Context-Dependent Connectionist Probability Estimation in a Hybrid HMM-Neural Net Speech Recognition System

Horacio Franco[†], Michael Cohen[†], Nelson Morgan[‡],
David Rumelhart[§] and Victor Abrash[†]

† SRI International

‡ International Computer Science Institute

§ Stanford University

Running title: Context-Dependent Probability Estimation

Address correspondence to:

Horacio Franco
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025.

Abstract

In this paper we present a training method and a network architecture for estimating context-dependent observation probabilities in the framework of a hybrid hidden Markov model (HMM) / multi layer perceptron (MLP) speaker-independent continuous speech recognition system. The context-dependent modeling approach we present here computes the HMM context-dependent observation probabilities using a Bayesian factorization in terms of context-conditioned posterior phone probabilities which are computed with a set of MLPs, one for every relevant context. The proposed network architecture shares the input-to-hidden layer among the set of context-dependent MLPs in order to reduce the number of independent parameters. Multiple states for phone models with different context dependence for each state are used to model the different context effects at the beginning and end of phonetic segments. A new training procedure that ‘‘smooths’’ networks with different degrees of context-dependence is proposed to obtain a robust estimate of the context-dependent probabilities. We have used this new architecture to model generalized biphone phonetic contexts. Tests with the speaker-independent DARPA Resource Management data base have shown average reductions in word error rates of 28% using a word-pair grammar, compared to our earlier context-independent HMM/MLP hybrid.

1 Introduction

Previous work by Bourlard & Morgan (1990) and Morgan & Bourlard (1990) has shown both theoretically and practically that multilayer perceptrons (MLPs) can be successfully used in a hidden Markov model (HMM) based speech recognition system for estimating the state-dependent observation probabilities. Recently this approach was applied to a state-of-the-art speech recognition system (Renals, Morgan, Cohen & Franco, 1992) in which an MLP provided estimates of context-independent posterior probabilities of phone classes, which were then converted to HMM context-independent state observation probabilities using Bayes rule.

Experience with HMM technology has shown that using context-dependent phonetic models significantly improves recognition accuracy (Schwartz *et al*. 1985). This is so because acoustic correlates of coarticulatory effects are explicitly modeled, 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. A potential problem with this approach is the lack of robustness and poor generalization of the resulting models due to the reduced amount of data available to train them in highly specific contexts. To solve this problem, many HMM systems train models at many different levels of context-specificity, including biphone (conditioned on the phone immediately to the left or right), generalized biphone (conditioned on the broad class of the phone to the left or right), triphone (conditioned on the phone to the left and the right), generalized triphone, and word-specific phone (Lee, 1990) (Murveit *et al*. 1989). Models conditioned by more specific contexts are linearly smoothed with more general models. The “deleted interpolation” algorithm (Jelinek & Mercer, 1980) provides linear weighting coefficients for the observation probabilities with different degrees of context-dependence by maximizing the likelihood of the smoothed models over new, unseen data. This approach is expensive to extend to MLP-based systems because, while the tied mixture weights can be “smoothed” together over different context-dependent models, the linear “smoothing” of MLP weights makes no sense; instead, the outputs of all the context-dependent nets with different degrees of context specificity should be smoothed. In addition, it would make more sense to smooth together discriminant probabilities using a discriminant or error-based procedure.

An earlier approach to context-dependent phonetic modeling with MLPs has been proposed by Bourlard, Morgan, Wooters & Renals (1992). It is based on a factorization of the context-dependent observation probabilities, 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.

The context-dependent modeling approach we present here uses a different factorization of the desired HMM context-dependent observation probabilities, a network architecture consisting of a set of context dependent nets that share the input-to-hidden layer to reduce the number of parameters, multiple states per phone with different context-dependence for each state, and a training procedure that “smooths” networks with different degrees of context-dependence in order to achieve robustness in probability estimates.

2 Hybrid HMM/MLP

The baseline HMM/MLP DECIPHER^{TM} hybrid (described in Renals *et al.* 1992) substitutes (scaled) probability estimates computed with MLPs for the tied mixture HMM state-dependent observation probability densities. The topology of the HMM system is kept unchanged.

The hybrid system is bootstrapped from the basic HMM DECIPHER system (Murveit *et al.* 1989) already trained using the forward-backward maximum likelihood method. Forced Viterbi alignments (to the HMM model sequence corresponding to the known word string) for every training sentence provide phone labels, among 69 phone classes, for every frame of speech.

A feedforward MLP is trained using stochastic gradient descent using these labeled data. The training criterion used is minimum relative entropy between the posterior target distribution and the posterior output distribution. The target distribution is defined as 1 for the index corresponding to the phone class label and 0 for the other classes. With this target distribution, assuming enough parameters in the MLP and enough training data, and assuming that the training does not get stuck in a local minimum, the MLP outputs will approximate the posterior class probabilities $p(q_j|Y_t)$, where q_j corresponds to the *j-th* phone class and Y_t is the acoustic vector at time t (Bourlard *et al.* 1990).

Frame classification error over an independent cross-validation set is used to control the learning rate and to decide when to stop training (as in Renals *et al.* 1992). The initial learning rate is kept constant until cross-validation performance increases less than 0.5%, after that point it is reduced as $1/2^n$ until performance does not increase any further.

The net architecture consists of an input layer of 234 units, spanning nine frames of cepstra, delta cepstra, energy, and delta energy features that are normalized to have zero mean and unit variance. It has a 1000-unit hidden layer and an output layer with 69 units, one per phone class. Both hidden and output layers consists of sigmoidal units.

During recognition, the posterior class probabilities are converted to observation probability densities conditioned on the phone class by using Bayes rule:

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

where $p(Y_t| q_j)$ is the desired observation probability density required by the HMM. The factor $p(q_j)$ is the prior probability of the phone class j, and it is computed by counting over the training data. All the HMM states which are tied to the same context-independent phone class use the same (scaled) MLP output as state-dependent observation probability. As the factor $p(Y_t)$ is constant over all the states for a given time t, it can be assigned any arbitrary value without affecting the optimal path.

Subsequent reestimation of MLP and HMM parameters based on new alignments provided by the new hybrid HMM/MLP (Morgan *et al.* 1990, Franzini, Lee & Waibel, 1990) may improve the performance of the hybrid system.

3 Context dependent HMM/MLP hybrid

The context-independent hybrid HMM/MLP described above has been extended to model context-dependent phonetic classes using a Bayesian factorization in terms of scaled context-dependent posterior phone probabilities computed with a set of context-specific MLPs. Two approaches are used to deal with the increased number of parameters: error-based smoothing of context-dependent and -independent parameters, and sharing of input-to-hidden weights between all context-specific networks. Separate nets are used to model different context effects in initial and final states of HMM phonetic models.

3.1 Context-dependent factorization

In the phonetic-based HMM framework, every state is associated with a specific phone class and context. States associated with the same phone and context are tied together (share common probability distributions). Context-dependent phonetic modeling requires the computation of $p(Y_t|q_j,c_k)$, the probability density of acoustic vector Y_t given the phone class q_j in the context class c_k. Since MLPs can compute Bayesian posterior probabilities, we propose to compute the required HMM probabilities using the following factorization:

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

where $p(Y_t| c_k)$ can be factored again in terms of posteriors as

$$p(Y_t|c_k) = \frac{P(c_k|Y_t)p(Y_t)}{P(c_k)} \qquad (3)$$

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. It can be computed with MLPs in a number of different ways. One possible implementation treats the c_k as M additional binary inputs to a single MLP. During training, only one of the M inputs is set to 1 for each pattern presentation (that input associated with the context class of the training example), and the others are set to 0. Bourlard *et al.* (1992) proposed this type of implementation in the framework of a different factorization of the context-dependent HMM probabilities, and also proposed additional simplifications to the topology of the MLP to reduce the computational load (because during recognition forward propagation has to be computed for every possible value of the context class).

Another possible implementation also uses the 1-of-M binary context inputs but with multiplicative connections that adjust the value of the network weights depending on which context is active. The modulation of weights, in principle, allows the network to have a complete different pattern of connections between features and output units for every different context. McClelland *et al.* (1986) have proposed this architecture in the framework of their TRACE model of speech perception.

An alternative implementation, which we have chosen here, is based on a direct interpretation of the definition of conditional probability, considering the conditioning on c_k in $p(q_j|Y_t,c_k)$ 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 similar to those used in the context-independent case, except that each MLP is trained using only input-output examples obtained when the corresponding context is c_k.

This implementation is appealing because the same network architecture and training method applied to the context-independent case can be applied to every context-specific net, permitting the smoothing scheme and sharing of parameters reported in the following sections. Every context-specific net performs a simpler classification than in the context-independent case because, in a given context, the acoustic correlates of different phones have much less overlap in their class boundaries (which also implies a lower minimum theoretical classification error rate).

The factor $p(c_k|Y_t)$ can be computed using a context-independent MLP whose outputs correspond to the context classes. It is interesting to observe that this MLP must estimate the probability of the context class of the previous or following phone given Y_t. The possibility of defining Y_t to be an extended vector, formed by stacking together several consecutive frames, allows some frames of the actual context phone to be included in the input vector.

The factors $p(q_j|c_k)$ and $p(c_k)$ are constants for a given training set and are estimated by counting over the training examples. Finally, the likelihood $p(Y_t)$ is common to all states for any given time frame, and can therefore be discarded in the computation of the Viterbi algorithm (see Levinson, Rabiner & Sondhi, 1983), since it will not change the optimal state sequence, which determines the recognized string.

3.2 Context-dependent training and smoothing

In order to achieve robust training of the context-specific nets that compute $p(q_j|Y_t,c_k)$, we propose the following method which consists of two stages

In the first stage, a 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 of the set of M context-specific nets.

In the second stage, the context-dependent training proceeds by presenting each training example (the acoustic vector with its associated phone label and context label) only to the corresponding context-specific net. In this stage we are actually training a set of M independent nets, each one trained on a nonoverlapping subset of the original training data. For each context-specific net the training procedure is similar to that used for the context-independent net, using a one-of-N target distribution, stochastic gradient descent, and a minimum relative entropy training criterion. The overall classification performance evaluated on an independent cross-validation set is used to determine a common learning rate using the same heuristics that were used in the context-independent training phase. Training stops when the overall cross-validation performance does not improve further.

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. Because of the initialization, the net starts estimating $p(q_j|Y_t)$, and from that point it follows a trajectory in weight space (see Fig. 1), incrementally moving away from the context-independent parameters so 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. Furthermore, the cross-validation classification error is the criterion that determines how much context-dependent learning is effective for discrimination, so the degree of smoothing is based on the point where cross-validation classification error attains a local minimum.

[FIGURE 1 ABOUT HERE]

Since we start from a good point in the parameter space, training time may be reduced compared to the case of random initialization. Also, because of cross-validation testing, we are guaranteed to perform at least as well as the context-independent net.

This approach can be extended to handle a hierarchy of context-dependent models that go from very broad context classes to highly specific ones. A hierarchy of context classes is defined, in which each context class at one level is included in a broader class at the previous level. Each context-specific MLP at a given level in the hierarchy is initialized with the weights of a previously trained context-specific MLP at the previous level in the hierarchy whose associated context class includes that of the MLP being initialized (see Fig. 2); a finer-context training stage proceeds from these initial parameters.

[FIGURE 2 ABOUT HERE]

3.3 Context-dependent architecture

It is well known that, in a two-layer network, learning the input-to-hidden weights is highly time-consuming. To reduce training time and the number of independent parameters to train, we propose a network architecture in which all the context-specific nets share the input-to-hidden layer (see Fig. 3). Consequently, the hidden layer representation of the acoustic features is shared by all context-specific nets. The different sets of hidden-to-output weights are expected to capture the different acoustic boundaries between phone classes in different contexts.

[FIGURE 3 ABOUT HERE]

As a further simplification to speed-up training, given that the input-to-hidden weights are already trained in the context-independent training phase, we keep them fixed during the context-dependent training phase. The underlying assumption here is that the hidden layer representation of the acoustic features is rich enough to allow accurate modeling of the class boundaries in the different contexts. The only new parameters to train for every context-specific net are the hidden-to-output weights.

3.4 Multiple states for phone models

Experience with HMM-based systems has shown the advantage of modeling phonetic units with a sequence of probability distributions rather than with a single probability distribution. This allows the model to capture some of the dynamics of the phonetic segments. In the SRI DECIPHER^{TM} system, on which the hybrid system is based, a left-to-right two- or three-state model represents each phonetic unit. Multiple-state phone models allow more precise modeling of context effects because the initial portion of a phone segment is more influenced by the previous phone while the final part of a phone segment is more influenced by the following phone. In the present implementation, two different sets of context classes were used: generalized left-biphone dependent for the first state and generalized right-biphone dependent for the final state of each phone model. For the three-state models the middle state was treated as context-independent.

3.5 Recognition

During recognition, first states of HMM phones are associated with the context-specific MLP output according to the context class to which the phone to its left belongs. Last states of HMM phones are associated with the context-specific output according to the context class to which the phone to its right belongs. Middle states of three-state HMM phones are associated with a context-independent layer which was trained only on frames that were aligned to middle HMM phone states.

Recognition itself is accomplished using the Viterbi algorithm, it requires the computation of the observation probabilities associated with each state of the HMM. To this end, the context-dependent posterior probabilities have to be converted to (scaled) state conditioned observation probabilities using the normalization factors provided by Eq. (2) and (3). However, because of the smoothing with the context-independent net, the conversion factors should be a combination of those corresponding to the context-independent and context-dependent cases. We use the following heuristic interpolation scheme for converting the smoothed posteriors $p^s(q_j|Y_t,c_k)$ to smoothed (scaled) observation probabilities $p^s(Y_t|q_j,c_k)$:

$$p^s(Y_t|q_j, c_k) = p^s(q_j|Y_t, c_k)\left(\alpha_j^k \frac{1}{p(q_j)} + (1-\alpha_j^k)\frac{p(c_k|Y_t)}{p(q_j|c_k)p(c_k)}\right) \qquad (4)$$

where

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

$N_{ci}(j)$ is the number of training examples for the phone class j for the context-independent net, and $N_{cd}(j,k)$ is the number of training examples for the phone class j and for the context-specific net corresponding to context class k. The constant b is optimized in a development set for minimum recognition error. This interpolation scheme allows different weighting for the conversion factors given by Eq. (1) -for context-independent training- and Eq. (2) and (3) -for context-dependent training- depending on the corresponding amount of training data for each phone and context class.

4 Experimental Evaluation

Training and recognition experiments with the HMM/MLP hybrid were conducted using the speaker-independent, continuous speech, DARPA Resource Management data base. The vocabulary size is 998 words. A word pair grammar with perplexity 60 or an all-word grammar (perplexity 998) can be used. The training set is composed of 3990 sentences equivalent to about 1.5 million frames. An additional development set of 600 sentences (formed by combining the Feb 89 and Oct 89 test sets) was used for cross-validation testing. A set of 69 phone (and subphone) classes is defined for the labeling of the database. The context classes were defined to be a set of eight left- and eight right-generalized biphone phonetic contexts. The phones belonging to each class were chosen primarily on the basis of place of articulation and gross acoustical characteristics.

The acoustic analysis consisted of a mel cepstrum computed every 10 ms. using overlapping windows of 25 ms., four acoustic features were computed resulting in 26 coefficients produced per frame: 12 cepstral coefficients, normalized cepstral energy, and their smoothed derivatives. For the context-dependent net which estimates $p(q_j|Y_t,c_k)$, a nine-frame window of 234 input values was presented as the input vector Y_t to the input layer. The phone class label associated with the central frame defined the target output class. The context class to which the previous or following phone belongs (depending on which phone state the frame was aligned with) determined the context class index. A hidden layer size of 1000 units was used. The size of the context-dependent net was about 1.4 million weights.

Training of the context-dependent net consisted of first training a context-independent net, which estimates $p(q_j|Y_t)$. Context-independent training took about 5 passes through the database to converge. Then this net’s weights were used to initialize the context-dependent net. Context-dependent training took about eight passes through the data base to converge. The final cross-validation error for the context-dependent net was 21.4% vs. 30.6% obtained with the context-independent network. Expecting this degree of improvement in actual recognition performance may be overoptimistic because the context and segmentation are assumed to be known for this cross-validation error evaluation. Nevertheless, it suggested that the context-dependent architecture was capable of much more detailed modeling of the acoustic variability of the speech signal.

The computational load for context-dependent training was approximately the same as for the context-independent training because, although the context-dependent net is significantly larger than the context-independent one, only the corresponding context-specific output layer is updated for each frame presentation. During recognition, the computational load for the context-dependent hybrid was more than four times that of the context-independent one. This is so because of the huge number of hypotheses that are explored in the Viterbi search, since forward propagation is computed for every context-specific output layer for every frame.

Two additional context-independent networks were trained to provide the posterior probabilities $p(c_k|Y_t)$ for the left and the right context classes. The input to the left context net was formed with 13 frames preceding the center frame, while the right context net used as input 13 frames following the center frame. For both nets, the hidden layer had 1000 hidden units, while the output layer consisted of 8 output units, one for each context class. The use of different input unit layer sizes does not invalidate the use of Eq. (2) and (3) because Y_t can be considered an extended vector including all the frames used in the different input layers; then, in each net --associated with a posterior probability factor-- it is possible to reduce the actual size of the input layer by assuming independence of the corresponding outputs relative to some input frames.

The development set of 600 sentences (Feb 89 plus Oct 89 releases) was also used for tuning parameters such as word transition penalties (see Murveit *et al*. 1989) and the constant b in Eq. (5). We found that the proposed heuristic [Eq. (4)] for combining the scaling factors was better than using either the pure context-dependent or the pure context-independent scalings by them-selves.

In Table I recognition error is presented over three test sets of 300 sentences each (called Feb 91, Sept 92-ind and Sept 92-dep) using the word-pair grammar. Performance is shown for three systems: the context-independent (CI MLP) and the context-dependent (CD MLP) HMM/MLP hybrids, and the context-dependent pure HMM system (CD HMM). In Table II the same test sets are presented using the all-word grammar. In Table III the number of parameters for each system is shown. The HMM topology and transition probabilities were the same over the three systems; the differences were only in the estimation of the state-dependent observation probabilities. The context-independent system used the same net that was used for initializing the context-dependent MLP. The pure HMM was a gender-independent version of the state-of-the-art SRI DECIPHER^{TM} system which used tied mixtures for the state-dependent observation probabilities.

[TABLE I, II, III about here]

We consistently found significant improvements going from the context-independent to the context-dependent hybrid systems over all test sets; the average reductions in word error rates being 28.16% using the word-pair grammar and 19.2% using the all-word grammar. Combining all test sets, the differences between the context-independent and the context-dependent hybrid systems were statistically significant at the 0.95 level in all cases. Comparing the context-dependent hybrid with the HMM, we see that the average performance of the hybrid was slightly better than that of the HMM, showing 10% reduction of the error rate in the grammar case and 6% reduction in the no-grammar case. While in the no-grammar case, these improvements were statistically significant at the 95% level, in the grammar case they were not statistically significant at the 95% level. Interestingly, this performance was achieved using a much simpler context-dependent system that models only generalized biphone classes and uses roughly one-fourth the number of parameters of the pure HMM.

5 Discussion

A number of questions had to be resolved to develop an effective context-dependent hybrid HMM/MLP system:

• *How to obtain context-dependent observation probabilities in terms of posterior probabilities as computed by the MLPs*. The factorization given by Eq. (2) and (3) is one possibility, permitting smoothing of the context-dependent MLPs with a context-independent MLP, guaranteeing a performance that is at least as good as that of the latter net.

• *How to smooth context-dependent nets with context-independent nets in order to obtain robust probability estimates*. This is probably the crucial issue in context-dependent modeling. The initialization of context-specific nets with context-independent weights combined with the use of cross-validation performance to control the degree of smoothing allows the training of a large number of parameters without losing robustness.

• *How to reduce the number of parameters*. The sharing of the pretrained input-to-hidden layer among context-specific nets substantially reduces the number of parameters and the amount of computation during training and recognition. The important decrease in cross-validation error going from context-independent to context-dependent MLPs 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, are useful to capture the more detailed context-specific phone classes. Other parameter-reduction schemes, possibly incorporating the use of binary inputs in addition to the multiple output layer architecture, are also currently under investigation.

• *How to use discrimination in context-dependent modeling with MLPs*. In the proposed architecture and associated training method, discrimination is applied only among output units associated to phone classes in a given context, but it is not applied among output units associated with the same phone in different contexts, because these units belong to different context-specific nets. Such would not have been the case if we had defined an architecture with MLP outputs for every phone-in-context. This is an open issue for further research, and is also related to the following topic.

• *How to use multiple-state HMM phone models with MLP probability estimation*. In the proposed scheme we use separate output layers --corresponding to different context classes-- for different states of an HMM phone model. A previous attempt to use multiple-state phones in a hybrid HMM/MLP system (Renals *et al.* 1992) used a single MLP with multiple outputs per phone, one for every state. This approach did not improve over the single-output-per-phone case. We speculate that the lack of improvement was due to the discriminative nature of the training, because we are attempting to discriminate into different classes acoustic vectors that correspond to the same phone and are probably very similar but which are aligned with different states. The appropriate level at which to train discriminatively is likely to be the phone level rather than the HMM-state level.

7 Conclusions

We have shown a way to do context-dependent phone modeling with MLPs. The resulting context-dependent MLP/HMM hybrid performed significantly better than the context-independent MLP/HMM hybrid, using only very general context classes.

The proposed context-dependent MLP architecture that shares the input-to-hidden layer parameters was able to capture context-specific features using the extra modeling power in the hidden-to-output weights.

The proposed training method was effective for robust training of a context-dependent MLP with a large number of parameters by combining context-independent and context-dependent training based on a cross-validation error criterion.

Acknowledgments

Partial support for this research project was provided by DARPA contract MDA904-90-C-5253. Horacio Franco was partially supported by a fellowship from CONICET, República Argentina. Thanks to Herve Bourlard and Steve Renals for useful discussions.

References

Bourlard, H. & Morgan, N. (1990). Merging Multilayer Perceptrons and Hidden Markov Models: Some Experiments in Continuous Speech Recognition. In Neural Networks: Advances and Applications, (E. Gelenbe, ed.), pp. 215-239. Elsevier Science Publishers B. V., North-Holland, Amsterdam.

Bourlard, H., Morgan, N., Wooters, C., Renals, S. (1992). CDNN: A Context Dependent Neural Network for Continuous Speech Recognition. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing 1992, Vol. 2, pp. 349-352, San Francisco.

Franzini, M., Lee, K. & Waibel, A. (1990). Connectionist Viterbi training: A new hybrid method for Continuous Speech Recognition. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing 1990, Vol.1, pp. 425-428, Alburquerque.

Jelinek, F. & Mercer, R. (1980). Interpolated estimation of Markov source parameters from sparse data. In Pattern Recognition in Practice, (E. S. Gelsema & L. N Kanal, ed), pp. 381-397, North-Holland, Amsterdam.

Lee, K. (1990). Context-Dependent Phonetic Hidden Markov Models for Speaker-Independent Continuous Speech Recognition, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 38, No. 4, pp. 599-609.

Levinson, S., Rabiner, L. & Sondhi, M. (1983). An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. Bell Systems Technical Journal 62, pp. 1035-1074.

McClelland, J. L. & Elman, J. L. (1986). Interactive Processes in Speech Perception: The TRACE Model. In Parallel Distributed Processing, Explorations in the Microstructure of Cognition, (D. Rumelhart and J. McClelland ed.) Vol. 2, pp. 58-121, MIT Press, Cambridge.

Morgan, N. & 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 1990, pp. 413-416, Alburquerque.

Murveit, H., Cohen, M., Price, P., Baldwin, G., Weintraub, M., & Bernstein, J. (1989). SRI's DECIPHER System. In Proceedings of the DARPA Speech and Natural Language Workshop 1989, pp. 238-242, Philadelphia.

Renals, S., Morgan, N., Cohen, M. & Franco, H. (1992). Connectionist Probability Estimation in the DECIPHER Speech Recognition System. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing 1992, Vol. 1, pp. 601-604, San Francisco.

Schwartz, R. M., Chow, Y. L., Kimball, O. A., Roucos S., Krasner, M. & Makhoul, J. (1985). Context-dependent modeling for acoustic-phonetic recognition of continuous speech. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing 1985, pp. 1205-1208, Tampa.

Table I: WORD PAIR GRAMMAR

	CI MLP	CD MLP	CD HMM
Feb 91	5.8	4.7	3.8
Sept 92 ind	10.9	7.6	10.1
Sept 92 dep	9.5	6.6	7.0
Overall test	8.77	6.3	7.0

Word recognition error rate over three testing sets for three recognition systems: context-independent hybrid (CI MLP), context-dependent hybrid (CD MLP) and pure HMM (CD HMM) using the word pair grammar.

Table II: NO GRAMMAR

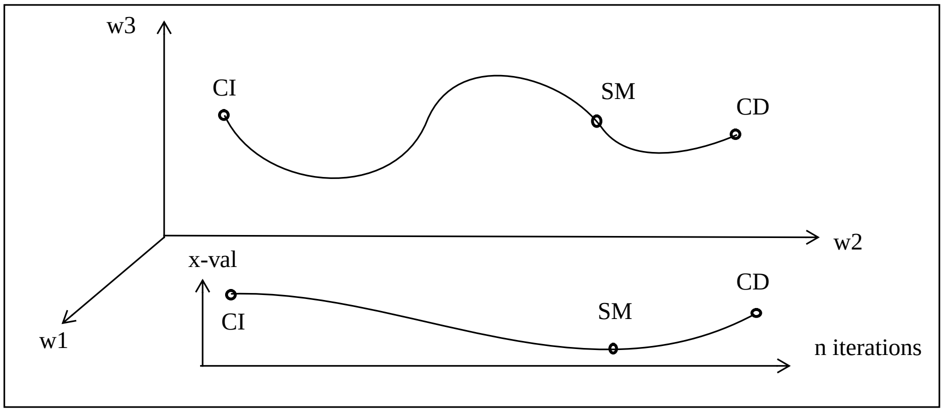
	CI MLP	CD MLP	CD HMM
Feb 91	24.7	18.4	19.3
Sep 92 ind	31.5	27.1	29.2
Sep 92 dep	30.9	24.9	26.6
Overall test	29.1	23.5	25.1

Word recognition error rate over three testing sets for three recognition systems: context-independent hybrid (CI MLP), context-dependent hybrid (CD MLP) and pure HMM (CD HMM) using the all word grammar.

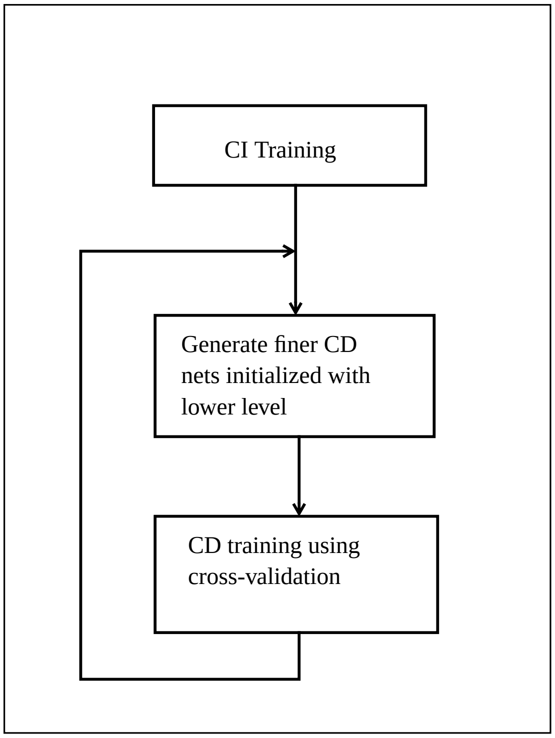
TABLE III: NUMBER OF PARAMETERS

	CI MLP	CD MLP	CD HMM
# Params	300K	1400K	5500K

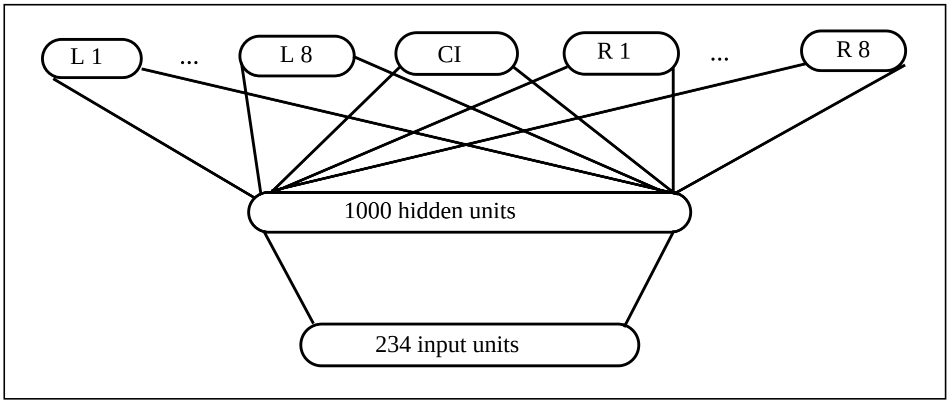
Number of parameters for the three recognition systems.



Horacio Franco *et al.* Fig. 1


CI Training
Generate finer CD nets initialized with lower level
CD training using cross-validation

Horacio Franco *et al.* Fig. 2



Horacio Franco *et al.* Fig. 3.

List of figure captions:

Fig. 1: Pictorial representation of context-dependent training evolution in weight space and associated cross-validation classification error. CI, CD and SM refer to the context-independent, dependent and “smoothed” nets, respectively.

Fig. 2: Extension of context-dependent training to handle a hierarchy of context classes.

Fig. 3: Context-dependent MLP architecture.

Context-Dependent Connectionist Probability Estimation in a Hybrid HMM-Neural Net Speech Recognition System

Horacio Franco[†], Michael Cohen[†], Nelson Morgan[‡],
David Rumelhart[§] and Victor Abrash[†]

† SRI International

‡ International Computer Science Institute

§ Stanford University

Running title: Context-Dependent Probability Estimation

Address correspondence to:

Horacio Franco
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025.

Abstract

In this paper we present a training method and a network architecture for estimating context-dependent observation probabilities in the framework of a hybrid hidden Markov model (HMM) / multi layer perceptron (MLP) speaker-independent continuous speech recognition system. The context-dependent modeling approach we present here computes the HMM context-dependent observation probabilities using a Bayesian factorization in terms of context-conditioned posterior phone probabilities which are computed with a set of MLPs, one for every relevant context. The proposed network architecture shares the input-to-hidden layer among the set of context-dependent MLPs in order to reduce the number of independent parameters. Multiple states for phone models with different context dependence for each state are used to model the different context effects at the beginning and end of phonetic segments. A new training procedure that ‘‘smooths’’ networks with different degrees of context-dependence is proposed to obtain a robust estimate of the context-dependent probabilities. We have used this new architecture to model generalized biphone phonetic contexts. Tests with the speaker-independent DARPA Resource Management data base have shown average reductions in word error rates of 28% using a word-pair grammar, compared to our earlier context-independent HMM/MLP hybrid.

1 Introduction

Previous work by Bourlard & Morgan (1990) and Morgan & Bourlard (1990) has shown both theoretically and practically that multilayer perceptrons (MLPs) can be successfully used in a hidden Markov model (HMM) based speech recognition system for estimating the state-dependent observation probabilities. Recently this approach was applied to a state-of-the-art speech recognition system (Renals, Morgan, Cohen & Franco, 1992) in which an MLP provided estimates of context-independent posterior probabilities of phone classes, which were then converted to HMM context-independent state observation probabilities using Bayes rule.

Experience with HMM technology has shown that using context-dependent phonetic models significantly improves recognition accuracy (Schwartz *et al*. 1985). This is so because acoustic correlates of coarticulatory effects are explicitly modeled, 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. A potential problem with this approach is the lack of robustness and poor generalization of the resulting models due to the reduced amount of data available to train them in highly specific contexts. To solve this problem, many HMM systems train models at many different levels of context-specificity, including biphone (conditioned on the phone immediately to the left or right), generalized biphone (conditioned on the broad class of the phone to the left or right), triphone (conditioned on the phone to the left and the right), generalized triphone, and word-specific phone (Lee, 1990) (Murveit *et al*. 1989). Models conditioned by more specific contexts are linearly smoothed with more general models. The “deleted interpolation” algorithm (Jelinek & Mercer, 1980) provides linear weighting coefficients for the observation probabilities with different degrees of context-dependence by maximizing the likelihood of the smoothed models over new, unseen data. This approach is expensive to extend to MLP-based systems because, while the tied mixture weights can be “smoothed” together over different context-dependent models, the linear "smoothing" of MLP weights makes no sense; instead, the outputs of all the context-dependent nets with different degrees of context specificity should be smoothed. In addition, it would make more sense to smooth together discriminant probabilities using a discriminant or error-based procedure.

An earlier approach to context-dependent phonetic modeling with MLPs has been proposed by Bourlard, Morgan, Wooters & Renals (1992). It is based on a factorization of the context-dependent observation probabilities, 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.

The context-dependent modeling approach we present here uses a different factorization of the desired HMM context-dependent observation probabilities, a network architecture consisting of a set of context dependent nets that share the input-to-hidden layer to reduce the number of parameters, multiple states per phone with different context-dependence for each state, and a training procedure that "smooths" networks with different degrees of context-dependence in order to achieve robustness in probability estimates.

2 Hybrid HMM/MLP

The baseline HMM/MLP DECIPHER^{TM} hybrid (described in Renals *et al.* 1992) substitutes (scaled) probability estimates computed with MLPs for the tied mixture HMM state-dependent observation probability densities. The topology of the HMM system is kept unchanged.

The hybrid system is bootstrapped from the basic HMM DECIPHER system (Murveit *et al.* 1989) already trained using the forward-backward maximum likelihood method. Forced Viterbi alignments (to the HMM model sequence corresponding to the known word string) for every training sentence provide phone labels, among 69 phone classes, for every frame of speech.

A feedforward MLP is trained using stochastic gradient descent using these labeled data. The training criterion used is minimum relative entropy between the posterior target distribution and the posterior output distribution. The target distribution is defined as 1 for the index corresponding to the phone class label and 0 for the other classes. With this target distribution, assuming enough parameters in the MLP and enough training data, and assuming that the training does not get stuck in a local minimum, the MLP outputs will approximate the posterior class probabilities $p(q_j|Y_t)$, where q_j corresponds to the *j-th* phone class and Y_t is the acoustic vector at time *t* (Bourlard *et al.* 1990).

Frame classification error over an independent cross-validation set is used to control the learning rate and to decide when to stop training (as in Renals *et al.* 1992). The initial learning rate is kept constant until cross-validation performance increases less than 0.5%, after that point it is reduced as $1/2^n$ until performance does not increase any further.

The net architecture consists of an input layer of 234 units, spanning nine frames of cepstra, delta cepstra, energy, and delta energy features that are normalized to have zero mean and unit variance. It has a 1000-unit hidden layer and an output layer with 69 units, one per phone class. Both hidden and output layers consists of sigmoidal units.

During recognition, the posterior class probabilities are converted to observation probability densities conditioned on the phone class by using Bayes rule:

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

where $p(Y_t| q_j)$ is the desired observation probability density required by the HMM. The factor $p(q_j)$ is the prior probability of the phone class *j*, and it is computed by counting over the training data. All the HMM states which are tied to the same context-independent phone class use the same (scaled) MLP output as state-dependent observation probability. As the factor $p(Y_t)$ is constant over all the states for a given time *t*, it can be assigned any arbitrary value without affecting the optimal path.

Subsequent reestimation of MLP and HMM parameters based on new alignments provided by the new hybrid HMM/MLP (Morgan *et al.* 1990, Franzini, Lee & Waibel, 1990) may improve the performance of the hybrid system.

3 Context dependent HMM/MLP hybrid

The context-independent hybrid HMM/MLP described above has been extended to model context-dependent phonetic classes using a Bayesian factorization in terms of scaled context-dependent posterior phone probabilities computed with a set of context-specific MLPs. Two approaches are used to deal with the increased number of parameters: error-based smoothing of context-dependent and -independent parameters, and sharing of input-to-hidden weights between all context-specific networks. Separate nets are used to model different context effects in initial and final states of HMM phonetic models.

3.1 Context-dependent factorization

In the phonetic-based HMM framework, every state is associated with a specific phone class and context. States associated with the same phone and context are tied together (share common probability distributions). Context-dependent phonetic modeling requires the computation of $p(Y_t|q_j,c_k)$, the probability density of acoustic vector Y_t given the phone class q_j in the context class c_k. Since MLPs can compute Bayesian posterior probabilities, we propose to compute the required HMM probabilities using the following factorization:

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

where $p(Y_t| c_k)$ can be factored again in terms of posteriors as

$$p(Y_t|c_k) = \frac{P(c_k|Y_t)p(Y_t)}{P(c_k)} \qquad (3)$$

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. It can be computed with MLPs in a number of different ways. One possible implementation treats the c_k as M additional binary inputs to a single MLP. During training, only one of the M inputs is set to 1 for each pattern presentation (that input associated with the context class of the training example), and the others are set to 0. Bourlard *et al.* (1992) proposed this type of implementation in the framework of a different factorization of the context-dependent HMM probabilities, and also proposed additional simplifications to the topology of the MLP to reduce the computational load (because during recognition forward propagation has to be computed for every possible value of the context class).

Another possible implementation also uses the 1-of-M binary context inputs but with multiplicative connections that adjust the value of the network weights depending on which context is active. The modulation of weights, in principle, allows the network to have a complete different pattern of connections between features and output units for every different context. McClelland *et al.* (1986) have proposed this architecture in the framework of their TRACE model of speech perception.

An alternative implementation, which we have chosen here, is based on a direct interpretation of the definition of conditional probability, considering the conditioning on c_k in $p(q_j|Y_t,c_k)$ 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 similar to those used in the context-independent case, except that each MLP is trained using only input-output examples obtained when the corresponding context is c_k.

This implementation is appealing because the same network architecture and training method applied to the context-independent case can be applied to every context-specific net, permitting the smoothing scheme and sharing of parameters reported in the following sections. Every context-specific net performs a simpler classification than in the context-independent case because, in a given context, the acoustic correlates of different phones have much less overlap in their class boundaries (which also implies a lower minimum theoretical classification error rate).

The factor $p(c_k|Y_t)$ can be computed using a context-independent MLP whose outputs correspond to the context classes. It is interesting to observe that this MLP must estimate the probability of the context class of the previous or following phone given Y_t. The possibility of defining Y_t to be an extended vector, formed by stacking together several consecutive frames, allows some frames of the actual context phone to be included in the input vector.

The factors $p(q_j|c_k)$ and $p(c_k)$ are constants for a given training set and are estimated by counting over the training examples. Finally, the likelihood $p(Y_t)$ is common to all states for any given time frame, and can therefore be discarded in the computation of the Viterbi algorithm (see Levinson, Rabiner & Sondhi, 1983), since it will not change the optimal state sequence, which determines the recognized string.

3.2 Context-dependent training and smoothing

In order to achieve robust training of the context-specific nets that compute $p(q_j|Y_t,c_k)$, we propose the following method which consists of two stages

In the first stage, a 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 of the set of M context-specific nets.

In the second stage, the context-dependent training proceeds by presenting each training example (the acoustic vector with its associated phone label and context label) only to the corresponding context-specific net. In this stage we are actually training a set of M independent nets, each one trained on a nonoverlapping subset of the original training data. For each context-specific net the training procedure is similar to that used for the context-independent net, using a one-of-N target distribution, stochastic gradient descent, and a minimum relative entropy training criterion. The overall classification performance evaluated on an independent cross-validation set is used to determine a common learning rate using the same heuristics that were used in the context-independent training phase. Training stops when the overall cross-validation performance does not improve further.

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. Because of the initialization, the net starts estimating $p(q_j|Y_t)$, and from that point it follows a trajectory in weight space (see Fig. 1), incrementally moving away from the context-independent parameters so 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. Furthermore, the cross-validation classification error is the criterion that determines how much context-dependent learning is effective for discrimination, so the degree of smoothing is based on the point where cross-validation classification error attains a local minimum.

[FIGURE 1 ABOUT HERE]

Since we start from a good point in the parameter space, training time may be reduced compared to the case of random initialization. Also, because of cross-validation testing, we are guaranteed to perform at least as well as the context-independent net.

This approach can be extended to handle a hierarchy of context-dependent models that go from very broad context classes to highly specific ones. A hierarchy of context classes is defined, in which each context class at one level is included in a broader class at the previous level. Each context-specific MLP at a given level in the hierarchy is initialized with the weights of a previously trained context-specific MLP at the previous level in the hierarchy whose associated context class includes that of the MLP being initialized (see Fig. 2); a finer-context training stage proceeds from these initial parameters.

[FIGURE 2 ABOUT HERE]

3.3 Context-dependent architecture

It is well known that, in a two-layer network, learning the input-to-hidden weights is highly time-consuming. To reduce training time and the number of independent parameters to train, we propose a network architecture in which all the context-specific nets share the input-to-hidden layer (see Fig. 3). Consequently, the hidden layer representation of the acoustic features is shared by all context-specific nets. The different sets of hidden-to-output weights are expected to capture the different acoustic boundaries between phone classes in different contexts.

[FIGURE 3 ABOUT HERE]

As a further simplification to speed-up training, given that the input-to-hidden weights are already trained in the context-independent training phase, we keep them fixed during the context-dependent training phase. The underlying assumption here is that the hidden layer representation of the acoustic features is rich enough to allow accurate modeling of the class boundaries in the different contexts. The only new parameters to train for every context-specific net are the hidden-to-output weights.

3.4 Multiple states for phone models

Experience with HMM-based systems has shown the advantage of modeling phonetic units with a sequence of probability distributions rather than with a single probability distribution. This allows the model to capture some of the dynamics of the phonetic segments. In the SRI DECIPHER^{TM} system, on which the hybrid system is based, a left-to-right two- or three-state model represents each phonetic unit. Multiple-state phone models allow more precise modeling of context effects because the initial portion of a phone segment is more influenced by the previous phone while the final part of a phone segment is more influenced by the following phone. In the present implementation, two different sets of context classes were used: generalized left-biphone dependent for the first state and generalized right-biphone dependent for the final state of each phone model. For the three-state models the middle state was treated as context-independent.

3.5 Recognition

During recognition, first states of HMM phones are associated with the context-specific MLP output according to the context class to which the phone to its left belongs. Last states of HMM phones are associated with the context-specific output according to the context class to which the phone to its right belongs. Middle states of three-state HMM phones are associated with a context-independent layer which was trained only on frames that were aligned to middle HMM phone states.

Recognition itself is accomplished using the Viterbi algorithm, it requires the computation of the observation probabilities associated with each state of the HMM. To this end, the context-dependent posterior probabilities have to be converted to (scaled) state conditioned observation probabilities using the normalization factors provided by Eq. (2) and (3). However, because of the smoothing with the context-independent net, the conversion factors should be a combination of those corresponding to the context-independent and context-dependent cases. We use the following heuristic interpolation scheme for converting the smoothed posteriors $p^s(q_j|Y_t,c_k)$ to smoothed (scaled) observation probabilities $p^s(Y_t|q_j,c_k)$:

$$p^s(Y_t|q_j, c_k) = p^s(q_j|Y_t, c_k)\left(\alpha_j^k\frac{1}{p(q_j)} + (1-\alpha_j^k)\frac{p(c_k|Y_t)}{p(q_j|c_k)p(c_k)}\right) \tag{4}$$

where

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

$N_{ci}(j)$ is the number of training examples for the phone class j for the context-independent net, and $N_{cd}(j,k)$ is the number of training examples for the phone class j and for the context-specific net corresponding to context class k. The constant b is optimized in a development set for minimum recognition error. This interpolation scheme allows different weighting for the conversion factors given by Eq. (1) -for context-independent training- and Eq. (2) and (3) -for context-dependent training- depending on the corresponding amount of training data for each phone and context class.

4 Experimental Evaluation

Training and recognition experiments with the HMM/MLP hybrid were conducted using the speaker-independent, continuous speech, DARPA Resource Management data base. The vocabulary size is 998 words. A word pair grammar with perplexity 60 or an all-word grammar (perplexity 998) can be used. The training set is composed of 3990 sentences equivalent to about 1.5 million frames. An additional development set of 600 sentences (formed by combining the Feb 89 and Oct 89 test sets) was used for cross-validation testing. A set of 69 phone (and subphone) classes is defined for the labeling of the database. The context classes were defined to be a set of eight left- and eight right-generalized biphone phonetic contexts. The phones belonging to each class were chosen primarily on the basis of place of articulation and gross acoustical characteristics.

The acoustic analysis consisted of a mel cepstrum computed every 10 ms. using overlapping windows of 25 ms., four acoustic features were computed resulting in 26 coefficients produced per frame: 12 cepstral coefficients, normalized cepstral energy, and their smoothed derivatives. For the context-dependent net which estimates $p(q_j|Y_t,c_k)$, a nine-frame window of 234 input values was presented as the input vector Y_t to the input layer. The phone class label associated with the central frame defined the target output class. The context class to which the previous or following phone belongs (depending on which phone state the frame was aligned with) determined the context class index. A hidden layer size of 1000 units was used. The size of the context-dependent net was about 1.4 million weights.

Training of the context-dependent net consisted of first training a context-independent net, which estimates $p(q_j|Y_t)$. Context-independent training took about 5 passes through the database to converge. Then this net's weights were used to initialize the context-dependent net. Context-dependent training took about eight passes through the data base to converge. The final cross-validation error for the context-dependent net was 21.4% vs. 30.6% obtained with the context-independent network. Expecting this degree of improvement in actual recognition performance may be overoptimistic because the context and segmentation are assumed to be known for this cross-validation error evaluation. Nevertheless, it suggested that the context-dependent architecture was capable of much more detailed modeling of the acoustic variability of the speech signal.

The computational load for context-dependent training was approximately the same as for the context-independent training because, although the context-dependent net is significantly larger than the context-independent one, only the corresponding context-specific output layer is updated for each frame presentation. During recognition, the computational load for the context-dependent hybrid was more than four times that of the context-independent one. This is so because of the huge number of hypotheses that are explored in the Viterbi search, since forward propagation is computed for every context-specific output layer for every frame.

Two additional context-independent networks were trained to provide the posterior probabilities $p(c_k|Y_t)$ for the left and the right context classes. The input to the left context net was formed with 13 frames preceding the center frame, while the right context net used as input 13 frames following the center frame. For both nets, the hidden layer had 1000 hidden units, while the output layer consisted of 8 output units, one for each context class. The use of different input unit layer sizes does not invalidate the use of Eq. (2) and (3) because Y_t can be considered an extended vector including all the frames used in the different input layers; then, in each net --associated with a posterior probability factor-- it is possible to reduce the actual size of the input layer by assuming independence of the corresponding outputs relative to some input frames.

The development set of 600 sentences (Feb 89 plus Oct 89 releases) was also used for tuning parameters such as word transition penalties (see Murveit *et al*. 1989) and the constant b in Eq. (5). We found that the proposed heuristic [Eq. (4)] for combining the scaling factors was better than using either the pure context-dependent or the pure context-independent scalings by them-selves.

In Table I recognition error is presented over three test sets of 300 sentences each (called Feb 91, Sept 92-ind and Sept 92-dep) using the word-pair grammar. Performance is shown for three systems: the context-independent (CI MLP) and the context-dependent (CD MLP) HMM/MLP hybrids, and the context-dependent pure HMM system (CD HMM). In Table II the same test sets are presented using the all-word grammar. In Table III the number of parameters for each system is shown. The HMM topology and transition probabilities were the same over the three systems; the differences were only in the estimation of the state-dependent observation probabilities. The context-independent system used the same net that was used for initializing the context-dependent MLP. The pure HMM was a gender-independent version of the state-of-the-art SRI DECIPHER^{TM} system which used tied mixtures for the state-dependent observation probabilities.

[TABLE I, II, III about here]

We consistently found significant improvements going from the context-independent to the context-dependent hybrid systems over all test sets; the average reductions in word error rates being 28.16% using the word-pair grammar and 19.2% using the all-word grammar. Combining all test sets, the differences between the context-independent and the context-dependent hybrid systems were statistically significant at the 0.95 level in all cases. Comparing the context-dependent hybrid with the HMM, we see that the average performance of the hybrid was slightly better than that of the HMM, showing 10% reduction of the error rate in the grammar case and 6% reduction in the no-grammar case. While in the no-grammar case, these improvements were statistically significant at the 95% level, in the grammar case they were not statistically significant at the 95% level. Interestingly, this performance was achieved using a much simpler context-dependent system that models only generalized biphone classes and uses roughly one-fourth the number of parameters of the pure HMM.

5 Discussion

A number of questions had to be resolved to develop an effective context-dependent hybrid HMM/MLP system:

• *How to obtain context-dependent observation probabilities in terms of posterior probabilities as computed by the MLPs*. The factorization given by Eq. (2) and (3) is one possibility, permitting smoothing of the context-dependent MLPs with a context-independent MLP, guaranteeing a performance that is at least as good as that of the latter net.

• *How to smooth context-dependent nets with context-independent nets in order to obtain robust probability estimates*. This is probably the crucial issue in context-dependent modeling. The initialization of context-specific nets with context-independent weights combined with the use of cross-validation performance to control the degree of smoothing allows the training of a large number of parameters without losing robustness.

• *How to reduce the number of parameters*. The sharing of the pretrained input-to-hidden layer among context-specific nets substantially reduces the number of parameters and the amount of computation during training and recognition. The important decrease in cross-validation error going from context-independent to context-dependent MLPs 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, are useful to capture the more detailed context-specific phone classes. Other parameter-reduction schemes, possibly incorporating the use of binary inputs in addition to the multiple output layer architecture, are also currently under investigation.

• *How to use discrimination in context-dependent modeling with MLPs*. In the proposed architecture and associated training method, discrimination is applied only among output units associated to phone classes in a given context, but it is not applied among output units associated with the same phone in different contexts, because these units belong to different context-specific nets. Such would not have been the case if we had defined an architecture with MLP outputs for every phone-in-context. This is an open issue for further research, and is also related to the following topic.

• *How to use multiple-state HMM phone models with MLP probability estimation*. In the proposed scheme we use separate output layers --corresponding to different context classes-- for different states of an HMM phone model. A previous attempt to use multiple-state phones in a hybrid HMM/MLP system (Renals *et al.* 1992) used a single MLP with multiple outputs per phone, one for every state. This approach did not improve over the single-output-per-phone case. We speculate that the lack of improvement was due to the discriminative nature of the training, because we are attempting to discriminate into different classes acoustic vectors that correspond to the same phone and are probably very similar but which are aligned with different states. The appropriate level at which to train discriminatively is likely to be the phone level rather than the HMM-state level.

7 Conclusions

We have shown a way to do context-dependent phone modeling with MLPs. The resulting context-dependent MLP/HMM hybrid performed significantly better than the context-independent MLP/HMM hybrid, using only very general context classes.

The proposed context-dependent MLP architecture that shares the input-to-hidden layer parameters was able to capture context-specific features using the extra modeling power in the hidden-to-output weights.

The proposed training method was effective for robust training of a context-dependent MLP with a large number of parameters by combining context-independent and context-dependent training based on a cross-validation error criterion.

Acknowledgments

Partial support for this research project was provided by DARPA contract MDA904-90-C-5253. Horacio Franco was partially supported by a fellowship from CONICET, República Argentina. Thanks to Herve Bourlard and Steve Renals for useful discussions.

References

Bourlard, H. & Morgan, N. (1990). Merging Multilayer Perceptrons and Hidden Markov Models: Some Experiments in Continuous Speech Recognition. In Neural Networks: Advances and Applications, (E. Gelenbe, ed.), pp. 215-239. Elsevier Science Publishers B. V., North-Holland, Amsterdam.

Bourlard, H., Morgan, N., Wooters, C., Renals, S. (1992). CDNN: A Context Dependent Neural Network for Continuous Speech Recognition. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing 1992, Vol. 2, pp. 349-352, San Francisco.

Franzini, M., Lee, K. & Waibel, A. (1990). Connectionist Viterbi training: A new hybrid method for Continuous Speech Recognition. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing 1990, Vol.1, pp. 425-428, Alburquerque.

Jelinek, F. & Mercer, R. (1980). Interpolated estimation of Markov source parameters from sparse data. In Pattern Recognition in Practice, (E. S. Gelsema & L. N Kanal, ed), pp. 381-397, North-Holland, Amsterdam.

Lee, K. (1990). Context-Dependent Phonetic Hidden Markov Models for Speaker-Independent Continuous Speech Recognition, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 38, No. 4, pp. 599-609.

Levinson, S., Rabiner, L. & Sondhi, M. (1983). An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. Bell Systems Technical Journal 62, pp. 1035-1074.

McClelland, J. L. & Elman, J. L. (1986). Interactive Processes in Speech Perception: The TRACE Model. In Parallel Distributed Processing, Explorations in the Microstructure of Cognition, (D. Rumelhart and J. McClelland ed.) Vol. 2, pp. 58-121, MIT Press, Cambridge.

Morgan, N. & 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 1990, pp. 413-416, Alburquerque.

Murveit, H., Cohen, M., Price, P., Baldwin, G., Weintraub, M., & Bernstein, J. (1989). SRI's DECIPHER System. In Proceedings of the DARPA Speech and Natural Language Workshop 1989, pp. 238-242, Philadelphia.

Renals, S., Morgan, N., Cohen, M. & Franco, H. (1992). Connectionist Probability Estimation in the DECIPHER Speech Recognition System. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing 1992, Vol. 1, pp. 601-604, San Francisco.

Schwartz, R. M., Chow, Y. L., Kimball, O. A., Roucos S., Krasner, M. & Makhoul, J. (1985). Context-dependent modeling for acoustic-phonetic recognition of continuous speech. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing 1985, pp. 1205-1208, Tampa.

Table I: WORD PAIR GRAMMAR

	CI MLP	CD MLP	CD HMM
Feb 91	5.8	4.7	3.8
Sept 92 ind	10.9	7.6	10.1
Sept 92 dep	9.5	6.6	7.0
Overall test	8.77	6.3	7.0

Word recognition error rate over three testing sets for three recognition systems: context-independent hybrid (CI MLP), context-dependent hybrid (CD MLP) and pure HMM (CD HMM) using the word pair grammar.

Table II: NO GRAMMAR

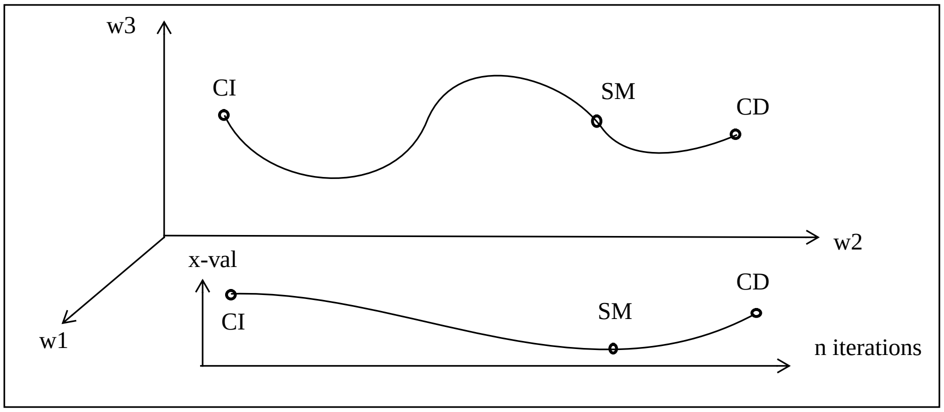
	CI MLP	CD MLP	CD HMM
Feb 91	24.7	18.4	19.3
Sep 92 ind	31.5	27.1	29.2
Sep 92 dep	30.9	24.9	26.6
Overall test	29.1	23.5	25.1

Word recognition error rate over three testing sets for three recognition systems: context-independent hybrid (CI MLP), context-dependent hybrid (CD MLP) and pure HMM (CD HMM) using the all word grammar.

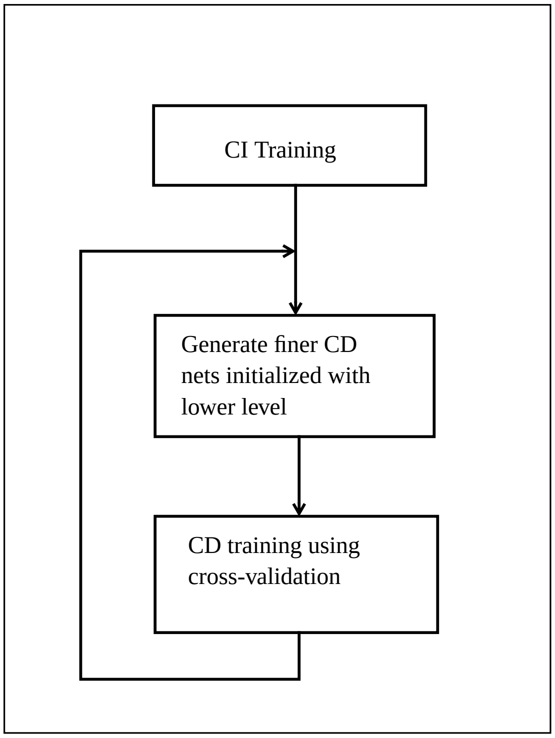
TABLE III: NUMBER OF PARAMETERS

	CI MLP	CD MLP	CD HMM
# Params	300K	1400K	5500K

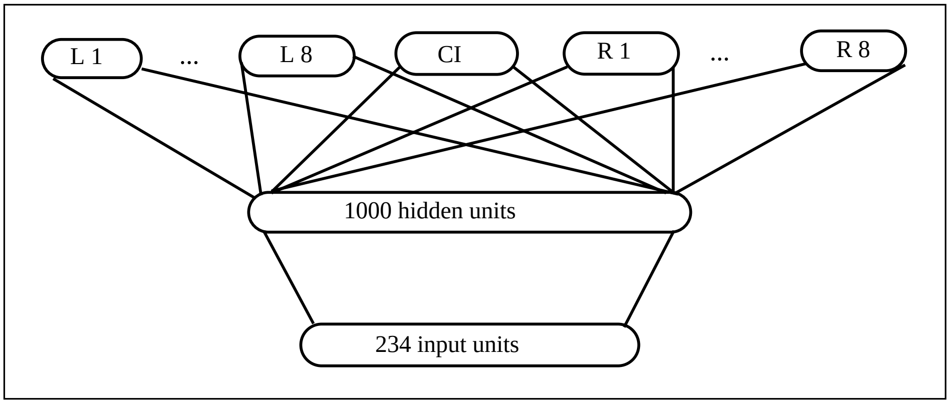
Number of parameters for the three recognition systems.



Horacio Franco *et al.* Fig. 1


CI Training
Generate finer CD nets initialized with lower level
CD training using cross-validation

Horacio Franco *et al.* Fig. 2



Horacio Franco *et al.* Fig. 3.

List of figure captions:

Fig. 1: Pictorial representation of context-dependent training evolution in weight space and associated cross-validation classification error. CI, CD and SM refer to the context-independent, dependent and “smoothed” nets, respectively.

Fig. 2: Extension of context-dependent training to handle a hierarchy of context classes.

Fig. 3: Context-dependent MLP architecture.