SIMPLIFIED NEURAL NETWORK ARCHITECTURES FOR A HYBRID SPEECH RECOGNITION SYSTEM WITH SMALL VOCABULARY SIZE

Hossein Sedarat, Rasool Khadem, Horacio Franco †

Dept. Electrical Eng., Information Systems Lab, Stanford University, CA 94305-9510
†Speech Technology and Research Laboratory, SRI International, Menlo Park, CA 94025

ABSTRACT

Recent studies suggest that a hybrid speech recognition system based on a hidden Markov model (HMM) with a neural network (NN) subsystem as the estimator of the state conditional observation probability may have some advantages over the conventional HMMs with Gaussian mixture models for the observation probabilities. The HMM and NN modules are typically treated as separate entities in a hybrid system. This paper, however, suggests that the *a priori* knowledge of HMM structure can be beneficial in the design of the NN subsystem. A case of isolated word recognition is studied to demonstrate that a substantially simplified NN can be achieved in a structured HMM by applying a Bayesian factorization and pre-classification. The results indicate a similar performance to that obtained with the classical approach with much less complexity in NN structure.

1. INTRODUCTION

Most state-of-the-art speech recognition systems are currently based on hidden Markov models (HMM). HMMs provide a non-stationary statistical model which fits the acoustic signals very well [1]. An HMM is a finite state machine defined by two sets of probability distributions. The first set, known as the transition probabilities, indicates how likely a transition from one state to another is. The second set, often referred to as observation probabilities, indicates the likelihood of observing acoustic features in each state. The HMM parameters are optimized for a training set by applying a maximum likelihood estimator. In the conventional systems, the observation probabilities are most widely modeled as a mixture of Gaussian distributions. Recent studies [2, 3, 4], however, have shown both theoretically and practically that a multi-layer perceptron (MLP) can be used to generate these probabilities. The main advantage of using an MLP is that it provides a flexible model with weaker assumptions on the functional form of the observation densities [4].

The HMM and NN modules are conventionally designed as independent blocks in a hybrid system. This paper points out the mutual effect of these two modules. More specifically, it will be shown that an HMM with regular two-dimensional structure may lead to an MLP with substantially reduced complexity.

In the following sections we examine a case of isolated digit recognition. We show that by casting a two-dimensional structure to the HMM, we can use a Bayesian factorization to break a large MLP into a few small ones. Bayesian factorization of observation probabilities in hybrid HMM-MLP systems was introduced by Bourlard *et. al.* [5], [6]. Various architectures derived from these factorizations have been applied to context-dependent speech recognition [7] and gender adaptation [8]. We present an extension of these procedures to the factorization of observation probabilities for the case of HMMs with regular structures such as those found in small vocabulary recognition systems.

We also present some of the related implementation issues and preliminary experimental results.

2. THEORY

It is well known that, when properly trained, a classifying MLP provides class *a posteriori* probabilities [9]. In a hybrid HMM-MLP speech recognition system, the MLP is used to generate the observation probabilities which are defined as follows:

$$b_i(\boldsymbol{x}) = p(\boldsymbol{X}(t) = \boldsymbol{x} | q_t = i) \tag{1}$$

where $\boldsymbol{X}(t)$ is the random vector of speech features and q_t indicates the state of the HMM at time t. It is not obvious from this equation, how to use an MLP to estimate $b_i(\boldsymbol{x})$. However, by using Baye's rule, this equation can be rewritten such that the new expression contains the *a posteriori* probability of states as:

$$b_i(\boldsymbol{x}) = p(\boldsymbol{x}|i) = \frac{P(i|\boldsymbol{x})p(\boldsymbol{x})}{P(i)} \tag{2}$$

Now, $P(i|\boldsymbol{x})$ is in a form that can be generated with an MLP. For the purpose of this paper, the value of the probability of acoustic features, $p(\boldsymbol{x})$, is not required because it is a common factor for all states at a given time. A normalized

b_i can be obtained by dividing Equation 2 by this common factor.

$$\bar{b_i}(\boldsymbol{x}) = \frac{P(i|\boldsymbol{x})}{P(i)} \tag{3}$$

Figure 1 shows a typical HMM for an isolated word recognition system. A case of digit recognition where the vocabulary consists of ten words is exemplified. We assume a 5-state model for each digit which accommodates the maximum number of phones per word in the vocabulary. It may seem that this model does not have an efficient structure because some words can be modeled with fewer states. We will shortly show that having similar structures for all the words helps us to simplify the MLP structure. There are also one initial and one final state in this figure representing silence. The total number of states in this

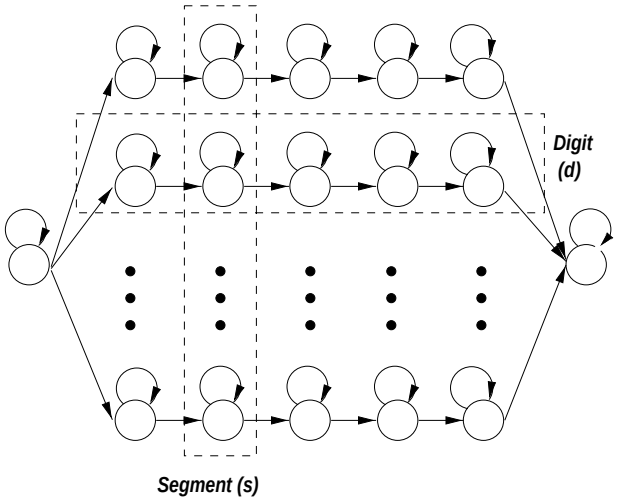

Figure 1: The HMM for isolated digit recognition

model is 50, not counting the silence states. Therefore, the MLP needs to classify 50 states and thus needs to have 50 outputs. However, this HMM is not an arbitrary collection of states and as it is clear from Fig. 1 it essentially forms a two-dimensional array of states. This particular structure is due to the left-to-right flow of the state transitions and the choice of equal number of states per word. We can take advantage of this structure to simplify either the complexity or the classifying task of the MLP.

In this two-dimensional array, each state can be referred to with two indices, row number and column number. We refer to each column of states as a segment s because it is associated with a time segment of speech signal. The second index is the row number which also identifies the corresponding digit d. Replacing each state number with the corresponding pair of indices in Eq. 3, we get

$$\bar{b_i}(\boldsymbol{x}) = \frac{P(s,d|\boldsymbol{x})}{P(s,d)} \qquad \begin{matrix} s = 1, \cdots, S \\ d = 1, \cdots, D \end{matrix} \tag{4}$$

where S is the total number of segments and D is the vocabulary size. For our example of digit recognition $S = 5$ and $D = 10$.

With a Bayesian factorization, Eq. 4 can be rewritten as:

$$\bar{b_i}(\boldsymbol{x}) = \frac{P(d|\boldsymbol{x},s)P(s|\boldsymbol{x})}{P(s|d)P(d)} \tag{5}$$

There are two terms in this expression that can be estimated by MLPs: the *a posteriori* probability of the segments $P(s|\boldsymbol{x})$, and the *a posteriori* probability of the digits for each segment $P(d|\boldsymbol{x},s)$. This suggests another structure for the observation probability estimator. In the original structure, a single MLP with $S \times D$ (50) outputs is used to generate $P(i|\boldsymbol{x})$. In the new structure, we have to use two MLPs. Each new MLP, however, is much smaller than the original one. The MLP that is used to estimate $P(s|\boldsymbol{x})$ has only S (5) outputs and the MLP corresponding to $P(d|\boldsymbol{x},s)$ has only D (10) outputs. Since the size of the input layers in both cases are almost equal, the new structure is much less complex than the original one.

There are several ways to implement the neural network associated with the factor $P(d|\boldsymbol{x},s)$. One is to consider the conditioning in the segment s as an additional input to the net. This input can take the form of a multi-valued single input or alternatively a 1-of-S binary valued set of inputs. Another way of implementing this neural network is based on the definition of conditional probability and is similar to the one proposed in [7]. It considers the conditioning on the segment s as restricting the set of input training vectors only to those belonging to that segment. This interpretation leads to a set of S MLPs, one for each value of s. Each MLP provides the observation probability of digit d, $P_s(d|\boldsymbol{x})$, for the corresponding segment s. This last scheme, essentially, pre-classifies an input training vector into S segments. This may simplify the classification task of each network as data from specific segments may be easier to classify. It may seem that the resulting neural network in this structure is more complex due to multiple MLPs being used. However, note that because of the pre-classification each new MLP is less complex than the original one and therefore the overall structure in both schemes can have the same order of complexity.

It is worth noting that the training of an MLP with Bayesian factorization not only requires the information on the digit d that the MLP is being trained for, but also needs the knowledge of the segment s that the input vector $\boldsymbol{x}$ belongs to. The optimal segmentation information is not available in advance and it is in fact obtained from the model itself. We use the so-called connectionist Viterbi training algorithm explained in [2]. It is an iterative approach which starts with an initial guess for segmentation. The model is optimized based on this guess and a better segmentation is obtained by finding the best state sequence using the Viterbi

algorithm. This procedure is repeated until the model parameters converge to a stationary point.

The *a posteriori* probabilities are not the only parameters to be estimated. To calculate the observation probabilities from Eq. 5 we also need to find the conditional probability of segments, $P(s|d)$, and the *a priori* probability of digits $P(d)$. The latter is usually obtained from a language model. In this work, it is assumed that all words in the vocabulary are equiprobable. The other term, $P(s|d)$, can be estimated from the segmentation information as

$$P(s|d) = \frac{\sum_n N_{ns}}{\sum_s \sum_n N_{ns}} \tag{6}$$

where N_{ns} denotes the number of input vectors allocated to the sth segment for the nth entry in the training set corresponding to digit d.

The other set of parameters in the HMM is the set of transition probabilities defined as

$$a_{ij} = P(q_t = j|q_{t-1} = i) \tag{7}$$

The maximum likelihood estimation of a_{ii} given the input segmentation is

$$a_{ii} = \frac{\sum_n N_{ns}}{\sum_n N_{ns} + \sum_n 1} \tag{8}$$

Since the HMM is strictly left-to-right then

$$a_{i,i+1} = 1 - a_{ii} \tag{9}$$

$$a_{ij} = 0 \qquad j \neq i \text{ or } i+1 \tag{10}$$

Figure 2 is the flow chart that summarizes the training process. Note that in addition to the re-segmentation loop, there is an implicit iterative gradient descent training loop inside the NN training block.

3. IMPLEMENTATION

We study three implementations of a hybrid system based on the HMM shown in Fig. 1. These implementations are different only in the architecture of the NN subsystem. Architecture 1 is the direct implementation which consists of just one MLP with 51 outputs, one for each state of the HMM. Architecture 2 is obtained by Bayesian factorization and has two MLPs. One MLP estimates the probability of segments, $P(s|\boldsymbol{x})$. It has 6 outputs, one for each segment and one extra for segments of silence. The second MLP provides *a posteriori* digit probabilities, $P(d|s, \boldsymbol{x})$. This MLP has 10 outputs and one additional input for segment index. Architecture 3 is obtained by taking into account the conditioning on the segment to train digit probabilities using 5 MLPs, one for each segment. Using the definition of conditional probability, each segment-specific MLP

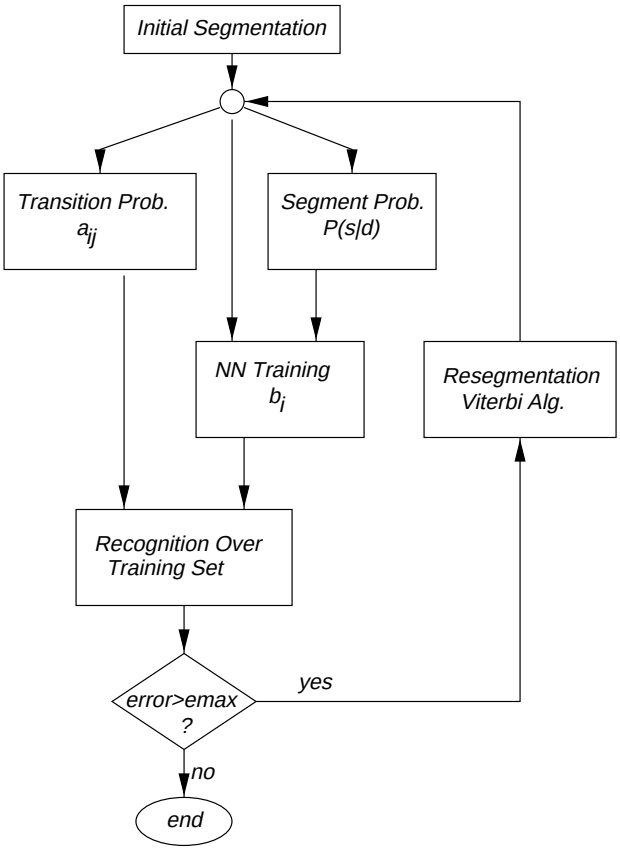

Figure 2: The training flow chart

is trained using the data associated with the corresponding segment.There is an MLP to compute segment probabilities identical to the one in the previous architecture.

We have considered a 3-layer structure for all MLPs. The acoustic input vector, $\boldsymbol{x}$, consists of the first 13 cepstral coefficients calculated over frames of length 25.6 ms. Each frame consists of 512 samples of speech signal and there is a 10 ms overlap between adjacent frames. An error back propagation algorithm with adaptive learning rate is used to train the neural networks. Both mean squared error and relative entropy have been examined as measures of error. A better performance is obtained consistently using relative entropy. This can be justified by considering the fact that the relative entropy is a better measure of distance for PDFs. The training set consists of 1000 utterances from a set of 50 diverse speakers. To avoid over-fitting, 20% of the training set is reserved for cross-validation.

4. RESULTS AND DISCUSSION

We have studied several implementations of each architecture with various sizes of hidden layer. Table 1 summarizes the best performance obtained from each architecture along with the corresponding relative size and complexity. The

size of each architecture is measured as the total number of weights in neural networks. The recognition complexity is the number of multiplications required to calculate the observation probabilities. The training complexity is a measure of the total training time for the neural networks. We normalized all numbers to that of architecture 1.

Arch.	Train (%)	Test (%)	size	complexity	
				training	recognition
1	99.87	98.40	1.0	1.0	1.0
2	99.87	98.60	0.7	0.3	1.5
3	100	99.00	1.1	0.3	1.1

Table 1: Best performance obtained for each architecture.

By applying the Bayesian factorization to architecture 1 we could substantially reduce the training complexity in architecture 2. However, this lead to higher recognition complexity. The use of segment-specific MLPs in architecture 3 has reduced this recognition time back to that of architecture 1 without increasing the training time. Architecture 3 has also exhibited the best recognition rate.

Table 2 is a comparison of the training and recognition complexities when all architectures provide the same level of performance. Architecture 1 is the most complex implementation and it has the highest number of parameters to be trained. Bayesian factorization greatly reduces the number of parameters in architecture 2. This allows better training and faster recognition. Architecture 3 is the least complex implementation obtained by using segment-specific MLPs. Although it has more parameters comparing to architecture 2, the corresponding training and recognition times are shorter due to its parallel structure. This structure can be used in a multi-processor system to further reduce the training and recognition time.

Arch.	Train (%)	Test (%)	size	complexity	
				training	recognition
1	99.87	98.40	1.0	1.0	1.0
2	99.87	98.40	0.4	0.3	0.7
3	99.87	98.40	0.6	0.2	0.6

Table 2: Complexity comparison when all architectures provide similar performance.

5. CONCLUSION

We introduced simplified neural network architectures in a hybrid system for speech recognition. In this system, like other hybrid systems, a hidden Markov model is used as the basic structure and a neural network is used to generate the necessary observation probabilities. We applied a Bayesian factorization to take advantage of the two-dimensional structure of the HMM in order to reduce the complexity of the neural network. We also suggested a segment-specific MLP architecture which simplifies the training process.

Although we presented our results in the case of isolated word recognition, our approach can be applied to any small-vocabulary hybrid HMM-MLP recognition system that can be casted in a regular two-dimensional structure.

6. REFERENCES

[1] L.R. Bahl, F. Jelinek, R.L. Mercer, "A Maximum Likelihood Approach to Continuous Speech Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-5, pp. 179-190, 1983.

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

[3] S. Renals, N. Morgan, M. Cohen, H. Franco, "Connectionist Probability Estimation in the DECIPHER Speech Recognition System," *ICASSP92*, vol. 1, pp. 601-604, San Francisco, 1992.

[4] S. Renals, N. Morgan, H. Bourlard, M. Cohen, H. Franco, "Connectionist Probability Estimators in HMM Speech Recognition," *IEEE Trans. Speech and Audio Processing*, vol. 2, no. 1, Part II, Jan 1994.

[5] H. Bourlard, N. Morgan, C. Wooters, and S. Renals, "CDNN: A Context Dependent Neural Network for Continuous Speech Recognition", *ICASSP92*, vol. 2, pp. 349-352, San Francisco, 1992.

[6] H. Bourlard, N. Morgan, "Continuous Speech Recognition by Connectionist Statistical Methods", IEEE Transactions on Neural Networks, Vol. 4, No. 6, Nov. 1993.

[7] H. Franco, M. Cohen, N. Morgan, D. Rumelhart, V. Abrash, "Context-dependent Connectionist Probability Estimation in a Hybrid Hidden Markov Model-Neural Net Speech Recognition System," *Computer Speech and Language*, (1994) **8**, 211-222.

[8] V. Abrash, H. Franco, M. Cohen, N. Morgan, Y. Konig, "Connectionist Gender Adaptation in a Hybrid Neural Network / Hidden Markov Model Speech Recognition System," *International Conference on Spoken Language Processing*, vol. 2, pp. 911-914, Oct 1992.

[9] M.D. Richard, R.P. Lippman, "Neural Network Classifiers Estimate Bayesian *a posteriori* Probabilities," *Neural Computation*, 3, 461-483, 1991.