# 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, 333 Ravenswood Ave., Menlo Park, CA 94025

‡ Intl. Computer Science Inst., 1947 Center Street, Suite 600, Berkeley, CA 94704

§ Stanford University, Dept. of Psychology, Stanford CA 94305

**Abstract**

In this paper we present a training method and a network achitecture for the estimation of 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 scaled 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 begining and end of phonetic segments. A new training procedure that ''smooths'' networks with different degrees of context-dependence is proposed in order 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 database have shown average reductions in word error rates of 20% in the word-pair grammar case, and 11% in the no-grammar case, compared to our earlier context-independent HMM/MLP hybrid.

## 1 Introduction

Previous work by Morgan, Bourlard, et al. [1, 2] has shown both theoretically and practically that multi layer perceptrons (MLPs) can be successfully used in a hidden Markov model (HMM) system for estimation of the state dependent observation probabilities. Recently this approach was applied to a state of the art speech recognition system [3] in which an MLP provided estimates of context-independent posterior probabilites 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 improves recognition accuracy significantly [4]. 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. This causes problems due to the reduced amount of data available to train phones in highly specific contexts. Thus, the resulting models are not robust and generalize poorly. The solution to this problem used by many HMM systems is to 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 [5,6]. Models conditioned by more specific contexts are linearly smoothed with more general models. The “deleted interpolation” algorithm [7] 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. A more subtle point is that for “smoothing” together discriminant probabilities it is more sensible to use a discriminant procedure.

An earlier approach to context-dependent phonetic modeling with MLPs has been proposed by H. Bourlard et al. [8]. 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 is not much greater than that 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 achitecture that shares the input-to-hidden layer among the context-dependent nets 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 hybrid (described in [3]) substitutes (scaled) probablity 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 [6] already trained using the forward-backward maximum likelihood method. Forced Viterbi alignments 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. A minimum relative entropy between posterior target distribution and posterior output distribution is used. 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, enough training data, and that the training does not get stuck in a local minimum, the MLP outputs will approximate the posterior class probabilities $p(q_j|Y_t)$, where $q_j$ corresponds to the *j-th* phone class and $Y_t$ is the acoustic vector at time $t$ [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 [3]. The initial learning rate is kept constant until cross-validation performance increases less that 0.5%, after which 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 9 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 (scaled) phone class conditioned observation probabilities using Bayes rule.

Subsequent reestimation of MLP and HMM parameters based on new alignments provided by the new hybrid HMM/MLP (as in “connectionist Viterbi training” [9,10]) 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 control the 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 first and last states of HMM phonetic models.

### 3.1 Context-dependent factorization

In the HMM framework, every state is associated with a specific phone class and context. During the Viterbi [11] rec-

ognition search, $p(Y_t|q_j,c_k)$ (the probability density of acoustic vector $Y_t$ given the phone class $q_j$ in the context class $c_k$) is required for each state. Since MLPs can compute Bayesian posterior probabilities, we propose to compute the required HMM probabilities using the following factorization:

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

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

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

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 zero. Bourlard et al. [8] have proposed this type of implementation in the framework of a different factorization of the context dependent HMM probabilities, also proposing 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 allows, in principle, the network to have a complete different pattern of connections between features and output units for every different context. McClelland et al. have proposed this architecture in the framework of their Trace model of speech perception [12].

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 (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 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 $c_k$.

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. 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 (implying a lower minimum theoretical classification error rate).

$p(c_k|Y_t)$ can be computed using a standard MLP whose outputs correspond to the context classes. $p(q_j,|c_k)$ and $p(c_k)$ 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 it can therefore be discarded in the computation of the Viterbi algorithm [11], since it will not change the optimal state sequence used to get the recognized string.

### 3.2 Context-dependent training and smoothing

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

Initially a context-independent MLP is trained as in [3] 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 context-specific nets. The context-dependent training proceeds by presenting each training example (the acoustic vector with an associated phone label and context label) only to the corresponding context specific net. Otherwise, the training procedure is similar to that for the context-independent net, using stochastic gradient descent and a relative entropy training criterion. The overall classification performance evaluated on an independent cross-validation set is used to determine the learning rate as in [3]. Training stops when overall cross-validation performance does not improve anymore. In this phase we are actually training a set of M independent nets, each one trained on a nonoverlapping subset of the original training data.

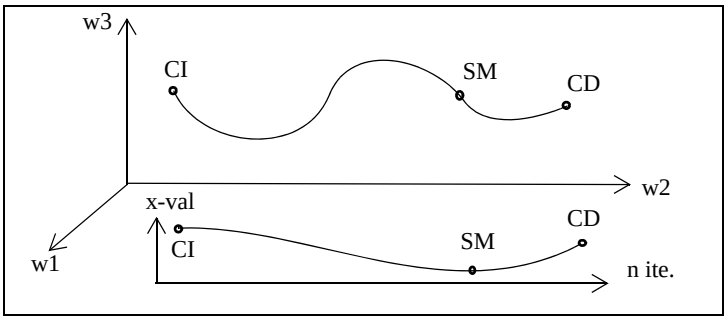


Fig. 1: Pictorial representation of context-dependent training evolution in weight space and associated cross-validation classification error. CI, CD and SM refer to the context independent, dependent and "smoothed" nets.

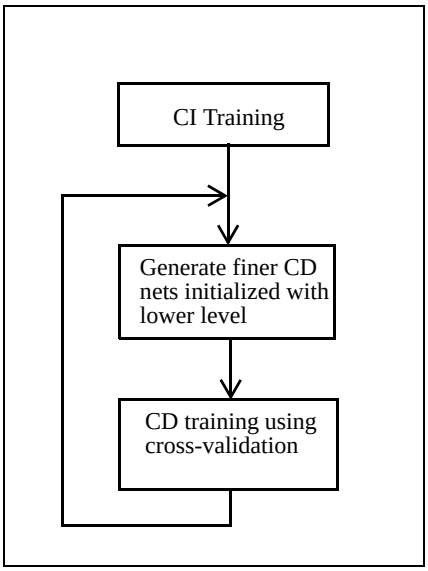


Fig. 2: Extension of context-dependent training to handle a hierarchy of context classes.

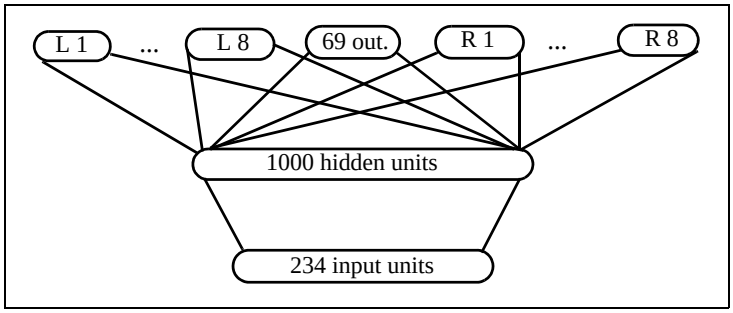


Fig. 3: Context-dependent MLP architecture.

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. Due to 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 non-linear smoothing between the pure context-independent parameters and the pure context-dependent parameters (see Fig. 1).

Furthermore, whereas in the deleted interpolation method the mixing coefficients are determined with a maximum likelihood approach, in the method proposed here 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.

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.

The same approach can be extended to handle a hierarchy of context-dependent models that go from very broad context classes to highly specific ones. In pure HMM systems, such a hierarchy of models for a given phone are typically smoothed together using the deleted interpolation algorithm. The training and smoothing procedure described here can be generalized by defining a hierarchy of context classes in which every context class at one level is included in a broader class at the previous level. Every 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).

### 3.3 Context-dependent architecture

It is well known that the learning of input-to-hidden weights is the hardest and most time consuming. Therefore, in order to reduce the number of independent parameters to train, we propose an architecture where 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.

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. Doing so, we are actually assuming that the features provided by the hidden layer are useful for the context-dependent classification. 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 system, on which the hybrid system is based, two or three states are used to model each phonetic unit. Multiple state phone models allow more precise modeling of context effects because the initial portion of a phone segment is influenced more by the previous phone while the final part of a phone segment is influenced more 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 last state of every phone model. For the three state models the middle state was treated as context independent.

## 4. Implementation details

We have used this new architecture to model generalized biphone phonetic contexts. The set of context-specific networks described above, can be thought of as a single network with multiple outputs layers, one for each context. Eight generalized classes of left-context and eight generalized classes of right-context have been defined, principally based on place of articulation and acoustic characteristics.

First states of HMM phones are associated with the context-specific 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 3-state HMM phones are associated with a context independent layer which was trained only on frames that were aligned to middle HMM phone states.

During recognition, pure context-dependent posterior probabilities have to be converted to (scaled) state conditioned observation probabilities using the normalization factors provided by (1) and (2). However, because of the smoothing with the context-independent weights, the conversion factors should be a combination of those corresponding to the context-independent and context-dependent cases. We use the following heuristics for converting the smoothed posteriors $p^S(q_j|Y_t,c_k)$ to smoothed (scaled) observation probabilities $p^S(Y_t|q_j,c_k)$:

$$p^s\left(Y_p|q_j, c_k\right) = p^s\left(q_j|Y_p, c_k\right)\left(\alpha_j^k \frac{1}{p\left(q_j\right)} + \left(1-\alpha_j^k\right)\frac{p\left(c_k|Y_p\right)}{p\left(q_j|c_k\right)p\left(c_k\right)}\right) \tag{3}$$

where

$$\alpha_j^k = \frac{N_{ci}(j)}{N_{ci}(j) + b\left(N_{cd}(j,k)\right)} \tag{4}$$

$N_{ci}(j)$ 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 in a development set for minimum recognition error.

## 5 Experimental Evaluation

Training and recognition experiments with the HMM/MLP hybrid were conducted using the speaker independent,

| | CI MLP | CD MLP |
|---|---|---|
| Test 1 | 7.8 | 6.8 |

Table 1: Recognition error in percent for the 512 hidden unit MLPs using the word-pair grammar.

| | CI MLP | CD MLP |
|---|---|---|
| Test 1 | 6.6 | 5.2 |
| Test 2 | 6.1 | 5.0 |

Table 2: Recognition error in percent for the 1000 hidden unit MLPs using the word-pair grammar.

| | CI MLP | CD MLP |
|---|---|---|
| Test 1 | 26.0 | 22.4 |
| Test 2 | 25.0 | 23.7 |

Table 3: Recognition error in percent for the 1000 hidden unit MLPs using the all-word grammar.

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 size was about 1.3 million frames. For every frame a 12th order mel cepstrum was computed and 26 coefficients produced: log energy, 12 cepstral coefficients and their smoothed derivatives. A 9 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) determine the context class index. Hidden layer sizes of 512 and 1000 units were tested. With 1000 hidden units the size of the context-independent net was about 300,000 weights, while the size of the context-dependent net was about 1.4 million weights. Two additional context-independent nets were trained to provide the probabilities of the right and left context classes. They have the same 234 input units, 512 hidden units and 8 output units, one for each context class.

After the context-independent net was trained as in [3], its 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 may be overoptimistic because the context is 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.

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 due to the huge number of hypotheses that are explored in the Viterbi search, forward propagation is computed for every context-specific output layer for every frame.

We used one additional set of 300 sentences (June 88 release) for tuning parameters such as word transition penalties and the constant $b$ in (4). We found that the proposed heuristic (3) for combining the scaling factors was better than using either the pure context-dependent or the pure context-independent scalings by themselves.

In Table 1 recognition error is presented for a test set of 600 sentences (Feb 89 + Oct 89 releases) called Test 1, using the word-pair grammar, for the context-independent and the context-dependent HMM/MLP hybrid using 512 hidden units. In Table 2 recognition error is presented for the 1000 hidden unit nets using the word pair grammar, evaluated on Test1 and an additional new test set of 300 sentences (Feb 91 release) named Test 2. In Table 3 the same cases as in Table 2 are presented using the all-word grammar.

We consistently found significant improvements going from context-independent to context-dependent hybrid systems and from using 512 hidden units to using 1000 hidden units. The average reductions in word error rates were 20% in the word-pair case and 11% for the all-word case using 1000 hidden units. Combining both test sets, the differences between the context-independent and the context-dependent hybrid systems were significant at the 0.95 level in all cases.

## 6 Discussion

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

- How to reliably train the large context-dependent net, having less training data per context class.

- How to use discrimination in large context specific nets.

- How to smooth with context independent nets in order to obtain robust probability estimates.

- How to use multiple state HMM phone models with MLP probability estimation.

The proposed solutions were:

i) Sharing of pre-trained input-to-hidden layer among context-specific nets. It substantially reduced the number of parameters and 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, were useful to capture the more detailed context-specific phone classes.

ii) Apply discrimination only among phone classes in a given context. An interesting feature of the proposed training method is that discriminative training is not applied among output units associated with the same phone in different contexts, because these units belong to different context specific nets. That would not have been the case if we had defined MLP outputs for every phone-in-context.

iii) A training procedure that smooths networks with different degrees of context-dependence, producing a robust context-dependent network. 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 training allowed the training of a large number of parameters without loosing robustness. The proposed scheme can be extended to use a hierarchy of levels of context-dependence as are used in state-of-the-art HMM systems.

iv) 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 [3] 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 a larger number of parameters to train and/or 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 to train discriminatively is likely to be at the phone level rather than at the HMM state level.

The results obtained with the context-dependent HMM/MLP hybrid still are not as good as those obtained with the best HMM systems. Nevertheless, the results are encouraging given that they were obtained with a simpler system that only models generalized diphone contexts and uses roughly one fourth the number of parameters.

Future work will deal with modeling finer context dependent categories and exploring alternative architectures for context-dependent modeling with MLPs.

## 7 Conclusions

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.

The context-dependent MLP/HMM hybrid performed significantly better than context-independent MLP/HMM hybrid, using only very general context classes.

**Acknowledgments**

Support for this research project was provided by DARPA contract MDA904-90-C-5253. Chuck Wooters helped with some training runs. Horacio Franco was partially supported by a fellowship from CONICET, Argentina.

**References**

[1] H. Bourlard, and N. Morgan, ‘‘Merging Multilayer Perceptrons and Hidden Markov Models: Some Experiments in Continuous Speech Recognition’’, Neural Networks: Advances and Applications, North Holland Press, E. Gelenbe editor, 1990.

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

[3] 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.

[4] 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.

[5] Kai-Fu Lee, “Context-Dependent Phonetic Hidden Markov Models for Speaker-Independent Continuous Speech Recognition”, IEEE Transactions on Acoust., Speech and Signal Proc., Vol. 38, No. 4, April 1990.

[6] 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.

[7] 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, Ed. Amsterdam: North-Holland, 1980, pp. 381-397.

[8] 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.

[9] N. Morgan and H. Bourlard, “Continuous speech recognition using multilayer perceptrons with hidden Markov models”, ICASSP 90, Vol. 1, pp. 413-416, 1990.

[10] M. Franzini, Kai-Fu Lee, and Alex Waibel, “Connectionist Viterbi training: A new hybrid method for Continuous Speech Recognition,” ICASSP 90, Vol.1, pp. 425-428, 1990.

[11] 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.

[12] J. L. McClelland and J. L. Elman, “Interactive Processes in Speech Perception: The TRACE Model”, in Parallel Distributed Processing, Explorations in the Microstructure of Cognition, D. Rumelhart and J. McClelland Eds, Vol. 2, pp. 58-121, MIT Press, 1986.