HYBRID NEURAL NETWORK/HMM BASED SPEECH RECOGNITION

Shivali Srivastava
Department for Electrical and Computer Engineering
Mississippi State University, Mississippi State, MS 39762
srivasta@isip.msstate.edu

ABSTRACT

Hidden Markov Model (HMM) is used as a dominant approach in most state-of-the-art speaker-independent continuous speech recognition system. In this paper attempt is made to stimulate a discussion on new approach- hybrid neural network/hidden Markov model (HMM) based speech recognition system. The system provides a remedy of the problems caused by fundamental strong statistical assumptions of hidden Markov model that are unlikely to be valid for speech. It combines the advantages of both approaches by using multilayer perceptrons (MLPs) to estimate the state-dependent observation probabilities of an HMM. In this paper we described the approach for integrating MLP-based estimation techniques. Comparisons with a pure HMM system illustrate the advantages of the hybrid approach both in terms of recognition accuracy and number of parameters required.

1. INTRODUCTION

Currently, the most speech recognition system use context-dependent hidden Markov models (HMMs) [1] as a standard approach for handling the variability due to local phonetic context. In the hybrid model, the multi-layer perceptron computes the HMM context-dependent observation probabilities using a Bayesian factorization in terms of scaled posterior phone probabilities [3] which reduce the assumption of independence for multi-feature probability computation. Another advantages of MLP probability estimation include the inherently discriminant nature of the training algorithm and the distribution representation, which leads to efficient use of the available parameters. This, when applied to speech, results in the reduction of the number of parameters needed for detailed phonetic modeling because of increased sharing of model parameters between phonetic classes. Because of the need for accurate segmentation of speech signal, pure MLP-based approaches have not previously been demonstrated well. HMMs provide a framework for simultaneous segmentation and classification of speech. Morgan and Bourlard [3] has shown theoretically and practically that MLPs and HMMs can be combined by using MLPs for the estimation of HMM state-dependent observation probabilities, thereby exploiting the advantages of both approaches.

2. HYBRID MLP/HMM SYSTEM

The baseline hybrid HMM/MLP speech recognition system [7] replaces the tied-mixture HMM state-dependent observation probability estimates computed by MLP, keeping the HMM topology unchanged. The MLP architecture is a feed-forward network with 234 inputs, 512 hidden units, and 69 outputs. The 234 inputs represent 9 frames of cepstra, delta cepstra, log energy and delta-log-energy that are normalized to have zero mean and unit variance. Both the hidden and output layers consists of sigmoidal units.

The hybrid system is a phone-based, speaker independent, continuous speech recognition system, based on semi continuous HMMs [4]. The system extracts four features from the input speech waveform including 12th-order mel cepstrum, log-energy and their smoothed derivatives. The front-end produces the 26 coefficients for theses four features for each 10 ms frame of speech. Training of the phonetic models is based on maximum-likelihood estimation using the forward-backward algorithm [5]. Most of the phonetic models in the system have three states. To allow for short realizations, a small number of phone models have two states.

High recognition performance with HMM systems generally requires context-dependent phonetic models. The context-dependent version of the hybrid system uses phone models trained at a variety of levels of context dependence. The levels include word-specific phone, triphone, generalized triphone, cross-word triphone, left and right biphone and generalized biphone. Models conditioned by more specific contexts are linearly smoothed with more general models using the deleted interpolation algorithm [6] in order to maintain robustness even in high specific contexts that have little training data.

The words in the hybrid system are represented as probabilistic networks of phone models, specifying multiple pronunciations. These networks are generated by the application of phonological rules to baseform pronunciations for each word. To limit the number of parameters that must be estimated, phonological rules are chosen based on measures of coverage and overcoverage of a database of pronunciations resulting in networks that maximize the coverage of observed pronunciations while minimizing network size. Probabilities of pronunciations are estimated by the forward-backward algorithm at the same time the phonetic models are trained, after tying together instances of the same phonological process in different words.

The MLP is trained using stochastic gradient descent and a relative-entropy error criterion. The hybrid system is bootstrapped from the pure HMM DECIPHER system [8]. Recognition uses the Viterbi algorithm [5] to find the HMM state sequence that has the highest probability of generating the observed acoustic sequence.

Assuming we have enough training data, choose an appropriate number of parameters in the MLP, and the training does not get stuck in poorly performing local minima, the back-propagation-trained three-layer feedforward MLP approximates [9] the posterior class probabilities $P(q_j/Y_t)$, where q_j corresponds to the j-th phone class and the Y_t is the acoustic vector at time t. 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. Frame classification performance on an independent cross-validation set is used to control the learning rate. During recognition, Bayes's rule is used to convert the network outputs to the scaled phone-class conditional observation likelihoods required by the HMM,

$$P(Y_t/q_j) = \frac{P(q_j/Y_t) \cdot P(Y_t)}{P(q_j)} \qquad (1)$$

$P(q_j)$ is the prior probability of phone class q_j and is estimated by counting class occurrences. $P(Y_t)$ is common to all states for any give time frame, and can therefore be discarded in the Viterbi computation, since it will not change the optimal state sequence used to get the recognized string.

Subsequent reestimation of MLP and HMM parameters based on new alignments provided by the new hybrid MLP/HMM may improve the performance of the hybrid system.

3. MULTIPLE PHONETIC DISTRIBUTION

Modeling phonetic units with a sequence of distributions rather than a single distribution improves the performance of HMM-based systems. This allows the model to capture some of the dynamics of phonetic segments. The hybrid MLP/HMM system models most phones with a sequence of three HMM states.

The current MLP architecture uses three separate output layers, corresponding to the three states of HMM phone models [10]. Each output layer consists of 69 units, one for each phonetic class. During training, only frames aligned with first states of HMM phones are presented to the first output layer, while frames aligned with last states of HMM phones are presented to the third output layer and those aligned with second states of three-state HMM phones are presented to the second output layer. Thus this can be viewed as a set of three MLPs, corresponding to the three HMM state-positions, which have the same input-to-hidden weights. Since the training proceeds as if each output layer were part of an independent network, the system learns discrimination between phonetic classes (as represented within each output layer), but does not learn the discrimination between the different states of the same phonetic class (because they are represented in different output layers). During the viterbi recognition search, the appropriate output layer is referenced depending on which HMM state-position is being visited. This technique has been combined with the approach to context-dependent modeling.

4. CONTEXT-DEPENDENT HYBRID

Experience with HMM technology has shown that use of context-dependent phonetic models improves recognition accuracy significantly [11], because acoustic correlates of coarticulatory effects are modeled explicitly, producing sharper and less overlapping probability density functions for the different phone classes.

Using a Bayesian factorization in terms of scaled context-dependent posterior phone probabilities computed with a set of context-specific MLPs, the context-independent hybrid MLP/HMM is extended to model context-dependent phonetic classes. Two approaches are used to deal with the increased number of parameters: error-based smoothing of context-dependent and context-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.

4.1. Context-Dependent Factoring

In a context-dependent HMM, every state is associated with a specific phone class and context. During the viterbi algorithm search, 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, for each phone. Required HMM probabilities are computed using the following factorization:

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

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

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

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 different ways. One possible implementation treats the c_k as M additional binary inputs [3] to a single MLP. During training, only one of the M inputs is set to 1 for each pattern presentation, and the others are set to 0.

Another possible implementation [11] 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 outputs units for every different context.

An alternative implementation is based on a direct interpretation of the definition of conditional probability, considering the condition 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 context-independent case, except that each MLP is trained using only input-output examples obtained when corresponding context is c_k. In this approach, the same network architecture and training method applied to every context-specific net, permitting the smoothing scheme and sharing of parameters.

The factor $P(c_k|Y_t)$ can be computed using a three-layer feed-forward context-independent MLP whose outputs correspond to the context classes. 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 trained examples. The likelihood $P(Y_t)$ is common to all states for any given time frame, and can be discarded in the computation of Viterbi algorithm.

4.2. Training and Smoothing

An initial context-independent MLP is trained to estimate the context-independent posterior probabilities over the N phone classes. After the convergence of context-independent training, the resulting weights are used to initialize the weights going to the context-specific output layers Context-dependent training proceeds by back-propagating error from only the appropriate output layer. Otherwise, the training procedure is similar to that for the context-independent net, using stochastic gradient descent and a relative-entropy training criterion. Overall classification performance evaluated on an independent cross-validation set is used to determine learning rate and stopping point. Only hidden-to-output weights are adjusted during context dependent training.

Every context-specific net asymptotically converges to the context conditioned posteriors $P(q_j|Y_t, c_k)$. As a result of the initialization, the net starts estimating $P(q_j|Y_t)$, and from that 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.

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 is initialized with the weights of a previously trained context-specific MLP at the previous level in the hierarchy Figure 1.

4.3. Recognition

Recognition is accomplished using the Viterbi algorithm, it requires computation of the observation probabilities associated with each state of HMM. The smoothed context-dependent posterior probabilities supplied by the MLP are converted during recognition to state-conditioned observation probabilities using the normalization factors provided by equations (1) and (2). However, because these values are a result of smoothing context-dependent and independent networks, the normalization factors should be a combination of those corresponding to the context-dependent and context-independent cases. Following interpolations for converting the smoothed posteriors $P^S(q_j|Y_t, c_k)$ to smoothed observation probabilities $P^S(Y_t|q_j, c_k)$:

$$P^S(Y_t|q_j, c_k) = P^S(q_j|Y_t, c_k) \cdot f(j, k, t) \quad (4)$$

where

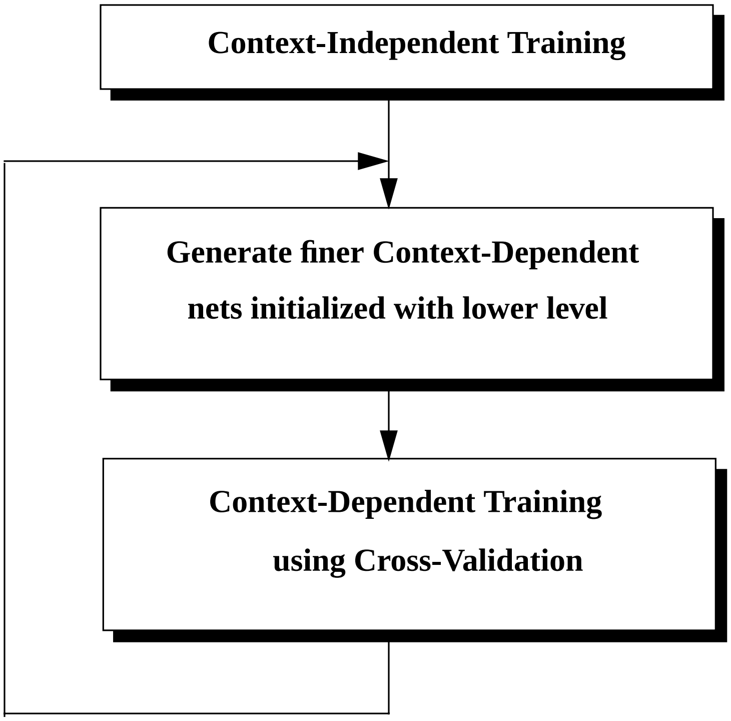
Context-Independent Training
Generate finer Context-Dependent nets initialized with lower level
Context-Dependent Training using Cross-Validation

Figure 1. Context-Dependent Training and Smoothing

$$f(j,k,t) = \alpha_k^j \cdot \frac{1}{P(q_j)} + \Psi(j,k,t) \quad (5)$$

$$\Psi(j,k,t) = (1-\alpha_k^j) \cdot \frac{P(c_k/Y_t)}{P(q_j/c_k) \cdot P(c_k)} \quad (6)$$

and

$$\alpha_k^j = \frac{N_{ci}(j)}{N_{ci}(j) + b[N_{cd}(j,k)]} \quad (7)$$

$N_{ci}(j)$ is the number of training examples for 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.

5. EVALUATION AND RESULTS

The training and recognition experiments comparing the MLP/HMM hybrid to pure HMM system was conducted by M. Cohen et al. [10] using the speaker-independent, continuous speech, DARPA Resource management database. The vocabulary size of 998 words was used. Tests were run both on a word pair grammar with perplexity 60 and an all-word grammar with perplexity 998. The training set was composed of 3990 sentences equivalent to about 1.5 million frames. 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. 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.

Training of the context-dependent net consisted of first training a context-independent net, which estimates $P^S(q_j|Y_t)$. Then this net's weights were used to initialize the context-dependent net. The final cross-validation error for the context-dependent net was 21.4% vs. 30.6% obtained with the context-independent network.

Combining all the tests, the differences between the context-independent and context-dependent hybrid was statistically significant at 0.05 level of significance. Tables given below show the percentage word error rate for pure HMM and hybrid MLP/HMM models for no grammar and all-word grammar.

	WER %	No. of Params.
CI-MLP	30.9	300K
CD-MLP	24.9	1,4000K
CD-HMM	26.6	5,500K
MIXED	21.5	6,100K

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

	WER (%)
CI-MLP	9.5
CD-MLP	6.6
CD-HMM	7.0
MIXED	5.7

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

6. DISCUSSION

The results shown in above tables suggest that MLP estimation of HMM observation likelihoods can improve the performance of standard HMMs. These results also suggest that systems that use MLP-based probability estimation make more efficient use of their parameters than standard HMM systems do. In standard HMMs, most of the parameters in the system are in the distributions associated with the individual states. MLPs use representations that are more distributed in nature, allowing more sharing of representational resources and better allocation of them based on training. In addition, since MLPs are trained to discriminate between classes, they focus on modeling boundaries between classes rather than class internals. The reduction in memory needs that may be attained by replacing HMM distributions with MLP based estimates must be traded off against increased computational load during training and recognition.

The tables show that the performance of CD-MLP system is roughly equivalent to that of CD-HMM, although CD-MLP is a far simpler system, with approximately a factor of four fewer parameters and modeling of only generalized biphone phonetic contexts. The best performance is that of the MIXED system.

7. CONCLUSIONS

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

[1] Eric Brill: *A corpus-Based Approach to Language Learning*, Ph.D. dissertation, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, 1993.

[2] Horacio Franco, Michael Cohen, Nelson Morgan, David Rumelhart and Victor Abrash, “Context-Dependent Connectionist Probability Estimation in a Hybrid Hidden Markov Model-Neural Net Speech Recognition System,” *Computer Speech & Language,* Vol. 8, pp. 211-222, 1994.

[3] N. Morgan and H. Boulard, “Continuous Speech Recognition Using Multilayer Perceptrons with Hidden Markov Models,” *ICASSP 90*, pp. 413-416, Alburquerque, New Mexico, 1990.

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

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

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

[7] S. Renals, N. Morgan, M. Cohen, and H. Franco, “Connectionist Probability Estimation in the DECIPHER Speech Recognition System,”*ICASSP 1,* pp. 601-604, 1992.

[8] H. Murveit, M. Weintraub, and M. Cohen, “Training Set Issues in SRI’s DECIPHER Speech Recognition System,” *DARPA Speech and Natural Language Workshop,* June 1990.

[9] M. Richard and R. Lippman, “Neural Network Classifiers Estimate Bayesian a posteriori Probabilities,” *Neural Computation,* pp. 461-483, Winter 1991.

[10] M. Cohen, D. Rumelhart, N. Morgan, H. Franco, V. Abrash and Y. Konig, “Combining Neural Networks and Hidden Markov Models for Continuous Speech Recognition,” *Proceedings of the DARPA Speech and Natural Language Workshop,* Harriman, NY, 1992.

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