COMBINING NEURAL NETWORKS AND HIDDEN MARKOV MODELS

FOR CONTINUOUS SPEECH RECOGNITION

Michael Cohen*, David Rumelhart, Nelson Morgan***, Horacio Franco*, Victor Abrash*, and Yochai Konig*****

* Speech Research Program, SRI International, Menlo Park, CA 94025
** Stanford University, Dept. of Psychology, Stanford, CA 94305
*** Intl. Computer Science Inst., 1947 Center Street, Suite 600, Berkeley, CA 94704

ABSTRACT

We present a speaker-independent, continuous-speech recognition system based on a hybrid multilayer perceptron (MLP)/hidden Markov model (HMM). The system combines the advantages of both approaches by using MLPs to estimate the state-dependent observation probabilities of an HMM. New MLP architectures and training procedures are presented that allow the modeling of multiple distributions for phonetic classes and context-dependent phonetic classes. Comparisons with a pure HMM system illustrate advantages of the hybrid approach both in recognition accuracy and in number of parameters required.

1. INTRODUCTION

Hidden Markov models (HMMs) are used in most state-of-the-art continuous-speech recognition systems. This approach is limited by the need for strong statistical assumptions that are unlikely to be valid for speech. Techniques using multilayer perceptrons (MLPs) for probability estimation have recently been introduced [1], which reduce the assumption of independence for multifeature probability computation. Additional advantages of MLP probability estimation include the inherently discriminant nature of the training algorithm and the distributed representation, which leads to efficient use of the available parameters. When applied to speech, this results in a reduction of the number of parameters needed for detailed phonetic modeling as a result of increased sharing of model parameters between phonetic classes.

Pure MLP-based approaches have not previously been demonstrated to function well for continuous-speech recognition because of the need for accurate segmentation of the speech signal. HMMs, on the other hand, provide a framework for simultaneous segmentation and classification of speech, which has been demonstrated to be useful for continuous recognition. Previous work by Morgan and Bourlard [1] has shown theoretically and practically that MLPs and HMMs can be combined by using MLPs for the estimation of the HMM state-dependent observation probabilities, thereby exploiting the advantages of both approaches.

We have incorporated MLP-based probability estimation techniques into the HMM-based SRI-DECIPHER (TM) system, which is a state-of-the-art, speaker-independent, continuous-speech recognition system. In this paper we describe the initial baseline DECIPHER (TM) system and the approach for integrating MLP-based estimation techniques; present a number of new techniques developed to allow the modeling of multiple distributions for phonetic classes and context-dependent phonetic classes; and show the results of recognition experiments for the systems described. Abrash et al. [2] and Konig et al. [3] have extended the system presented here to the modeling of long-term speech consistencies.

2. HYBRID MLP/HMM

2.1 The DECIPHER (TM) System

The baseline system into which we incorporated MLP probability estimation is the SRI-DECIPHER (TM) system, a phone-based, speaker-independent, continuous-speech recognition system, based on semicontinuous (tied Gaussian mixture) HMMs [4]. The system extracts four features from the input speech waveform, including 12th-order mel cepstrum, log energy, and their smoothed derivatives. The front end produces the 26 coefficients for these four features for each 10-ms frame of speech.

Training of the phonetic models is based on maximum-likelihood estimation using the forward-backward algorithm [5]. Most of the phonetic models in DECIPHER (TM) have three states, each state having a self transition and a transition to the following state. A small number of phone models have two states, to allow for short realizations.

High recognition performance with HMM systems generally requires context-dependent phonetic models. The context-dependent version of the DECIPHER (TM) system uses phone models trained at a variety of levels of context dependence (in addition to context-independent models), including word-specific phone, triphone, generalized triphone, cross-word triphone (constrained to connect to appropriate contexts in preceding or following words), left and right biphone, and generalized biphone. Models conditioned by more specific contexts are linearly smoothed with more general models using the deleted interpolation algorithm [6] in order to maintain robustness even in highly specific contexts where few training data are available.

In DECIPHER (TM) words are represented as probabilistic networks of phone models, specifying multiple pronunciations. These networks are generated by the application of phonological rules to baseform pronunciations for each word. To limit the number of parameters that must be estimated, phonological rules are chosen based on measures of coverage and overcoverage of a database of pronunciations. This process results in networks that maximize the coverage of observed pronunciations while minimizing network size. Probabilities of pronunciations are estimated by the forward-backward algorithm at the same time the phonetic models are trained, after tying together instances of the same phonological process in different words. Phonological rules can be specified to apply across words, adding initial or final arcs that are constrained to connect only to arcs fulfilling the context of the rules. Phonological modeling in the DECIPHER (TM) system is described in detail by Cohen [7].

Recognition uses the Viterbi algorithm [5] to find the HMM state sequence (corresponding to a sentence) that has the highest probability of generating the observed acoustic sequence.

2.2 Incorporating MLPs into DECIPHER

The hybrid MLP/HMM DECIPHER (TM) system substitutes (scaled) probability estimates computed with MLPs for the tied-mixture HMM state-dependent observation probability densities. No changes are made in the topology of the HMM system.

The initial hybrid system used an MLP to compute context-independent phonetic probabilities for the 69 phone classes in the DECIPHER (TM) system. Separate probabilities were not computed for the different states of phone models. During the Viterbi recognition search, the probability of acoustic vector Y_t given the phone class q_j, $P(Y_t | q_j)$, is required for each HMM state. Since MLPs can compute Bayesian posterior probabilities, we compute the required HMM probabilities using

$$P(Y_t | q_j) = \frac{P(q_j | Y_t) P(Y_t)}{P(q_j)} \quad . \qquad (1)$$

The factor $P(q_j | Y_t)$ is the posterior probability of phone class q_j given the input vector Y at time t. This is computed by a back-propagation-trained [8] three-layer feed-forward MLP. $P(q_j)$ is the prior probability of phone class q_j and is estimated by counting class occurrences in the examples used to train the MLP. $P(Y_t)$ is common to all states for any given time frame, and can therefore be discarded in the Viterbi computation, since it will not change the optimal state sequence used to get the recognized string.

The MLP has an input layer of 234 units, spanning 9 frames (with 26 coefficients for each) of cepstra, delta-cepstra, log-energy, and delta-log-energy that are normalized to have zero mean and unit variance. The hidden layer has 1,000 units, and the output layer has 69 units, one for each context-independent phonetic class in the DECIPHER (TM) system. Both the hidden and output layers consist of sigmoidal units.

The MLP is trained to estimate $P(q_j | Y_t)$, where q_j is the class associated with the middle frame of the input window. Stochastic gradient descent is used. The training signal is provided by the HMM DECIPHER system previously trained by the forward-backward algorithm. Forced Viterbi alignments (alignments to the known word string) for every training sentence provide phone labels, among 69 classes, for every frame of speech. The target distribution is defined as 1 for the index corresponding to the phone class label and 0 for the other classes. A minimum relative entropy between posterior target distribution and posterior output distribution is used as a training criterion. With this training criterion and target distribution, assuming enough parameters in the MLP, enough training data, and that the training does not get stuck in a local minimum, the MLP outputs will approximate the posterior class probabilities $P(q_j | Y_t)$ [1]. Frame classification on an independent cross-validation set is used to control the learning rate and to decide when to stop training as in [9]. The initial learning rate is kept constant until cross-validation performance increases less than 0.5%, after which it is reduced as $1/2^n$ until performance increases no further.

3. MULTIPLE PHONETIC DISTRIBUTIONS

Experience with HMM-based systems has shown the importance of modeling phonetic units with a sequence of distributions rather than a single one. This allows the model to capture some of the dynamics of phonetic segments. SRI's DECIPHER (TM) system models most phones with a sequence of three HMM states. Our initial hybrid system, described in Section 2.2, used an MLP with a single output unit for each phone class. For any particular HMM phone model considered during the Viterbi search, the activation of the output unit corresponding to phone class q_j would be used as $P(q_j | Y_t)$ for all states of the phone model for phone class q_j.

Our initial attempt to extend the hybrid system to the modeling of a sequence of distributions for each phone involved increasing the number of output units from 69 (corresponding to phone classes) to 200 (corresponding to the states of the HMM phone models). This resulted in an increase in word-recognition error rate by almost 30%. Experiments at ICSI had a similar result [9]. The higher error rate seemed to be due to the discriminative nature of the MLP training algorithm. The new MLP, with 200 output units, was attempting to discriminate subphonetic classes, corresponding to HMM states. As a result, the MLP was attempting to discriminate into separate classes acoustic vectors that corresponded to the same phone and, in many cases, were very similar but were aligned with different HMM states. There were likely to have been many cases in which almost identical acoustic training vectors were labeled as a positive example in one instance and a negative example in another for the same output class. The appropriate level at which to train discrimination is likely to be the level of the phone (or higher) rather than the subphonetic HMM-state level (to which these outputs units correspond).

The reason multiple-state models did not cause a similar problem for HMMs is that the standard HMM training procedure is based on maximum-likelihood estimation, which uses each data point to update the density model of the class it has been assigned to. Discriminative training, such as the MLP training procedure described above, focuses on modeling the boundaries between classes. In discriminative training, for each data point, the parameters are adjusted to increase the probability of the correct class and decrease the probability of the incorrect class.

We have developed a new MLP architecture that uses three separate output layers, corresponding to the three states of HMM phone models. Each output layer consists of 69 units, one for each phonetic class. During training, only frames aligned with first states of HMM phones are presented to the first output layer, while frames aligned with last states of HMM phones are presented to the third output layer and those aligned with second states of three-state HMM phones are presented to the second output layer. This MLP can thus be viewed as a set of three MLPs, corresponding to the three HMM state-positions, which have the same input-to-hidden weights. Since the training proceeds as if each output layer were part of an independent net, the system learns discrimination between phonetic classes (as represented within each output layer), but does not learn discrimination between the different states of the same phonetic class (because they are represented in different output layers). During the Viterbi recognition search, the appropriate output layer is referenced depending on which HMM state-position is being visited. This technique has been combined with the approach to context-dependent modeling, described next.

4. CONTEXT DEPENDENCE

Experience with HMM technology has shown that using context-dependent phonetic models improves recognition accuracy significantly [10]. This is because acoustic correlates of coarticulatory effects are modeled explicitly, producing sharper and less overlapping probability density functions for the different phone classes. Context-dependent HMMs use different probability distributions for every phone in every different relevant context. This practice reduces the amount of data available to train phones in highly specific contexts, resulting in models that are not robust. The solution to this problem used by many HMM systems (including DECIPHER) is to use the deleted interpolation algorithm to linearly smooth models trained at different levels of context specificity. This approach cannot be directly extended to MLP-based systems because the smoothing of MLP weights makes no sense. It would be possible to use this approach to average the probabilities from different MLPs; however, since the MLP training algorithm is a discriminant procedure, it would be desirable to use a discriminant procedure to smooth the MLP probabilities.

An earlier approach to context-dependent phonetic modeling with MLPs was proposed by Bourlard et al. [11]. It is based on factoring the context-dependent likelihood and uses a set of binary inputs to the network to specify context classes. The number of parameters and the computational load using this approach are not much greater than those for the original context-independent net.

We have developed a context-dependent modeling approach that uses a different factoring of the desired context-dependent likelihoods, a network architecture that shares the input-to-hidden layer among the context-dependent classes to reduce the number of parameters, and a training procedure that smooths networks with different degrees of context-dependence in order to achieve robustness in probability estimates [12, 13].

Our initial implementation of context-dependent MLPs models generalized biphone phonetic categories. We chose a set of eight left and eight right generalized biphone phonetic-context classes, based principally on place of articulation and acoustic characteristics. The context-dependent architecture is shown in Figure 1. A separate output layer (consisting of 69 output units corresponding to 69 context-dependent phonetic classes) is trained for each context. The context-dependent MLP can be viewed as a set of MLPs, one for each context, which have the same input-to-hidden weights. Separate sets of context-dependent output layers are used to model context effects in different states of HMM phone models, thereby combining the modeling of multiple phonetic distributions described in Section 3 with context-dependent modeling. During training and recognition, speech frames aligned with first states of HMM phones are associated with the appropriate left context output layer, those aligned with last states of HMM phones are associated with the appropriate right context output layer, and middle states of three state models are associated with the context-independent output layer. As a result, since the training proceeds (as before) as if each output layer were part of an independent net, the system learns discrimination between the different phonetic classes within an output layer (which now corresponds to a specific context and HMM-state position), but does not learn discrimination between occurrences of the same phone in different contexts or between the different states of the same HMM phone.

Figure 1: Context-dependent MLP.

4.1 Context-Dependent Factoring

In a context-dependent HMM, every state is associated with a specific phone class and context. During the Viterbi recognition search, $P(Y_t | q_j, c_k)$ (the probability of acoustic vector Y_t given the phone class q_j in the context class c_k) is required for each state. We compute the required HMM probabilities using

$$P(Y_t | q_j, c_k) = \frac{P(q_j | Y_t, c_k) P(Y_t | c_k)}{P(q_j | c_k)} \tag{2}$$

where $P(Y_t | c_k)$ can be factored again as

$$P(Y_t | c_k) = \frac{P(c_k | Y_t) P(Y_t)}{P(c_k)} \quad . \tag{3}$$

The factor $P(q_j | Y_t, c_k)$ is the posterior probability of phone class q_j given the input vector Y_t and the context class c_k. To compute this factor, we consider the conditioning on c_k in Eq. (1) as restricting the set of input vectors only to those produced in the context c_k. If M is the number of context classes, this implementation uses a set of M MLPs (all sharing the same input-to-hidden layer) similar to those used in the context-independent case except that each MLP is trained using only input-output examples obtained from the corresponding context, c_k.

Every context-specific net performs a simpler classification than in the context-independent case because within a context the acoustics corresponding to different phones have less overlap.

$P(c_k | Y_t)$ is computed using a three-layer feed-forward MLP with an output unit corresponding to each context class, the same 234 inputs as the MLP described above, and 512 hidden units. $P(q_j | c_k)$ and $P(c_k)$ are estimated by counting over the training examples. Finally, $P(Y_t)$ is common to all states for any given time frame, and can therefore be discarded in the Viterbi computation, since it will not change the optimal state sequence used to get the recognized string.

4.2 Context-Dependent Training and Smoothing

In order to achieve robust training of context-specific nets, we use the following method:

An initial context-independent MLP is trained, as described in Section 2, to estimate the context-independent posterior probabilities over the N phone classes. After the context-independent training converges, the resulting weights are used to initialize the weights going to the context-specific output layers. Context-dependent training proceeds by backpropagating error from only the appropriate output layer for each training example. Otherwise, the training procedure is similar to that for the context-independent net, using stochastic gradient descent and a relative-entropy training criterion. Overall classification performance evaluated on an independent cross-validation set is used to determine the learning rate and stopping point. Only hidden-to-output weights are adjusted during context-dependent training. We can view the separate output layers as belonging to independent nets, each one trained on a nonoverlapping subset of the original training data.

Every context-specific net would asymptotically converge to the context conditioned posteriors $P(q_j | Y_t, c_k)$ given enough training data and training iterations. As a result of the initialization, the net starts estimating $P(q_j | Y_t)$, and from that point it follows a trajectory in weight space, incrementally moving away from the context-independent parameters as long as classification performance on the cross-validation set improves. As a result, the net retains useful information from the context-independent initial conditions. In this way, we perform a type of nonlinear smoothing between the pure context-independent parameters and the pure context-dependent parameters.

4.3 Recognition

The smoothed context-dependent posterior probabilities supplied by the MLP have to be converted during recognition to (scaled) state-conditioned observation probabilities using the normalization factors provided by Eqs. (1) and (2). However, because these values are a result of smoothing context-dependent and independent networks, the normalization factors should be a combination of those corresponding to the context-dependent and context-independent cases. We use the following interpolation for converting the smoothed posteriors $P^s(q_j | Y_t, c_k)$ to smoothed (scaled) observation probabilities $P^s(Y_t | q_j, c_k)$:

$$P^s(Y_t | q_j, c_k) = P^s(q_j | Y_t, c_k) \cdot \left[\alpha_j^k \frac{1}{P(q_j)} + (1 - \alpha_j^k) \frac{P(c_k | Y_t)}{P(q_j | c_k) P(c_k)} \right] \quad (4)$$

where

$$\alpha_j^k = \frac{N_{ci}(j)}{N_{ci}(j) + b\,[N_{cd}(j,k)]} \quad . \quad (5)$$

$N_{ci}(j)$ is the number of training examples for phone class j for the context-independent net. $N_{cd}(j,k)$ is the number of training examples for the context-dependent net for phone class j and for the context class k. Constant b is optimized on a development set for minimum word-recognition error.

5. EVALUATION

5.1 Methods

We conducted training and recognition experiments comparing the MLP/HMM hybrid to the pure HMM DECIPHER (TM) system using the speaker-independent, continuous-speech, DARPA Resource Management database. The vocabulary size is 998 words. Tests were run both with a word-pair (perplexity 60) grammar and with no grammar. The training set for both the HMM and the MLP consisted of the 3990 sentences that make up the standard DARPA speaker-independent training set for the Resource Management task. The 600 sentences which make up the Resource Management test sets released in February and October of 1989 were used for cross-validation during MLP training and for tuning the word transition weights of the HMM. None of the tests described here use the gender-dependent DECIPHER (TM) system, which is described elsewhere [2].

5.2 Results

Table 1 presents word recognition error and number of system parameters for four different versions of the system, for three different Resource Management test sets (February 91 and two sets released in September 92) using the word-pair grammar. Table 2 presents word recognition error for the corresponding tests with no grammar (the number of parameters are the same as those shown in Table 1).

Comparing context-independent MLP (CI-MLP) with context-dependent MLP (CD-MLP) shows improvements with CD-MLP in all six tests, with an average reduction in word error of 23%. The CD-MLP system combines the multiple-distribution modeling technique described in Section 3 with the context-dependent modeling technique described in Section 4. The CD-MLP system improves word recognition over the context-dependent HMM (CD-HMM) in five out of the six cases, with roughly one-fourth the number of parameters.

The MIXED system uses a weighted mixture of the logs of state observation likelihoods provided by a 2,000-hidden-unit version of the CI-MLP and the CD-HMM [9]. This system shows the best recognition performance so far achieved with the DECIPHER (TM) system on the Resource Management database. In all six tests, it performs better than the CD-HMM system.

The CI-MLP system performs worse than any of the other (context-dependent) systems. This is not surprising, given the simplicity of the system, using a single context-independent output unit for each phonetic class. The CI-MLP system has far fewer parameters than any of the other systems. It is, in fact, interesting that such a relatively simple system performs as well as it does compared with far more complex systems.

Table 1: Number of system parameters and percent word error for pure HMM and hybrid MLP/HMM with no grammar.

	Feb 91	Sep 92a	Sep 92b	# Parameters
CI-MLP	24.7	31.5	30.9	300K
CD-MLP	18.4	27.1	24.9	1,400K
CD-HMM	19.3	29.2	26.6	5,500K
MIXED	15.9	25.4	21.5	6,100K

Table 2: Percent word error for pure HMM and hybrid MLP/HMM with word-pair grammar.

	Feb 91	Sep 92a	Sep 92b
CI-MLP	5.8	10.9	9.5
CD-MLP	4.7	7.6	6.6
CD-HMM	3.8	10.1	7.0
MIXED	3.2	7.7	5.7

5.3 Discussion

The results shown in Tables 1 and 2 suggest that MLP estimation of HMM observation likelihoods can improve the performance of standard HMMs. These results also suggest that systems that use MLP-based probability estimation make more efficient use of their parameters than standard HMM systems. In standard HMMs, most of the parameters in the system are in the observation distributions associated with the individual states. MLPs use representations that are more distributed in nature, allowing more sharing of representational resources and better allocation of representational resources based on training. In addition, since MLPs are trained to discriminate between classes, they focus on modeling boundaries between classes rather than class internals. HMMs which model gender-based speech consistencies typically have twice the number of parameters of gender-independent HMMs. For example, a version of the DECIPHER (TM) system that models male and female speech separately has more than 11 million parameters. Abrash et al. [2] and Konig et al. [3] discuss methods for exploiting the distributed nature of MLPs to model gender-based speech consistencies with only a relatively small increase in the number of parameters.

The distributed representation used by MLPs is exploited in the context-dependent modeling approach described in Section 4 by sharing the input-to-hidden layer weights between all context classes. This sharing substantially reduces the number of parameters to train and the amount of computation required during both training and recognition. The context-dependent MLP has 17 times as many output units (classes) as the context-independent MLP, but has only a factor of 4.6 times as many parameters. In addition, we do not adjust the input-to-hidden weights during the context-dependent phase of training, assuming that the features provided by the hidden layer activations are relatively low level and are appropriate for context-dependent as well as context-independent modeling. This procedure substantially reduces context-dependent training time. The large decrease in cross-validation error observed when going from context-independent to context-dependent MLPs (30.6% to 21.4%) suggests that the features learned by the hidden layer during the context-independent training phase, combined with the extra modeling power of the context-specific hidden-to-output layers, were adequate to capture the more detailed context-specific phone classes.

One should keep in mind that the reduction in memory needs that may be attained by replacing HMM distributions with MLP-based estimates must be traded off against increased computational load during both training and recognition. The MLP computations during training and recognition are much larger than the corresponding Gaussian mixture computations for HMM systems.

The performance of the CD-MLP system was better than that of the CD-HMM system in five out of six tests, despite the fact that the CD-MLP is a far simpler system, with approximately a factor of four fewer parameters and modeling of only generalized biphone phonetic contexts. The CD-HMM system models a range of context-dependent classes, including generalized biphone, specific biphone, generalized and specific triphone, and word-specific phone. The fact that context-dependent MLPs can perform as well or better than context-dependent HMMs relying on far less specific contexts suggests that MLPs may be more vocabulary-independent. In the future, we will compare the performance of the CD-MLP and CD-HMM systems on cross-vocabulary tasks. We are currently exploring an alternative CD-MLP architecture which models context-dependent phonetic classes with many fewer parameters than the current CD-MLP. We are also extending the CD-MLP system to the modeling of more specific context classes, with the hope of further improving recognition performance.

The best performance shown in Tables 1 and 2 is that of the MIXED system, which combines CI-MLP and CD-HMM probabilities. The CD-MLP probabilities can also be combined with CD-HMM probabilities; however, we expect that the planned extension of our CD-MLP system to finer contexts will lead to a system that performs better than our current best system without the need for such mixing, therefore resulting in a simpler system.

6. CONCLUSIONS

MLP-based probability estimation can be useful for both improving recognition accuracy and reducing memory needs for HMM-based speech recognition systems. These benefits, however, must be weighed against the increased computational requirements using MLPs, especially during training.

We have demonstrated a number of systems of different sizes and complexities, and shown some of the tradeoffs between system size, complexity, and performance. In the near future we intend to extend our system to the modeling of finer phonetic contexts and to combine our context-dependent MLPs with MLPs that model gender-based speech consistencies.

ACKNOWLEDGMENTS

The work reported here was partially supported by DARPA Contract MDA904-90-C-5253. Talks with Herve Bourlard were very helpful.

BIBLIOGRAPHY

[1] N. Morgan and H. Bourlard, ‘‘Continuous Speech Recognition Using Multilayer Perceptrons with Hidden Markov Models,’’ *ICASSP 90* , pp. 413-416, Albuquerque, New Mexico, 1990.

[2] V. Abrash, H. Franco, M. Cohen, N. Morgan, and Y. Konig, ‘‘Connectionist Gender Adaptation in a Hybrid Neural Network/Hidden Markov Model Speech Recognition

System,'' *ICSLP 92,* Vol 2., pp. 911-914, Banff, 1992.

[3] Y. Konig, N. Morgan, and C. Chandra, ''GDNN: A Gender-Dependent Neural Network for Continuous Speech Recognition,'' *International Computer Science Institute Technical Report TR-91-071,* December 1991.

[4] H. Murveit, M. Cohen, P. Price, G. Baldwin, M. Weintraub, and J. Bernstein, ''SRI's DECIPHER System,'' *DARPA Speech and Natural Language Workshop,* February 1989.

[5] S. E. Levinson, L. R. Rabiner, and M.M. Sondhi, ''An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition,'' *Bell Syst. Tech. Journal 62,* 1035-1074, 1983.

[6] F. Jelinek and R. L. Mercer, ''Interpolated estimation of markov source parameters from sparse data,'' in *Pattern Recognition in Practice,* E. S. Gelsema and L. N Kanal, Eds. Amsterdam: North-Holland, 1980, pp. 381-397.

[7] M. Cohen, ''Phonological Structures for Speech Recognition,'' PhD thesis, Computer Science Department, UC Berkeley, April 1989.

[8] D. Rumelhart, G. Hinton, and R. Williams, ''Learning Internal Representations by Error Propagation,'' in *Parallel Distributed Processing: Explorations of the Microstructure of Cognition,* vol 1: Foundations, D Rumelhart & J. McClelland, Eds., MIT Press, Cambridge, MA, 1986.

[9] S. Renals, N. Morgan, M. Cohen, H. Franco, ''Connectionist Probability Estimation in the DECIPHER Speech Recognition System,'' *ICASSP 92,* Vol. 1, pp. 601-604, San Francisco, 1992.

[10] R. M. Schwartz, Y. L. Chow, O. A. Kimball, S. Roucos, M. Krasner, and J. Makhoul, ''Context-dependent modeling for acoustic-phonetic recognition of continuous speech,'' *ICASSP 85,* 1205-1208, 1985.

[11] H. Bourlard, N. Morgan, C. Wooters, S Renals, ''CDNN: A Context Dependent Neural Network for Continuous Speech Recognition,'' *ICASSP 92, Vol. 2,* pp. 349-352, San Francisco, 1992.

[12] H. Franco, M. Cohen, N. Morgan, D. Rumelhart, V. Abrash, ''Context-Dependent Connectionist Probability Estimation in a Hybrid HNN-Neural Net Speech Recognition System,'' *IJCNN,* Beijing, 1992.

[13] M. Cohen, H. Franco, N. Morgan, D. Rumelhart, V. Abrash, ''Multiple-State Context-Dependent Phonetic Modeling with MLPs,'' *Speech Research Symposium XII,* Rutgers, 1992.