THE CASCADE HMM/ANN HYBRID : A NEW FRAMEWORK FOR DISCRIMINATIVE TRAINING IN SPEECH RECOGNITION

Iman Gholampour, Kambiz Nayebi

Electrical Engineering Department
Sharif University of Technology
Tehran, 11365, Iran
Gholamp@ee.sharif.ac.ir

ABSTRACT

In this paper, a new formulation for discriminative training of HMMs is presented. This formulation uses a properly trained MLP in a simple interconnection with HMMs called *"Cascade HMM/ANN Hybrid"*. Our training algorithm has simple realization in comparison with other discriminative training for HMMs such as MDI and MMI. We also present a rigid mathematical proof of its convergence. We found that using cascade HMM/ANN for isolated word recognition in noisy environment results in increasing the recognition accuracy from 93.3% in classic HMMs to 99.1% using a two layer MLP. No significant increase in computational requirements is needed in recognition phase. Both theoretical and experimental achievements are included in the paper.

1. INTRODUCTION

In spite of the fact that speech exhibits features that can not be represented by a first order Markov models, Hidden Markov Models (HMMs) of speech units (e.g. words or phonemes) have been used with a good degree of success in Automatic Speech Recognition (ASR)[1],[4]. On the other hand, Artificial Neural Networks (ANNs) and in particular Multi Layer Perceptrons (MLPs) trained with Back Propagation (BP), have proven to be useful for nonlinear classification of speech properties of limited duration[2]. Various attempts have been made to interpret time evolution of ANN outputs and discriminative training of HMMs[3]. Bourlard and Morgan used an MLP with large number of hidden layer nodes to estimate HMM state probabilities[5],[6]. Kershaw et al. reported similar results by using recurrent neural networks in place of MLP[7]. Rigoll et al. introduced another combination of MLP and discrete density HMMs. In this combination MLP performs the vector quantization task[8],[9]. All of these researchers reported significant improvement over classic HMM-based speech recognizers. In this paper, we outline a new formulation for discriminative training of HMMs using properly trained MLP in a new hybrid combination named as *"Cascade HMM/ANN"*. In this combination MLP is placed in a cascade interconnection with HMMs and performs a nonlinear classification of HMMs scores. We found (both theoretically and experimentally) that this cascade interconnection inherits both time domain modeling capability of HMM and discriminative property of MLP which results in a significant improvement in recognition accuracy. In classic HMM architecture for isolated word recognition, each word HMM is trained separately using *Maximum Likelihood* (ML) approach of Baum-Welch recursion. In such a system, the word HMM with maximum score for the input utterance determines the recognition result regardless of other word HMM scores. This gave us a strong clue to use all HMM scores in an MLP nonlinear classifier to recognize any input utterance. First, HMM parameters are estimated using Baum-Welch recursion as an initial point of our training scheme. Then an MLP with equal number of input and output layer nodes, and properly selected number of hidden layer nodes (about 10 times of input layer nodes) is used for classification of the normalized HMM scores. The MLP is trained using BP procedure with the same training set used in Baum-Welch recursion. After this phase, we update the initial estimates of output probability density of each HMM state using another training set keeping the MLP weights and HMM state transition probabilities fixed. Several experiments prove that output probability densities are much more important in recognition rate than state transition probabilities[4],[5]. We use a new formulation for passing gradient of back propagation procedure from MLP outputs to the HMM scores and then to the output probability density of each HMM state. In other words, the gradient of MLP outputs with respect to each output probability density of HMM state is computed and used for updating these probability densities using "Steepest Descent" algorithm. We showed that this gradient computation can be performed efficiently using forward and backward HMM variables. We also proved that the convergence of this updating scheme to a local maximum is guaranteed. In the realization phase, we use the cascade HMM/ANN hybrid in an isolated word recognition task as a performance test. After all training phases, our new structure recognized the test utterances much better than classically trained HMMs (about 6% increase in recognition accuracy). This paper is organized as follows. In Section 2 we describe cascade HMM/ANN structure and its training phases. In Section 3 convergence of training algorithms is discussed. Experimental evaluation of our algorithms is also included in this section. Finally we summarize our major findings and experimental results.

2. CASCADE HMM/ANN HYBRID AND DISCRIMINATIVE TRAINING

In this section we assume that for each unit to be recognized (e.g. word or phone) a unique left-right HMM is assigned. As stated earlier, scores of these HMMs for any isolated input utterance is used as inputs to an MLP. Figure 1 shows this structure with K HMMs and a two layer MLP. This structure is very similar to the classic structure of HMM-based isolated word recognition and can be used directly for isolated word

recognition. But for phoneme-based continuous speech recognition tasks a presegmentation algorithm must be used to convert the problem to an isolated utterance task. In the following subsections we present the mathematical basis of our training algorithm.

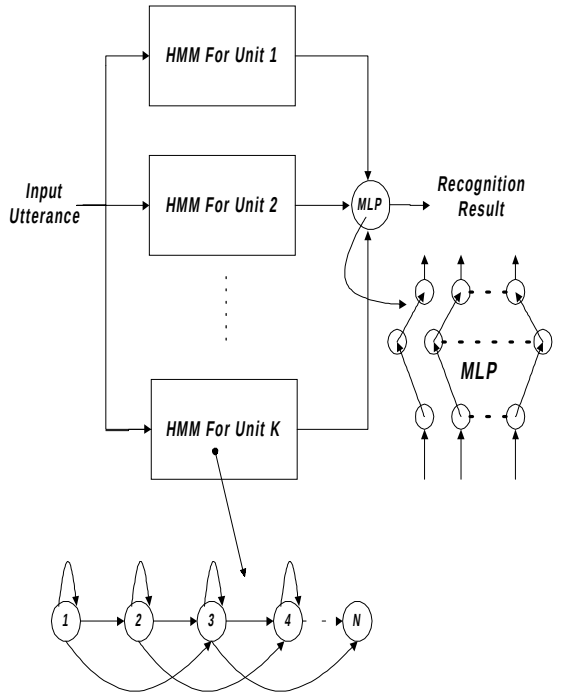

Figure 1. The Cascade HMM/ANN Hybrid.

2.1. Training Procedure

Our proposed training procedure for cascade HMM/ANN Hybrid can be performed in three phases. In the first phase, each HMM is trained separately with several samples of its related units using Baum-Welch Recursion. This phase is completely similar to the training phase of HMMs in classic isolated word recognizer. In the second phase, all of the training utterances, used in the first phase are applied to all trained HMMs. Scores of these HMMs are used to train a two layer MLP with properly selected number of hidden layer nodes. This MLP has equal number of inputs and outputs, as depicted in Figure 2, and is trained using BP procedure. The trained MLP then is capable of classifying the HMM scores. The third phase is responsible for fine tuning HMM parameters using steepest descent algorithm. For this purpose gradient of MLP outputs with respect to HMM parameters must be computed. We assumed that only state probability densities of HMMs are important for fine tuning. As stated earlier, state transition probabilities are not as important as state probability densities. So our training scheme updates only the state probability densities and the state transition probabilities remained unchanged. The MLP weights are also kept fixed in this training phase. We divide the gradient computation procedure into two parts. First, the gradient of HMM scores with respect to state probability densities is computed and then BP gradients and chain rule are used to calculate desired gradients. These steps are fully described in next subsections.

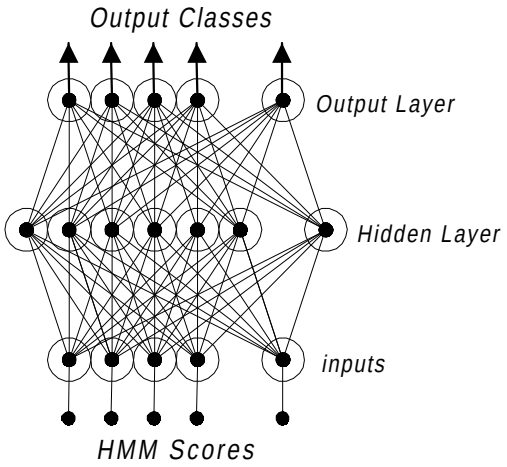

Figure 2. Appropriate MLP structure in cascade HMM/ANN hybrid.

2.2. HMM Scores Gradient Computation

As stated previously, we derive a new formulation for gradient of HMM scores with respect to the HMM state probability densities. The mathematical foundation of this derivation is presented here. Let $L(k)$ assigned to the HMM score of kth unit:

$$L(k) = P(\underline{O}|\lambda_k) \tag{1}$$

λ_k : The kth HMM.

$\underline{O}$: The observation sequence (feature vector sequence of input utterance).

According to the ML training criterion each $L(k)$ can be maximized separately regardless of other $L(i)$ ($i \neq k$) [1]. If "c" denotes the index of winner HMM, $L(c)$ is the maximum of $L(i)$ $(i=1,..K)$ and we can define another useful variable as:

$$H(i) = \frac{L(i)}{\sum_k L(k)} \tag{2}$$

If the units are equiprobable, maximizing $H(c)$ or $logH(c)$ is equivalent to a discriminative training scheme called MMI (Maximum Mutual Information)[3],[4]. This means that only a simple scaling converts a nondiscriminative criterion to a discriminative one, but we lose the simplicity of ML training procedure. Now we need an appropriate gradient of $H(k)$ with respect to the HMM parameters. First we compute the gradient of $H(k)$ with respect to the state probability densities of HMMs. It is worth mentioning that this step does not depend on the type of these probability densities (mixture of Gaussians in continuous and semicontinuous HMMs or discrete type probability density function in discrete HMMs). For $i = c$ we have:

$$\frac{\partial H(c)}{\partial b(j,t)} = \frac{\sum_{k \neq c} L(k)}{(\sum_k L(k))^2} \frac{\partial L(c)}{\partial b(j,t)} \quad (3)$$

and for $i \neq c$:

$$\frac{\partial H(i)}{\partial b(j,t)} = \frac{-L(i)}{(\sum_k L(k))^2} \frac{\partial L(c)}{\partial b(j,t)} \quad (4)$$

in which $b(j,t)$ is the jth state probability density at time t.

According to Eq. (3) and Eq. (4) regardless of i, the derivative of $H(k)$ always depends on the derivative of $L(c)$, that is the score of the winner HMM. Derivative of $L(c)$ can be expressed in terms of HMM forward and backward variables (α and β respectively). Recall that in HMM classic formulation:

$$L(c) = \alpha_{F_c}(T) \quad (5)$$

$$\alpha_i(t) = b(i,t)\sum_j a_{ji}\alpha_j(t-1) \quad (6)$$

$$\beta_i(t) = \sum_j a_{ij} b(j,t+1)\beta_j(t+1) \quad (7)$$

$\alpha_{F_c}(T)$: Forward HMM variable in the final state of the winner.

$\alpha_i(t)$: Forward HMM variable in the ith state at time t.

$\beta_i(t)$: Backward HMM variable in the ith state at time t.

T : The length of input utterance.

So by chain rule we have:

$$\frac{\partial L(c)}{\partial b(i,t)} = \frac{\partial \alpha_{F_c}(T)}{\partial \alpha_i(t)} \frac{\partial \alpha_i(t)}{\partial b(i,t)}$$

From Eq. (5) and Eq. (6) and by little manipulation:

$$\frac{\partial L(c)}{\partial b(i,t)} = (\sum_j \frac{\partial \alpha_j(t+1)}{\partial \alpha_i(t)} \frac{\partial \alpha_{F_c}(T)}{\partial \alpha_j(t+1)})(\sum_j a_{ji}\alpha_j(t-1))$$

$$= (\sum_j b(j,t+1)a_{ij} \frac{\partial \alpha_{F_c}(T)}{\partial \alpha_j(t+1)})(\sum_j a_{ji}\alpha_j(t-1))$$

According to Eq. (6) the second parenthesis in the above equation can be further simplified to $\alpha_i(t)/b(i,t)$. On the other hand, the expression, $\partial\alpha_{F_c}(t)/\partial\alpha_j(t+1)$ in the first parenthesis is satisfied in backward recursion of Eq. (7) as β. That is :

$$\frac{\partial \alpha_{F_c}(T)}{\partial \alpha_i(t)} = \sum_j a_{ij} b(j,t) \frac{\partial \alpha_{F_c}(T)}{\partial \alpha_j(t+1)} \quad (8)$$

Meanwhile:

$$\frac{\partial \alpha_{F_c}(T)}{\partial \alpha_{F_c}(t)} = 1 = \beta_{F_c}(T)$$

So we can conclude that:

$$\frac{\partial L(c)}{\partial b(i,t)} = \beta_i(t) \frac{\alpha_i(t)}{b(i,t)} \quad (9)$$

By Eq. (10), Eq. (3) and Eq. (4) can be compressed in a single equation:

$$\frac{\partial H(k)}{\partial b(i,t)} = \frac{\delta_{kc} - H(k)}{\sum_j L(j)} \beta_i(t) \frac{\alpha_i(t)}{b(i,t)} \quad (10)$$

in which δ_{kc} is discrete "*Dirac*" delta function.

Because of the simplicity of Eq. (10), our updating scheme can be implemented efficiently. The remaining step in gradient computation depends on the HMMs type. For continuous and semicontinuous HMMs, $b(i,t)$ is a parametric probability density function. Let θ denotes the parameters of $b(i,t)$. We can compute $\partial b(i,t)/\partial\theta$ for the mixtures of Gaussians easily and then the updating formula can be derived directly from Eq. (10) and the chain rule:

$$\frac{\partial H_k}{\partial \theta} = \sum_i \frac{\partial H_k}{\partial b(i,t)} \frac{\partial b(i,t)}{\partial \theta} \quad (11)$$

For discrete HMMs $b(i,t)$ is a nonparametric discrete probability distribution and Eq. (10) presents the desired gradient. So we can assume $\theta=b(i,t)$ for discrete HMMs to unify these situations.

2.3. Total Gradient and Updating Scheme

Adding the MLP to the classic HMM configuration needs a little manipulation of the above formulation. We can regard the MLP as a differentiable function of the HMM scores if and only if the differentiable functions are selected as its neuron nonlinearities. For example a *Sigmoid* type nonlinearity is appropriate for this purpose[1]. Let η denotes the outputs of this MLP, then we have:

$$\frac{\partial \eta}{\partial \theta} = \sum_k \frac{\partial H_k}{\partial \theta} \frac{\partial \eta}{\partial H_k} \quad (12)$$

The second derivative in the above sum is the gradient of MLP output with respect to MLP input that can be accessed via BP. So the Eq. (12) gives us the desired gradient or as we call it the *"total gradient"*. Now we can write the updating rule for HMM parameters easily using steepest descent algorithm :

$$\hat{\theta}^{(t)} = \hat{\theta}^{(t-1)} + \mu \frac{\partial \eta}{\partial \theta} \quad (13)$$

where μ is a positive real number and " ^ " sign denotes estimation. For Discrete HMMs a normalization must be performed each time we use the Eq. (13), so that the statistical restrictions on all $b(i,t)$, as discrete probably distributions, are

satisfied. It is worth mentioning that for each element of each observation sequence, only one element of $b(i,t)$ is updated. That is:

$$\hat{b}(i,o_t)^{(n)} = \hat{b}(i,o_t)^{(n-1)} + \mu \frac{\partial \eta}{\partial b(i,t)} \qquad (14)$$

In the next section, a convergence proof of our training scheme and its experimental evaluation will be presented.

3. ADVANTAGES OF OUR TRAINING SCHEME

In this section we present some advantages of our training scheme. First, we show that the convergence of this updating algorithm is guaranteed. Then the results of our experimental evaluation are discussed. These results show that our method adds acceptable discrimination to classic HMM-based isolated word recognizers.

3.1. Convergence Proof

Using Eq. (13) we can simply write :

$$\eta_t - \eta_{t-1} = \frac{\partial \eta}{\partial \theta}(\hat{\theta}^{(t)} - \hat{\theta}^{(t-1)})$$

$$= \mu \left| \frac{\partial \eta}{\partial \theta} \right|^2$$

But $\mu > 0$ and we have:

$$\eta_t > \eta_{t-1} \qquad (15)$$

Thus η is strictly increasing during the updating procedure. Figure 3 shows a sigmoid type function usually used as neuron nonlinearity in neural networks. Using this type of nonlinearity or any other limited function, η remains limited during the updating phase, that is:

$$\|\eta\| < M \qquad (16)$$

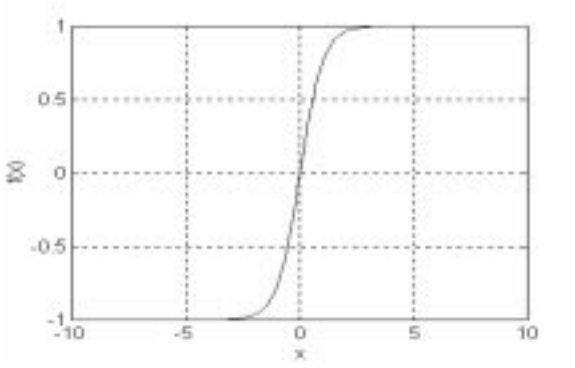

Figure 3. Sigmoid Type Nonlinearity.

According to Eq. (15) and Eq. (16), the values of η form a strictly increasing and limited sequence during the updating procedure. These conditions imply that this sequence will converge to a local maximum. So our training procedure has a guaranteed convergence if and only if the neuron nonlinearities are limited functions, as we stated before.

3.2. Experimental Evaluation

In the realization phase, we use the cascade HMM/ANN hybrid in an isolated word recognition task for *12* words (*0*, *1*, *2*, *3*, *4*, *5*, *6*, *7*, *8*, *9*, *Yes* and *No* in Farsi). The training set includes 100 repetitions of each word from *20* native speakers (*1200* utterances in sum). Another similar set is used for test. All recordings are performed in a noisy room condition with average signal to noise ratio of *21* dB and saved in PCM wave format with *16* KHz sampling rate and 16 bits per sample. Features include *12* LP derived cepstral coefficients, *12* delta-cepstral coefficients, energy and delta-energy which are extracted from *25* msec speech frames every *10* msec. We perform three evaluations on the test set. First, classic HMM-based isolated word recognizer is tested. That is *12* HMMs are trained (one HMM for each word) through Baum-Welch Recursion. This structure shows the recognition rate of *93.3%*. The second evaluation is performed on cascade structure before performing the updating procedure. In this experiment an MLP with *12* nodes in output and *120* nodes in hidden layer is trained using BP. The cascade structure shows the recognition rate of *95%*. The third experiment is performed after the updating scheme. The fine-tuned cascade structure shows considerably higher recognition rate (namely *99.1%*). The justification of this improvement is the discrimination added by our training scheme. Following comments clarify the situation.

Our experiments show that the recognition errors mostly occur between pairs (*0*,*3*) and (*2*,*9*) (which are pronounced /s/e/f/r/, /s/e/, /d/o/ and /n/o/h/ respectively in Farsi). As you see, these pairs are similarly pronounced words. By using cascade HMM/ANN hybrid such recognition errors reduced significantly in comparison with classically trained HMMs (about *86%* reduction in number of such errors). So, our final recognizer discriminates between similarly pronounced words much better than ML-based recognizers. Table 1 summarizes the improvements introduced by our training scheme for these confusing pairs.

Confusable Pair	Classic Structure	Cascade Structure
(0 , 3)	49	7
(2 , 9)	23	3
Sum	72	10

Table 1. Comparison between classic and cascade structures in number of confusing pairs errors.

During the training phases, an experiment is performed for determining the optimum number of hidden layer nodes in MLP. Figure 4 summarizes the results of this experiment. According to this test, the optimum number of hidden layer nodes is about *10* times of output layer nodes. In theory, the classification power of an MLP increases when the number of hidden layer nodes is increased. But our experiment shows that incorporating large number of nodes in hidden layer degrades the recognition error. This situation occurred because of the limitation in the size of our training set. Thus recognition rate can not be improved further by increasing number of hidden layer nodes from this optimum value.

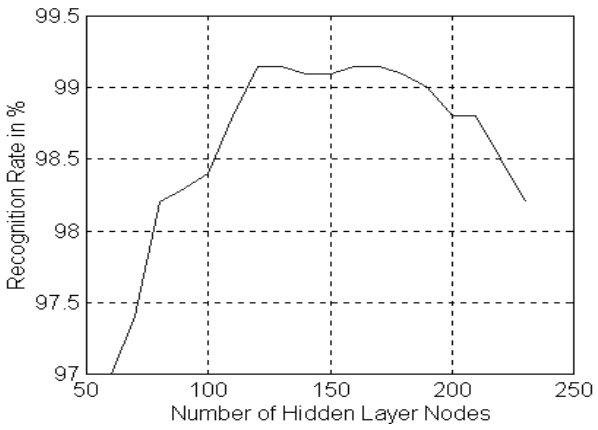

Figure 4. Recognition rate versus number of hidden layer nodes in MLP.

Training Phase	Recognition Rate	Improvement
(a)	93.3%	0%
(b)	95%	1.7%
(c)	99.1%	5.8%

Table 2. Recognition rate and its improvement (compared to classic HMM-based isolated word recognizer) in each training phase according to our isolated word recognition experiment, (a) Classic HMM structure, (b) After adding the MLP, (c) After fine tuning.

4. CONCLUSION

In Summary we developed a new framework for discriminative training in speech recognition with guaranteed convergence and comparably simple realization. This framework uses the cascade interconnection of HMMs and an MLP named as *"Cascade HMM/ANN Hybrid"*. The initial estimates of system parameters in our training procedure are obtained via standard training schemes for each part of the cascade structure, that is Baum-Welch for HMMs parameters, and BP for MLP. Then an updating step is performed to improve the initial estimates. We also prove that this updating scheme is converged if and only if MLP neuron nonlinearities are limited functions. This cascade structure is used in an isolated word recognition task and shows much better performance than classic HMM-based systems. Although we use discrete density HMMs in our experiment, our techniques can be used easily for continuous and semicontinuous density HMMs as shown mathematically in this paper. Table 2 summarizes the results of our tests. Referring to this table each training step increases the recognition rate. Meanwhile our final system has considerably higher performance than classic recognizer (about 6% more accurate). On the other hand, analysis of recognition errors in this experiment reveals that the cascade structure discriminates between similarly pronounced words much better. This analysis is summarized in Table 1. The cascade HMM/ANN hybrid can also be incorporated in continuous speech phoneme recognition using a two pass procedure. In the first pass, the continuous speech is segmented to the phone-like units and the problem is changed to a discrete utterance recognition. In the second pass a cascade HMM/ANN hybrid which includes phoneme HMMs, is used to recognize these discrete utterances.

5. REFERENCES

[1] K. F. Lee and H. W. Hon, "Speaker Independent Phone Recognition Using HMM's", IEEE Trans. Audio, Speech and Signal Processing, vol. ASSP-37, pp. 1641-1648, 1989.

[2] Y. Bengio, et al., "Global Optimization of a Neural Network Hidden Markov Model Hybrid", IEEE Trans. Neural Networks March 1992.

[3] H. Bourlard and C. J. Wellekenes, "Links Between Markov Models and Multilayer Perceptrons", IEEE Trans. Pattern Analysis and Machine Intelligence, Dec. 1990.

[4] L. Rabiner and B. H. Juang , "Fundamentals of Speech Recognition", PTR Prentice Hall Inc., 1993.

[5] S. Renals and N. Morgan, "Connectionist Probability Estimation in HMM Speech Recognition", Technical Report TR-92-081, ICSI Berkeley University , Dec. 1992.

[6] H. Bourlard and N. Morgan, "A Continuous Speech Recognition system Embedding MLP into HMM", In D. S. Touretzky, editor, Advances in Neural Information Processing Systems, volume 2 , pp. 413-416. Morgan Kaufmann, San Mateo CA, 1990.

[7] D. J. Kershaw, M. M. Hochberg and A .J. Robinson, "Incorporating Context-Dependent Classes in a Hybrid Recurrent Network-HMM Speech Recognition System", F-INFENG TR217, Cambridge University Engineering Department, May 1995.

[8] G. Rigoll and C. Neukirchen, "A New Approach to Hybrid HMM/ANN Speech Recognition Using Mutual Information Neural Networks", in Advances in Neural Information Processing Systems 9, NIPS*96, Denver, Dec. 1996, pp. 772-778.

[9] G. Rigoll and D. Willet, "A NN/HMM Hybrid for Continuous Speech Recognition with a Discriminant Nonlinear Feature Extraction", IEEE Intern. Conference on Acoustics, Speech and Signal Processing (ICASSP), Seatle, 1998, pp. 9-12.