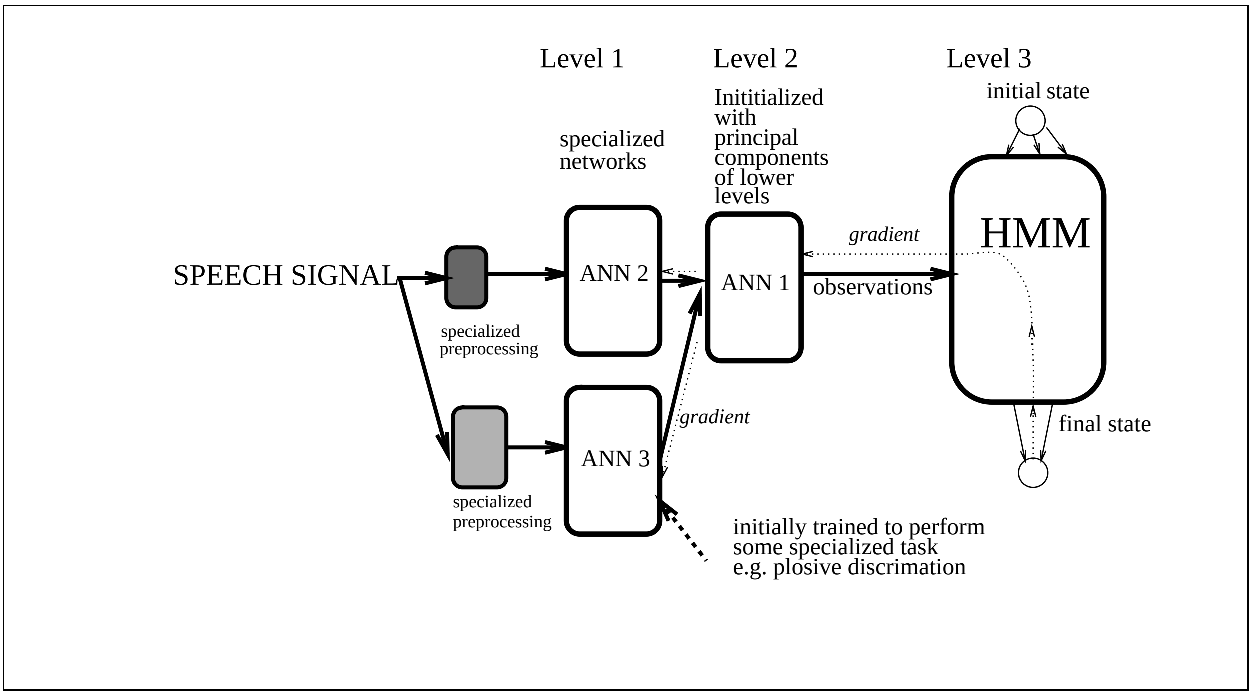
Level 1
Level 2
Level 3
Inititialized with principal components of lower levels
initial state
specialized networks
HMM
gradient
SPEECH SIGNAL
ANN 2
ANN 1
observations
specialized preprocessing
gradient
final state
ANN 3
specialized preprocessing
initially trained to perform some specialized task e.g. plosive discrimation

Figure 1: *Extension of the ANN/HMM hybrid to a hierarchy of modules, with three levels.*

Table 1: Comparative Results

	% rec	% ins	% del	% subs	% acc
ANNs	85	32	0.04	15	53
ANNs+HMM	86	11	0.70	13	75
ANNs+HMM+global opt.	90	3.8	1.4	9.0	86

test sentences, 135 training speakers and 28 test speakers [2]. The following 8 classes have been considered: /p/,/t/,/k/,/b/,/d/,/g/,/dx/[3], /all other phonemes/ Speaker-independent recognition of plosive sounds in continuous speech is a particularly difficult task because these sounds are made of short and non-stationary events that are often confused with other acoustically similar consonants or may be included into other unit segments by a recognition system.

The experimental system is based on the scheme shown in Figure 1. Rather than having a single ANN that computes the vector Y of parameters, we have a hierarchy of networks. Such an architecture is built on three levels. Level 3 contains the HMMs. Level 2 is made of a single ANN that acts as an integrator of parameters generated by more specialized ANNs. ANN1 is a linear network that initially computes the principal components of the concatenated output vectors of the lower level networks (ANN2 and ANN3). At level 1, two ANNs are initially trained to perform plosive recognition (ANN3) and broad classification (ANN2) respectively. In the experiment described below, the combined network (ANN1+ANN2+ANN3) has 23578 weights. The broad classification net (ANN2) has five outputs corresponding to five broad categories[4]. The twelve input nodes to ANN2 are the energies of five band-pass filters in the time domain covering the range up to 7 kHz, the signal total energy, and their six time derivatives. The plosive recognition net (ANN3) has sixteen outputs corresponding to place, manner and degree of voicing, with different instantiations of each place nodes depending on the right context[5]. The 74 inputs to ANN3 are the outputs of 32 Bark-scaled triangular filters computed from the short-time Fast Fourier Transform of

[2] The training speakers were those with initial between "a" and "r" inclusively; the remaining speakers were used for test.

[3] The flapped alveolar plosive */dx/* is considered as a distinct phoneme in the TIMIT database.

[4] non-nasal sonorant, nasal, plosive, fricative, and silence.

[5] Each of the four different places of articulation (labial, alveolar, velar, and flapped alveolar) corresponds to two different nodes, depending on whether the following phoneme has a front or non-front place of articulation. The remaining eight nodes are labeled: unvoiced plosive, voiced plosive, vocalic front,vocalic non-front, liquid, fricative, nasal, silence.

the windowed signal, 30 property detectors approximating a second order derivative over short intervals of frequency and time[6], 7 slope coefficients describing the frequency derivative of the spectrum, the total energy and the voicing energy (in the 60-500 Hz band) and their time derivatives, and a measure of distance (dot product) between neighbouring spectral frames. Input parameters are fed to the networks every 5 msec. ANN2 has time-delay links, while ANN3 has time-delay links between the input nodes and the hidden layer, and recurrent links between some of the hidden nodes and the output nodes. ANN1 computes 8 features for the continuous densities HMM. Each of the 11 unit models [7] had 14 states, 28 transitions, 3 self loops, without explicitly modeling the state duration. Each HMM has tied distributions with 3 basic different distributions characterizing the beginning, middle and final part of a segment modeled by the unit. Each of these distributions is modeled by a gaussian mixture with 5 densities. The covariance matrix is assumed to be diagonal since the parameters are initially principal components and this assumption reduces significantly the number of parameters to be estimated.

In order to assess the value of the proposed approach as well as the improvement brought by the HMM as a post-processor for time alignment, the performance of the hybrid system was evaluated and compared with that of a simple post-processor applied to the outputs of the ANNs. The simple post-processor assigns a symbol to each output frame of the ANNs by comparing the target output vectors with actual output vectors. It then smooths the resulting string to remove very short segments and merges consecutive segments that have the same symbol. The comparative results are summarized in Table I. The overall recognition rate (100% - %deletions - %substitutions) for the 8 classes with the hybrid system after two training iterations is 90% on a total of 7214 phonemes, and its accuracy (100% - %deletions - %substitutions - %insertions) is 86%. Note that this is a significative improvement over the performance obtained with a HMM trained without global optimization (86% recognition and 75% accuracy). The ANNs alone yielded 85% recognition but only 53% accuracy, because of the high number of insertions (32%), mostly due to short plosive segments. The ANNs perform a good classification but have a noisy output with many insertions. The HMM eliminates most of these insertions because of its better duration and temporal modeling. In addition to providing a good temporal model the HMM provides more appropriate target values for the outputs of the ANN. With these target outputs for the ANN, the hybrid system significantly improves its performance. It is interesting to note that the effect of equation 11 is to generate a gradient that tends to bring the output of the ANN closer to the means of the normal densities which are close to the ANN output as well as consistent with the the training string.

6 Conclusion

A system has been proposed to combine the advantages of ANNs and HMMs for speech recognition. The parameters of the ANN and HMM subsystems can influence each other. We showed how to perform a global optimization of such a system by driving the network gradient descent with parameters computed in the HMM. Encouraged by the results of the above-described initial experiments, we will explore further the possibilities of such a hybrid system, and extend it to the recognition of all American-English phonemes. We have seen how such a hybrid system could be extended to integrate multiple ANN modules, which may be recurrent. Note that the hybrid system can use semi-continuous HMMs rather than continuous densities HMMs, and this would probably improve the performance by allowing better models at a lower computational cost. Another interesting extension would be to perform speaker adaptation with the hybrid system. This could be obtained by first training the system as previously described for multiple speakers, and in a second step, adapting *only the ANN parameters* with known sentences from the new speaker. In such a system, the ANN adaptation represents a tuning of the feature space to the new speaker, whereas the temporal model remains unchanged (see (Bridle & Cox 91) for a related speaker adaptation mechanism).

References

Bengio Y., Cardin R., De Mori R. and Normandin Y. 1990. A hybrid coder for hidden Markov models using a recurrent neural network. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Albuquerque, NM, April 90, pp. 537-540.

Bengio Y., Cardin R., De Mori R. and Merlo E. 1989. Programmable execution of multi-layered networks for

[6] This parameter is inspired by studies in Acoustic-Phonetics (see Stevens 1975).

[7] In order to improve its modeling, the rejection class was composed out of four models: nasals, fricatives, non-nasal sonorants, and silence. The recognition results are obtained by merging these four subclasses, such that the total number of classes to recognize is eight.

automatic speech recognition. *Communications of the Association for Computing Machinery*, vol. 32, no. 2, Feb. 89, pp. 195-199.

Bourlard, H. and Wellekens, C.J. 1988. Links between Markov models and multi-layer perceptrons. *Advances in Neural Information Processing Systems 1*, (ed. D.S. Touretzky) Morgan Kauffman Publ., pp. 502-510.

Bridle, J.S. 1990. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *Advances in Neural Information Processing Systems 2*, (ed. D.S. Touretzky) Morgan Kauffman Publ., pp. 211-217.

Bridle J.S. and Cox S.J. 1991. RECNORM: simultaneous normalisation and classification applied to speech recognition. To appear in *Advances in Neural Information Processing Systems 3*, (ed. D.S. Touretzky) Morgan Kauffman Publ.

Cosi P., Bengio Y. and De Mori R. 1990. Phonetically-based multi-layered networks for acoustic property extraction and automatic speech recognition. *Speech Communication* special issue on neurospeech, vol. 9, no. 1, pp. 15-30.

Franzini, M., Lee K-F. and Waibel A. 1990. Connectionist Viterbi training: a new hybrid method for continuous speech recognition. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Albuquerque, NM, April 90, pp. 425-428.

Lee K.F., and Hon H.-W. 1989. Speaker-Independent Phone Recognition Using Hidden Markov Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. ASSP-37, pp. 1641-1648.

Levin, E. 1990. Word recognition using hidden control neural architecture. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Albuquerque, NM, April 90, pp. 433-436.

Lippman R.P. 1989. Review of neural networks for speech recognition. *Neural Computation*, vol. 1, no. 1, pp. 1-38.

Morgan, N. and Bourlard, H. 1990. Continuous speech recognition using multilayer perceptrons with hidden Markov models. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Albuquerque, NM, April 90, pp. 413-416.

Nadas, A., Nahamoo, D. and Picheny, M.A. 1988. On a model-robust training method for speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-36, no. 9., pp. 1432-1436.

Rabiner L.R. and Levinson S.E. 1985. A speaker-independent, syntax-directed, connected word recognition system based on hidden Markov models and level building. *IEEE Trans. Acoust. Speech Signal Processing*, vol. ASSP-33, no. 3, pp.561-573.

Rabiner L.R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, vol. 77, no. 2, Feb. 89., pp. 257-285.

Robinson, T. and Fallside, F. 1990. Phoneme recognition from the TIMIT database using recurrent error propagation networks. Engineering Dept., Cambridge University, CUED/F-INFENG/TR 42.

Rumelhart D.E., Hinton G.E. and Williams R.J. 1986. Learning internal representation by error propagation. *Parallel Distributed Processing* vol. 1, MIT Press, pp. 318-362.

Stevens K.N. 1975. The potential role of properties detectors in the perception of consonants, in *Auditory Analysis and Perception of Speech* G. Fant and M.A. Tatham ed., Academic Press, London, pp. 303-330.

Stewart G.W. 1973. *Introduction to Matrix Computations*. Academic Press, London.

Zue V., Seneff S. and Glass J. 1990. Speech database development: TIMIT and beyond. *Speech Communication*, Vol. 9. No. 4. august 1990, pp. 351-356.