DISCRIMINATIVE FEATURE AND MODEL DESIGN FOR AUTOMATIC SPEECH RECOGNITION

Mazin Rahim, Yoshua Bengio and Yann LeCun

AT&T Labs - Research, 600 Mountain Avenue, Murray Hill, New Jersey 07974, USA

ABSTRACT

A system for discriminative feature and model design is presented for automatic speech recognition. Training based on minimum classification error with a single objective function is applied for designing a set of parallel networks performing feature transformation and a set of hidden Markov models performing speech recognition. This paper compares the use of linear and non-linear functional transformations when applied to conventional recognition features, such as spectrum or cepstrum. It also provides a framework for integrated feature and model training when using class-specific transformations. Experimental results on telephone-based connected digit recognition are presented.

1. INTRODUCTION

Improving the performance of hidden Markov model (HMM) based automatic speech recognition (ASR) systems has been a central issue that has dominated the entire field of speech recognition during the past two decades. One effort to improving HMMs has been by extending the training paradigm beyond that of maximum likelihood (ML) to minimize a cost function that more directly relates to the recognition error rate. With the advent of discriminative training techniques, such as maximum mutual information (MMI) [1] and minimum classification error (MCE) [5], model learning has become a task of maximizing class separability rather than a likelihood function. Although this progress has been crucial in the development of more accurate HMMs, it is limited by the type of features used in the recognition design. Feature extraction plays an important role in ASR where the objective is to extract a set of parameters from the speech that provides class *discrimination* as well as *robustness* to extraneous signal components. Although cepstral based features have widely dominated this field, their design criterion is *not* consistent with the objective of minimizing recognition error rate. Integrated feature and model design under a single training objective clearly provides an additional benefit over conventional systems and remains a challenging problem in speech recognition research.

Integrated feature and model design through discriminative training has been the subject of several recent studies [2, 4, 6, 3]. These studies have reported encouraging results when applying either MMI or MCE training for designing a feature extractor based on a linear transformation and a classifier based on an artificial neural network (ANN), K-nearest neighbor or a HMM. In Bengio *et al* [2], both the ANN which was used for phonetic classification and the HMM recognizer were designed through MMI training. Euler [4] reported improved recognition performance on spelled names when applying MCE for training a feature-based transformation matrix with an HMM recognizer. Improved recognition performance using MCE was also reported by Chengalvarayan and Deng [3] when testing on the TIMIT database.

In [8], we proposed an integrated framework for discriminative feature and model design. The parameters of the feature extractor as well as the HMM-based recognizer were jointly optimized to minimize a single objective function based on MCE training. "Discriminative" features were extracted by training a set of *class-specific* affine transformations. Thus, feature extraction was considered as part of the recognizer design, with each transformation being associated with a specific unit model.

In this paper, we investigate the use of non-linear transformation for discriminative feature design. The proposed system for feature extraction has been adopted in adaptive inverse control [9] and includes a set of class-specific networks, each having a linear transformation represented by an affine and a non-linear transformation represented by an ANN. The outputs of both transformations are combined, and their corresponding parameters are jointly optimized with the HMM parameters through MCE training. We will describe some of the issues involved in discriminative feature and model design and report experimental findings on telephone-based connected digit recognition.

2. SYSTEM ARCHITECTURE

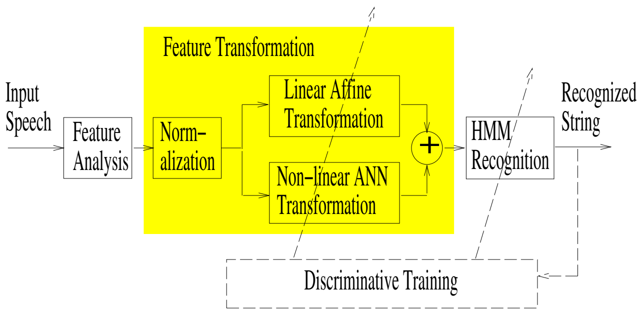

Figure 1: A block diagram for discriminative feature and model design.

A general block diagram of the proposed system for discriminative feature and model design is shown in Fig. 1. Feature extraction includes analyzing the speech signal and converting it into a set of meaningful coefficients, such as log spectrum or cepstrum.

Each sequence of feature coefficients is normalized and then applied through a parallel network which includes a set of linear and non-linear transformations. The network is bootstrapped so as to map the input coefficients into a conventional feature vector which includes cepstrum and energy along with their first and second order time derivatives. In the case of cepstrum/energy being the input to the network, the transformation is bootstrapped with an identity matrix that performs a self-mapping. When the input coefficients are log mel spectrum/energy[3], then the network is bootstrapped with a discrete cosine transformation that computes cepstrum and their higher order derivatives. In either case, the problem is strictly linear and therefore only the linear transform is set, while the non-linear transform is initialized with small random values. The combined output from both transforms is finally passed to the recognizer which adopts context-dependent HMMs that are initialized through ML training.

The feature transformation network can be considered as a link for providing an interaction between feature analysis and speech recognition. As illustrated in Fig. 1, the parameters of the transformation network as well as the HMMs are trained discriminatively using a unified objective function that aims to maximize class separability. In the current study, class-specific feature transformations have been employed such that a different transformation can be associated with each state, or unit, or word, etc. The framework for discriminative feature and model design is described next.

3. DISCRIMINATIVE FEATURE AND MODEL DESIGN

Let $X = \{X_1, X_2, .., X_T\}$ be a sequence of feature vectors belonging to string class C_i. The objective in MCE training is to maximize class separability by minimizing the class loss function [5]

$$J = \sum_i \mathcal{S}\{d_i(\mathcal{F}(X,\Psi);\Lambda)\} \cdot 1(X \in C_i), \qquad (1)$$

where $1(\cdot)$ is an indicator function, $\mathcal{S}\{\cdot\}$ is a sigmoid non-linear activation function, $d_i(\cdot)$ is a *misclassification* measure for string class i, $\mathcal{F}(X,\Psi)$ is a feature transformation network with corresponding set of parameters Ψ, and Λ are the parameters of the HMM model. In this study, minimizing the loss function in Eqn. (1) will be achieved by optimizing Ψ and Λ.

The misclassification measure $d_i(\cdot)$ is essentially a normalized log likelihood ratio between the correct class i and other classes competing with i. It is defined as

$$d_i(\mathcal{F}(X,\Psi);\Lambda) = -g_i(\mathcal{F}(X,\Psi);\Lambda) + G_i(\mathcal{F}(X,\Psi);\Lambda), \qquad (2)$$

where $g_i(\cdot)$ is the log likelihood of the transformed vector $\mathcal{F}(X,\Psi)$ given the HMM model with parameters Λ. $G_i(\cdot)$ in Eqn. (2) represents an anti-discriminant function and includes the scores of the N-best strings to class i in a set S. It is defined as

$$G_i(\mathcal{F}(X,\Psi);\Lambda) = \log[\frac{1}{N}\sum_{j \in S} \exp\{\eta \cdot g_j(\mathcal{F}(X,\Psi);\Lambda)\}]^{\frac{1}{\eta}}, \quad \eta > 0 \qquad (3)$$

Minimizing the loss function in Eqn. (1) is achieved through gradient descent such that at the n^{th} iteration

$$\Gamma_n = \Gamma_{n-1} - \epsilon_n \frac{\partial J}{\partial \Gamma}|_{\Gamma=\Gamma_{n-1}} \quad \epsilon_n > 0, \qquad (4)$$

where $\Gamma = \{\Psi, \Lambda\}$ and $\partial J/\partial\Gamma$ is the gradient of the loss function J. The steps for updating the parameters of Λ are described in [5]. The process for updating the parameters of the transformation network Ψ is described next.

Feature transformation design

Combining a linear transformation of the features, represented by an affine, with a non-linear transformation that is represented by an ANN can be done in either cascade or parallel. In this study, we opted for the parallel approach since the initial mapping (i.e., cepstrum-to-cepstrum or spectrum-to-cepstrum) is essentially linear in nature. Therefore, bootstrapping using the parallel approach would maintain the initial performance of the system with no feature transformation. However, one practical problem in this design is that transformations may evolve at a different learning rate and, consequently, their integration may prove to be difficult. In this study, the two transformations have been combined through an additional network which uses a linear activation function.

The functional transformation $\mathcal{F}(X,\Psi)$ includes M individual networks, $\{\Psi_i\}_{i=1,M}$, each consisting of weight matrices $[\mathcal{A}_i, \mathcal{B}_i, \mathcal{C}_i]_{i=1,M}$ and offset vectors $[a_i, b_i, c_i]_{i=1,M}$. As pointed out earlier, each network may be associated with a state, unit, word, etc. The linear transformation module is essentially an affine which produces an output vector $L_t = \mathcal{A}_i \cdot X_t - a_i$ at time t. The non-linear transformation module is a single layer ANN having a sigmoid activation function, $\mathcal{S}(\cdot)$, and producing the output vector $H_t = \mathcal{S}(\mathcal{B}_i \cdot X_t - b_i)$. The outputs of both transformations, namely, $LH_t = [L_t, H_t]$, are then combined using an additional network which applies the functional transformation $\mathcal{F}(X_t, \Psi_i) = \mathcal{C}_i \cdot LH_t - c_i$. This entire structure for feature transformation is a special case of a two-layer ANN in which non-linearity is being strictly applied to a limited number of hidden nodes.

Minimizing the loss function in Eqn. 1 by optimizing the parameters Ψ can be achieved through a chain rule. Let p, q, r be the indices for the outputs of the feature transformation network at the three layers, namely, input, hidden and output. Then

$$\frac{\partial J}{\partial \Psi_i} = \sum_r \frac{\partial J}{\partial \mathcal{F}^r} \cdot \frac{\partial \mathcal{F}^r}{\partial \Psi_i}. \qquad (5)$$

The partial derivative $\partial J/\partial\mathcal{F}^r$ is defined in [8]. $\partial\mathcal{F}^r/\partial\Psi_i$ is computed as follows: $\frac{\partial \mathcal{F}^r}{\partial \mathcal{C}_i^{q,r}} = LH_t^q$, $\frac{\partial \mathcal{F}^r}{\partial c_i^r} = -1$, $\frac{\partial \mathcal{F}^r}{\partial \mathcal{A}_i^{p,q}} = \mathcal{C}_i^{q,r} \cdot X_t^p$, $\frac{\partial \mathcal{F}^r}{\partial a_i^q} = -\mathcal{C}_i^{q,r}$, $\frac{\partial \mathcal{F}^r}{\partial \mathcal{B}_i^{p,q}} = \mathcal{C}_i^{q,r} \cdot X_t^p \cdot \mathcal{S}_q^{'}$, $\frac{\partial \mathcal{F}^r}{\partial b_i^q} = -\mathcal{C}_i^{q,r} \cdot \mathcal{S}_q^{'}$,

where $\mathcal{S}_q^{'}$ denotes the first-order derivative of the loss function $\mathcal{S}_q$ on the q^{th} output node of the ANN.

4. DATABASE AND BASELINE SYSTEM

A speaker-independent telephone-based connected digits database was used in this study. Utterances, ranging from one to sixteen digits in length, were extracted from different field-trial collections with varied environmental conditions and transducer equipments. The database was divided into 9500 strings for training and 960 strings for testing. The average string length was 11.1 digits.

The baseline system is similar to that shown in Fig. 1 without the feature transformation module. Feature analysis is performed as follows. An input utterance is first segmented at every 10 msec intervals into frames of 30 msec duration. Each frame is then processed to give 12 mel-based cepstral coefficients along with a normalized log energy coefficient. The 13-dimensional vector is also augmented with its first and second order time-derivatives, resulting in a vector of 39 features per frame.

Following feature analysis, each feature vector in the baseline system is directly passed to the recognizer which models each word (i.e., digit) in the vocabulary by a set of left-to-right continuous-density quasi-triphonic HMMs [7]. Each word is divided into three units, namely, head, body and tail. To model inter-word coarticulation, each word is made to have a single body with multiple heads and tails, resulting in a total of 274 subword models. Each subword model consists of 3 to 4 states, with each state having a mixture of Gaussian components. Training included updating all the parameters of the model, namely, means, variances and mixture gains using ML estimation followed by MCE to further refine the estimate of the parameters.

Table 1: Percentage word error rate (Wd_Er) and percentage string error rate (St_Er) for the baseline system and for systems introducing discriminative feature transformation.

	1 mix/state		**4 mix/state**	
System	Wd_Er	St_Er	Wd_Er	St_Er
Base-ML	6.4	36.1	3.6	23.3
Base-MCE	4.0	24.2	2.3	15.1
Affine1	3.7	23.1	X	X
Affine12	3.2	20.8	X	X
Affine/ANN1	3.4	21.8	2.2	14.9
Affine/ANN12	2.6	16.7	1.8	12.1
Spectrum1	X	X	X	X
Spectrum12	X	X	X	X

Table 1 presents the word and string error rates for the baseline system when using either one or four Gaussian components per state.[1] These results include insertion, deletion and substitution errors when running with a free grammar network. Baseline results following model training with ML (labeled as "Base-ML") and with MCE (labeled as "Base-MCE") are displayed. A drop in the word error rate of about 36% is obtained when employing discriminative training versus ML alone.

[1] A one-state silence/background model is used with 32 mixture components.

5. EXPERIMENTS WITH FEATURE TRANSFORMATION

The purpose of the experiments presented in this section is two-fold:

1. Investigate the effect of introducing non-linear in addition to linear discriminative feature transformations.
2. Compare the performance of the transformation network when spectrum, as opposed to cepstrum, is used at the input layer.

In the first set of experiments, we applied a linear affine transformation on each frame of the original 39-dimensional feature vector. The parameters of the network were bootstrapped with an identity mapping thus maintaining the initial performance of the recognizer. As pointed out in [8], this transformation helps to reflect the correlation that may exist in the empirical cepstral coefficients which is particularly important since our ASR system adopts diagonal covariances, rather than full covariances, when estimating the observation probabilities.

There are clearly several strategies for joint optimization of the feature and model parameters. One could, for example, optimize the feature network as well as the model parameters simultaneously. The different learning characteristics of the two systems, however, create difficulties in achieving convergence. In this study, discriminative feature and model design was conducted iteratively by first optimizing the network parameters while freezing the model parameters, and then vice versa. The results of this procedure are shown in Table 1 when employing a single network (labeled as "Affine1") and 12 networks, one for each vocabulary word (labeled as "Affine12"). As we have already established in [8], introducing multiple networks seems to provide an additional improvement in recognition performance over a baseline system not employing feature transformations.

The second set of experiments were conducted to evaluate the system shown in Fig. 1. The affine transformation was bootstrapped in the same way as that described in the previous experiment. The ANN transformation included 39 output nodes having a sigmoid activation function. As described in Section 3, the outputs of both transformations were combined using an additional network with 39 output nodes. The initial performance of the recognizer was maintained by setting the parameters of the ANN to small random values.

The strategy we selected for training the system in Fig. 1 is similar to that used in the previous experiment. However, due to the different learning rates of the affine and the ANN, optimization was performed by updating the affine, followed by the ANN and finally the combined system. This procedure was iterated and combined with model training. The results when using a single network (labeled as "Affine/ANN1") and 12 networks (labeled as "Affine/ANN12") are shown in Table 1. Clearly whether using one or four mixture components per state, with either a single or 12 networks, the non-linear transformation seems to provide an additional improvement in recognition accuracy. This improvement can be up to 35% reduction in the word error rate over the baseline system.

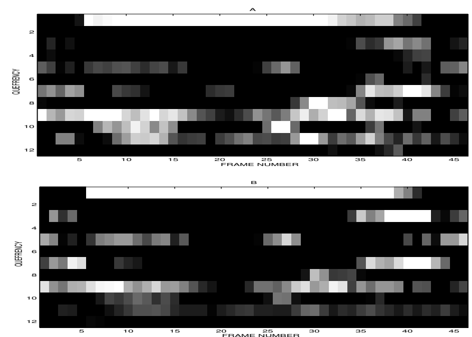

Figure 2: "Cepstrogram" for the utterance "four": (A) original mel-based cepstrum before feature transformation, (B) after feature transformation.

To provide some light into what the transformation is doing to the features, Fig. 2 shows a "cepstrogram" of the first twelve inputs to the network (representing mel cepstrum) and the first twelve outputs of the network (representing the transformed features). These plots correspond to the digit 'four' which has been incorrectly recognized as 'zero one' by the base system but correctly recognized when introducing the feature transformation. Although it is difficult to judge why the cepstrogram in (B) resulted in the correct decision, it is clear, however, that certain coefficients have been emphasized by the transformation while others have been deemphasized. This is reflected by the intensity of plots. The cepstrogram in (B) also appears less noisy than that in (A). More detailed analysis of these plots may reveal some interesting characteristics of this transformation.

The last set of experiments was conducted to compare the performance of the recognizer when using spectrum as opposed to cepstrum as inputs to the transformation network. Rather than using a cepstral feature vector of 39 coefficients, we employed a spectral feature vector of 72 coefficients. This includes 23 log mel filter bank spectral energies and a normalized energy coefficient along with their first and second order time derivatives. The nature of this mapping not only reflect the correlation among the features but also the transformation from spectrum to cepstrum. Discriminative feature and model design was performed in the same way as for the previous experiment.(NOT COMPLETE)

6. SUMMARY AND FUTURE WORK

This paper proposed a framework for discriminative feature and model design. The intent was to integrate feature extraction and model training under a unified objective function that relates to the recognition error rate. Accordingly, discriminative training based on an MCE criterion was adopted for designing a set of parallel networks performing feature transformation and a set of HMMs performing speech recognition. An integrated approach to feature transformation was described in this paper which includes a linear affine network and a non-linear ANN. Experimental results on a connected digits task show that (a) feature transformation using class-specific affine networks has the potential of reducing the word error rate by about 20%, (b) introducing non-linear transformation on the features provides additional benefits to recognition and can reduce the word error rate by up to 35% over a baseline system not incorporating feature transformation, (c) setting the transformation to accept spectrum as opposed to cepstrum leads to similar improvement in recognition performance.

The particular set-up of our system provides a framework for generating task-specific features and models. The feature transformation, however, is sensitive to the bootstrapping procedure and may already be restricted by the nature of the features used in the mapping, i.e., cepstrum. A new framework should be considered to handle larger contexts and to take advantage of the flexibility of non-linear feature transformation. We are currently experimenting with convolutional networks which we believe to be more suitable for this task.

7. REFERENCES

1. L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer. Maximum mutual information of hidden Markov model parameters for speech recognition. In *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pages 49–52, 1986.
2. Y. Bengio, R. De Mori, G. Flammia, and R. Kompe. Global optimization of a neural network - hidden Markov model hybrid. *McGill University Technical Report*, TR-SOCS-90.22, 1990.
3. R. Chengalvarayan and L. Deng. HMM-based speech recognition using state-dependent, discriminatively derived transforms on mel-warped dft features. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 5:243–256, 1997.
4. S. Euler. Integrated optimization of feature transformation for speech recognition. In *Proc. Eurospeech '95*, pages 109–112, 1995.
5. B.-H. Juang and S. Katagiri. Discriminative learning for minimum error classification. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 40:3043–3054, 1992.
6. K. K. Paliwal, M. Bacchiani, and Y. Sagisaka. Minimum classification error training algorithm for feature extractor and pattern classifier in speech recognition. In *Proc. Eurospeech '95*, pages 541–544, 1995.
7. R. Pieraccini and A. E. Rosenberg. Coarticulation models for continuous digit recognition. In *Proc. Acoust. Soc. Am.*, page 106, May 1990.
8. M. Rahim and C.-H. Lee. Simultaneous ANN feature and HMM recognizer design using string-based minimum classification error(MCE) training. In *Proc. ICSLP '96*, pages 1824–1827, 1996.
9. B. Widrow and E. Walach. *Adaptive inverse control*. Prentice Hall, Englewood Cliffs, NJ, 1995.