INTEGRATING NEURAL NETWORKS INTO COMPUTER SPEECH RECOGNITION SYSTEMS

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

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

test

1. Introduction

Most current state-of-the-art continuous-speech recognition systems are based on hidden Markov modeling techniques. The work described here involved integrating neural networks into a hidden Markov model-based state-of-the-art continuous-speech recognition system, resulting in improvements in recognition accuracy and reductions in model complexity.

Hidden Markov models (HMMs) may be thought of as doubly stochastic finite state machines, consisting of a set of states, transition probabilities between states, and probability distributions over output symbols associated with each state. When used to model speech, these output symbols represent acoustic observations, modeling subphonetic acoustic events (e.g., closures, bursts, transitions). Current HMM-based speech recognition systems typically model phonetic units, or "phones" (e.g., the sound "m" in the word "map"), with a sequence of such states. Sequences of phone models can be concatenated to form word models. Word models can be connected according to grammatical constraints forming large networks that model any allowable sentence within an application. This approach allows a hierarchy of levels of linguistic description to be encoded within a uniform mathematical framework.

HMMs provide a convenient mechanism for modeling continuous speech. Recognition consists of finding the highest probability path through the model to account for the input speech. Segmentation and classification are performed simultaneously. Systems based on HMMs have achieved substantial improvements in large-vocabulary speaker-independent continuous-speech recognition in recent years.

Despite recent progress, current systems are still far from achieving the performance needed to allow flexible and habitable human-machine interfaces. Problems with current systems range from weaknesses in low level acoustic modeling to poor high level syntactic and semantic modeling. The work described here is focussed on better modeling of low level acoustics.

A number of weaknesses of the HMM approach have limited their ability to model certain important speech structure. In order to be computationally tractable, the HMM

formulation requires strong statistical assumptions (e.g., assumptions of statistical independence between speech features and between feature vectors in adjacent frames) which are unlikely to be valid for speech. These assumptions make it difficult for HMMs to model short-term and long-term correlations in the speech signal.

Techniques using neural networks (in particular, multilayer perceptrons) for probability estimation have recently been introduced, and methods have been shown for using multilayer perceptrons (MLPs) for the estimation of HMM state-dependent observation probabilities[1]. Using MLP probability estimation in this way overcomes some of the weaknesses of HMMs by allowing multi-feature probability estimation without assuming independence between features, allowing the modeling of short-term correlations within the MLP input window, and providing a framework for modeling long-term correlations[2,3]. Additional advantages of MLP probability estimation include the inherently discriminant nature of the training algorithm (using both positive and negative examples to learn class boundaries) and the distributed representation, which leads to efficient use of the available parameters. When applied to speech, this results in a reduction of the number of parameters needed for detailed phonetic modeling as a result of increased sharing of model parameters between phonetic classes.

We have incorporated MLP-based probability estimation techniques into the HMM-based SRI-DECIPHER system, which is a state-of-the-art, speaker-independent, continuous-speech recognition system. In this paper we describe the initial baseline DECIPHER system and the approach for integrating MLP-based estimation techniques; present a number of new techniques that have been developed to allow the modeling of context-dependent phonetic classes and long-term speech consistencies; and show the results of recognition experiments for the systems described.

2. Hybrid MLP/HMM

2.1 The DECIPHER System

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

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

represented as probabilistic networks of phone models, specifying multiple pronunciations. Word pronunciation modeling in the DECIPHER system is described in detail by Cohen [7].

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

2.2 Incorporating MLPs into DECIPHER

The hybrid MLP/HMM DECIPHER system substitutes (scaled) probability estimates computed with MLPs for the tied-mixture HMM state-dependent observation probability densities. The HMM topology is not changed.

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

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

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

The MLP has an input layer of 234 units, spanning 9 frames (with 26 coefficients for each) of cepstra, delta-cepstra, log-energy, and delta-log-energy that are normalized to have zero mean and unit variance. The hidden layer has 1000 units, and the output layer has 69 units, one for each context-independent phonetic class in the DECIPHER system. Both the hidden and output layers consist of sigmoidal units.

The MLP is trained to estimate $P(q_j|Y_t)$, where q_j is the class associated with the middle frame of the input window. Stochastic gradient descent is used. The training signal is provided by the HMM DECIPHER system previously trained by the forward-backward algorithm. Forced Viterbi alignments (alignments to the known word string) for every training sentence provide phone labels, among 69 classes, for every frame of speech. The target distribution is defined as 1 for the index corresponding to the phone class label and 0 for the other classes. A minimum relative entropy between posterior target distribution and posterior output distribution is

used as a training criterion. With this training criterion and target distribution, assuming enough parameters in the MLP, enough training data, and that the training does not get stuck in a local minimum, the MLP outputs will approximate the posterior class probabilities $P(q_j|Y_t)$ [1]. Frame classification on an independent cross-validation set is used to control the learning rate and to decide when to stop training as in [9]. The initial learning rate is kept constant until cross-validation performance increases less than 0.5%, after which it is reduced as $1/2^n$ until performance increases no further.

3. Context-Dependence

Experience with HMM technology has shown that using context-dependent phonetic models improves recognition accuracy significantly [10]. This is because acoustic correlates of coarticulatory effects are modeled explicitly, producing sharper and less overlapping probability density functions for the different phone classes. Context-dependent HMMs use different probability distributions for every phone in every different relevant context. This practice reduces the amount of data available to train phones in highly specific contexts, resulting in models that are not robust. The solution to this problem used by many HMM systems (including DECIPHER) is to use the deleted interpolation algorithm [6] to linearly smooth models trained at different levels of context specificity, ranging from context-independent to word specific. This approach cannot be directly extended to MLP-based systems because the smoothing of MLP weights makes no sense. It would be possible to use this approach to average the probabilities from different MLPs; however, since the MLP training algorithm is a discriminant procedure, it would be desirable to use a discriminant procedure to smooth the MLP probabilities.

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

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

Our initial implementation of context-dependent MLPs models generalized biphone phonetic categories (phone models dependent on the broad phonetic class of the immediately preceding or following phone). We chose a set of eight left and eight right generalized biphone phonetic-context

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

3.1 Context-Dependent Factoring

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

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

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

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

The factor $P(q_j|Y_t,c_k)$ is the posterior probability of phone class q_j given the input vector Y_t and the context class c_k. To compute this factor, we use 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 (all sharing the same input-to-hidden layer) similar to those used in the context-independent case except that each MLP is trained using only input-output examples obtained from the corresponding context, c_k.

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

$P(c_k|Y_t)$ is computed using a three-layer feed-forward MLP with an output unit corresponding to each context class,

the same 234 inputs as the MLP described above, and 512 hidden units. $P(q_j|c_k)$ and $P(c_k)$ are estimated by counting over the training examples. Finally, $P(Y_t)$ is common to all states for any given time frame, and can therefore be discarded in the Viterbi computation, since it will not change the optimal state sequence used to get the recognized string.

3.2 Context-Dependent Training and Smoothing

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

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

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

4. Long-Term Correlations

One of the weaknesses of the pure HMM approach is the lack of modeling of long-term correlations in speech due to the assumption that the acoustic observations in the early parts of an utterance are independent of those in later parts of the same utterance. This is clearly inaccurate; all portions of any given utterance are produced by the same speaker, the same vocal tract, in the same acoustic environment, and share common regional dialectal characteristics.

Some recent HMM systems try to model long-term consistencies in speech by training separate male and female models. This has the disadvantage of reducing the amount of training data available to train each of the models. Despite substantial differences between male and female speech, there

is clearly much the two have in common. We are currently exploring methods of modeling gender-based speech consistencies with MLPs which share most of their parameters between the two genders, and only split off a relatively small portion of the total parameters to model gender-specific characteristics. Comparisons of a number of alternative MLP architectures for gender-consistency modeling are presented in Abrash et al.[2] and Konig et al.[3]

5. Evaluation

5.1 Methods

Training and recognition experiments comparing the MLP/HMM hybrid to the pure HMM DECIPHER system were conducted using the speaker-independent, continuous-speech, DARPA Resource Management database. The vocabulary size is 998 words. Tests were run both with a word-pair (perplexity 60) grammar and with no grammar. The training set for the HMM system consisted of the 3990 sentences that make up the standard DARPA speaker-independent training set for the Resource Management task. To train the MLP, the training set was divided into a set of 3510 sentences for adjusting weights during back-propagation training, and 480 sentences for cross-validation. None of the tests described here use the gender-dependent DECIPHER system, which is described elsewhere [2].

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

	Feb89	Oct89	Feb91	# Parameters
CI-HMM	44.1	45.3	44.8	125K
CI-MLP	24.9	27.0	25.0	311K
CD-MLP	19.4	20.8	20.5	1,409K
CD-HMM	22.1	21.7	19.7	5,541K
MIXED	17.2	18.9	16.9	5,853K

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

	Feb89	Oct89	Feb91
CI-HMM	14.1	14.0	10.8
CI-MLP	5.4	7.6	6.2
CD-MLP	4.7	5.7	5.0
CD-HMM	5.0	4.9	4.0
MIXED	3.9	4.1	3.3

5.2 Results

Table 1 presents word recognition error and number of system parameters for five different versions of the system, for three different Resource Management test sets (February 89, October 89, and February 91) using the word-pair grammar. Table 2 presents word recognition error for the corresponding tests with no grammar (the number of parameters are the same as those shown in Table 1).

Comparing context-independent HMM (CI-HMM) to context-independent MLP (CI-MLP) consistently shows very substantial reductions in error rates using MLPs to estimate context-independent observation probabilities. One should keep in mind that although the CI-MLP computes context-independent phonetic probabilities of the form $P(q_j|Y_t)$, it can make use of some acoustic context, since the input window scans nine frames. However, this acoustic context is typically on a different timescale than that for phonetic context, and differs conceptually from computing context-dependent probabilities. The CI-MLP system, which uses about 300,000 parameters, showed performance close to the range shown by other systems listed with many more parameters.

The context-dependent HMM system (CD-HMM) performs consistently better than the CI-MLP system. This is not surprising, since the CD-HMM system is far more complex, with almost a factor of 20 more parameters, context-dependent phone models, and a sequence of (two or three) states with different observation distributions for each phone model. The CI-MLP system has a single context-independent output unit for each phonetic class. It is, in fact, interesting that such a relatively simple system performs as well as it does compared with far more complex systems.

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

Comparing CI-MLP to context-dependent MLP (CD-MLP) shows improvements with CD-MLP in all six tests, with an average reduction in word error of 20%. The CD-MLP system, using the context-dependent modeling technique described in Section 3, has roughly equivalent recognition performance to the CD-HMM system (with about one fourth the number of parameters): in three cases CD-HMM does better and in three cases CD-MLP does better.

5.3 Discussion

The results shown in Tables 1 and 2 suggest that MLP estimation of HMM observation likelihoods can improve the performance of standard HMMs. These results also suggest that systems that use MLP-based probability estimation make more efficient use of their parameters than standard HMM systems. In standard HMMs, most of the parameters in the system are in the distributions associated with the individual states. MLPs use representations that are more distributed in nature, allowing more sharing of representational resources and better allocation of representational resources based on training. In addition, since MLPs are trained to discriminate between classes, they focus on modeling boundaries between classes rather than class internals.

The distributed representation used by MLPs is exploited in the context-dependent modeling approach

described in Section 3 by sharing the input-to-hidden layer weights between all context classes. This sharing substantially reduces the number of parameters to train and the amount of computation required during both training and recognition. The context-dependent MLP has 17 times as many output units (classes) as the context-independent MLP, but has only a factor of 4.6 times as many parameters. In addition, we do not adjust the input-to-hidden weights during the context-dependent phase of training, assuming that the features provided by the hidden layer activations are relatively low level and are appropriate for context-dependent as well as context-independent modeling. This procedure substantially reduces context-dependent training time. The large decrease in cross-validation error observed going from context-independent to context-dependent MLPs (30.6% to 21.4%) suggests that the features learned by the hidden layer during the context-independent training phase, combined with the extra modeling power of the context-specific hidden-to-output layers, were adequate to capture the more detailed context-specific phone classes.

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

The performance of the CD-MLP system was roughly equivalent to that of the 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. We are currently extending the CD-MLP system to model more specific context classes, with the hope of exceeding the recognition performance of CD-HMM.

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

6. Conclusions

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

We have demonstrated a number of systems of different sizes and complexities, and shown some of the tradeoffs between system size, complexity, and performance. In the near future we intend to extend the system to model finer

phonetic contexts and to combine context-dependent MLPs with MLPs that model gender-based speech consistencies.

ACKNOWLEDGEMENTS

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

REFERENCES

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

[2] V. Abrash, H. Franco, M. Cohen, N. Morgan, and Y. Konig, ‘‘Connectionist Gender Adaptation in a Hybrid Neural Network/Hidden Markov Model Speech Recognition System,’’ ICSLP 92.

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

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

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

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

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

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

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

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

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

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

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