NEURAL NETWORK BASED OPTIMAL FEATURE EXTRACTION FOR ASR

Narada D. Warakagoda and Magne H. Johnsen

Department of Telecommunications
Signal Processing group
NTNU, O.S. Bragstad Plass 2B
N-7034, Trondheim, Norway
warakago@tele.ntnu.no
mhj@tele.ntnu.no

ABSTRACT
The procedure of calculating Mel Frequency based Cepstral Coefficients (MFCC) is shown to resemble a three layer Multilayer Perceptron (MLP) like structure. Such an MLP is employed as a preprocessor in a hybrid HMM-MLP system, and the possibility of optimizing the whole system as a single entity, with respect to a suitable criterion, is pointed out. This system, together with the Maximum Mutual Information (MMI) criterion was tested on a speaker independent, five broad class, isolated phoneme recognition task. Results of these preliminary experiments, which clearly indicate the advantage of optimizable preprocessing, are reported.

1. INTRODUCTION
From the pattern recognition point of view, an Automatic Speech Recognition (ASR) system can be considered to consist of two cascade elements; a preprocessing (or parameter extraction) stage and a classification stage. In a typical speech recognizer, classification is carried out with the help of a language model and an acoustic model. The preprocessor’s job is to deliver feature vectors extracted from the input signal to the classifier, whose decisions are heavily dependent upon the discriminative quality of these features. Therefore the preprocessor’s ability to pick up discriminative information is a critically important factor for the overall performances of an ASR system.

Widely used preprocessing schemes today such as Perceptual Linear Prediction (PLP), Mel Frequency derived Cepstral Coefficients (MFCC) and Mel Frequency Filter Bank Amplitudes (MFFBA) etc., are all rooted in crude spectral models of human speech production and perception. In other words, these procedures cannot be considered as accurate reflections of the complex nonlinear processes involved in the human speech communication, but rather “engineering approximations” to them. Even considered from an engineering point of view, they do not seem to posses some kind of optimality with respect to a relevant criterion. This is because in usual practice of ASR system training, only the classifier stage parameters (or some of them) are optimized, while the preprocessing stage is kept untouched. However with the view that preprocessing is an integrated part of the ASR system, the preprocessing stage too should be included in the training process. There are some reported work in this direction, and one which is worthwhile to be mentioned here is the so called Discriminative Feature Extraction (DFE) [1, 2]. In DFE, however, focus is mainly on the filter bank, which is only a part of the preprocessor. In this paper, we follow a bit broader approach and consider the optimization of the whole preprocessing stage which is viewed as a nonlinear mapping from the space of the frames of raw speech to the feature space.

The rest of the paper is organized as follows. In section 2 we show that computing steps of the popular MFCCs can be seen as a 3-layer MLP like structure. A hybrid HMM-MLP system incorporating a preprocessor MLP is described in the section 3. Sections 4 and 5 contain respectively the details of the experiments performed and a discussion about the results. In section 6 we give some final comments.

2. MFCC-EXTRACTION AS AN MLP
Let’s consider the MFCC derivation procedure for a speech frame \( s_t(n), n = 0, \ldots, N - 1 \), where \( N \) is the number of samples. Note that \( t = 0, \ldots, T - 1 \) indicates the frame number, with \( T \) as the total number of frames. The first step of the procedure is to take the magnitude \( S_t(k) \) of the \( L \)-point Discrete Fourier transform of the weighted speech frame, as in eqn. 1.

\[
S_t(k) = \sum_{n=0}^{N-1} w(n) s_t(n) \exp\left(-\frac{2\pi ink}{L}\right), \quad 0 \leq k < L \quad (1)
\]

where \( w(n) \), \( 0 \leq n \leq N - 1 \) represents a weighting (windowing) scheme, typically a Hamming window.

The next step is to send \( S_t(k) \), through a \( P \)-channel filter bank linearly arranged in the Mel frequency scale, whose \( i \)-th filter is defined by \( H_i(k), k = 0, \ldots, L - 1 \) in the frequency domain. We then take the logarithm of the result for preparing it for the next step. These operations can be combined in eqn. 2.

\[
\tilde{S}_t(i) = \log \left( \frac{1}{T} \sum_{k=0}^{L-1} H_i(k) S_t(k) \right), \quad 0 \leq i < P \quad (2)
\]

Finally we take the inverse Fourier transform or equivalently the cosine transform of \( \tilde{S}_t(i) \), since it is real symmetric, to obtain the MFCCs, \( c_t(m) \).

\[
c_t(m) = \sum_{i=0}^{P-1} \tilde{S}_t(i) \cos\left(\frac{2\pi im}{P}\right), \quad 0 \leq m < M \quad (3)
\]

Equations 1, 2 and 3 can be interpreted as the governing equations of a Multilayer Perceptron (MLP) with three layers, whose input and output are \( s_t(n) \) and \( S_t(k) \) respectively.
To see this, recall the equation which describe a single layer (of size $Q \times R$) of an MLP [3],

$$y(j) = F \left( \sum_{i=0}^{Q-1} w_{ij} x(i) \right), \quad j = 0, \ldots, R - 1 \quad (4)$$

Here $w_{ij}$, $x(i)$ and $y(j)$ represent the weights, inputs and outputs of the layer respectively, whereas $F()$, in general, is a non-linear function known as the activation function. By comparing eqns. 1, 2 and 3 with eqn. 4 we observe that $w(n) \cdot \exp(-\frac{2\pi n k}{L})$, $H_k$ and $\cos(\frac{2\pi n k}{L})$ represent weights of the layers 1, 2 and 3 respectively. Also it is clear that layer 3 has a linear activation function, while layers 2 and 3 have $|\cdot|$ and $\log |\cdot|$ as the respective activation function.

According to this interpretation however, weights of the layer 1 take complex values. Even though this is not a forbidden case in the theory of MLPs, in order to make the computational procedures as simple as possible, we would like to have real valued weights in our MLP. In the following we will describe two approaches through which this can be achieved. Since the source of the complex terms in the equations lie in the complex valued kernel of the Fourier Transform, one obvious way to avoid those is to use a Fourier-like transform with a real kernel instead. The transform known as Hartley Transform [4] fits nicely to this requirement, whose kernel is given by,

$$\cos(\frac{2\pi n k}{L}) = \cos(\frac{2\pi n k}{L}) + \sin(\frac{2\pi n k}{L}) \quad (5)$$

The quantity $\cos(\frac{2\pi n k}{L})$ is referred to as Hartley argument in this paper. Further we refer to $\frac{2\pi n k}{L}$, $n$ and $k$ as Hartley angle, time index and frequency index respectively. It is known that Hartley and Fourier transform have many interesting relationships and similar properties [4]. However the Hartley transform is more phase sensitive than the Fourier transform, and therefore offers an inferior shift invariance.

With this transform we can rewrite the eqn. 1 as in eqn. 6.

$$S_L(k) = \left[ \sum_{n=0}^{N-1} w(n) s_t(n) \cos(-\frac{2\pi n k}{L}) \right]^2, \quad 0 \leq k < L \quad (6)$$

Thus eqns. 6, 2 and 3 now define a 3-layer real valued MLP, (approximately) representing the MFCC calculation procedure. In the following we call it MFCC-CAS-MLP for easy reference.

The other approach to avoid the complex terms involved in MFCC calculation, is to rewrite the eqn. 1 as follows.

$$S_L(k) = \left\{ \begin{array}{ll}
\left[ w(n) s_t(n) \cos(\frac{2\pi n k}{L}) \right]^2 & , \quad 0 \leq k < L \\
\left[ w(n) s_t(n) \sin(\frac{2\pi n k}{L}) \right]^2 & , \quad 0 \leq k < L
\end{array} \right. \quad (7)$$

We see that each of the terms on the right hand side of eqn. 7 represents a layer of a real valued MLP. Therefore the whole operation given by this equation can be thought of as a parallel combination of two layers of neurons. This is schematically shown in fig. 1. A representation for whole MFCC calculation can be obtained by serially adding the two layers given by eqns. 2 and 3 to this structure. For convenience we call this structure MFCC-FT-MLP.

As a summary, in this section we described two MLP (or MLP-like) architectures which can represent the MFCC calculation procedure. From this it is clear that MFCC calculation is really a non-linear mapping from the speech frame space to the feature space. Since this mapping is now represented in a familiar form, it can easily be optimized in a training session, together with the classification stage parameters, using for example a gradient descent algorithm. Note that in the following we use the common name MFCC-MLP to denote these MLPs.

### 3. DESCRIPTION OF THE SYSTEM

Since the preprocessing part can be generalized to a trainable MLP-like structure, it is desirable if the classification part of the system too is interpreted as a neural net. As a starting point to do this, assume that our classifier consists of multiple mixture, continuous density, left to right HMMs with no skips. Operation of the system can be considered either in a Maximum Likelihood (ML) or discriminative setting. We prefer the latter, and select the Maximum Mutual Information (MMI) criterion for training [5]. In this case the classifier is a single global HMM representing all the classes. Calculation of the mutual information can be done by applying the forward algorithm to this HMM. This operation can be interpreted as a recurrent net, known as the Alpha-net [6]. The input to the Alpha-net must be the observation densities, which is usually provided by the Gaussian mixtures. However, Gaussian mixtures represent simply a mapping from observation space to the observation density space. To enhance the generality of this mapping one can use an MLP instead of the Gaussian mixtures [7, 8]. This is called PROB-MLP henceforth in this paper.

Note that the input $o_t$ to the PROB-MLP, comes from the preprocessor. That is,

$$o_t = [c_t(0), c_t(1), \ldots, c_t(M - 1)] \quad (8)$$

The whole system can be shown schematically as in fig. 2.

The objective function to be minimized

$$E_{M_M} = - \sum_{\tau=1}^{N_{\tau}} \log P_\theta(w^\tau | s^\tau) \quad (9)$$

can be expressed in terms of the output of the Alpha-net as in [6]. In eqn. 9, $P_\theta(w^\tau | s^\tau)$ is the conditional probability of the class $w^\tau$ to which the $\tau$th speech signal $s^\tau$ of the training
set belongs. $\mathcal{N}$ is the total number of speech signals in the training set, and $\theta$ represents the parameters of the whole system. Derivatives of $E_{MM}$ can be calculated as in [6] and can be spread through the whole system using the back-propagation algorithm. In this way a truly global optimization is possible, which includes even the preprocessing part.

4. EXPERIMENTS AND RESULTS

Results of some preliminary experiments carried out to evaluate the system, are reported here. In order to keep the simulation time at a reasonably low level, 5 broad class [9] isolated phoneme recognition task on the TIMIT data base was selected. Two small training and test sets representing all dialect regions and a considerable speaker variability, were created in the following way. First a single SI sentence from each of the 462 train set speakers were selected. From this, two disjoint sets of 100 sentences, each representing all dialect regions, are extracted. Those two were then used as the trainset and testset.

Two series of experiments were designed to investigate mainly the effects of the optimization of the preprocessing part, one for MFCC-CAS-MLP and the other for MFCC-FT-MLP. Common features of the experiments are as follows.

Input Speech Preparation: In the experiments input speech sequences are prepared by applying a 50% overlapping rectangular window of length 400 samples (at 16kHz rate).

System dimensions: MFCC-MLPs are dimensioned so that they correspond to the calculation of a 12-dimensional MFCC vector, from a 400 samples long speech frame, through 128-point Hartley/Fourier transform and a 23-channel filter bank. PROB-MLP has a two layer architecture with dimensions 12x60x15. The first layer of this MLP has a tan-sigmoid nonlinearity while the second layer has a softmax nonlinearity. The Alpha-net contains 5 HMMs (representing 5 broad classes) each of which is a 3-state left to right model with no skips. No transitions are allowed between HMMs, because they are used in the isolated mode.

Initialization: Transition matrices of the each HMM in the Alpha-net are initialized using previous results of a conventional HMM recognizer. Initial weights for the PROB-MLP are chosen from a random normal distribution with mean 0.5 and variance 0.0005. Weights of the MFCC-MLPs are initialized in accordance with eqns. 6, 7), 2 and 3. In particular, in the layer 1, weighting function corresponding to the Hamming window is used. Further, for the layer 2 (filter bank layer), weights are obtained by sampling a Gaussian-shaped Mel scale critical band filter bank uniformly at 128 points in the acoustic frequency scale.

Training and Testing: Training is carried out for 100 epochs in the batch mode. After each epoch, system parameters are updated using the algorithm proposed by Silva and Almeida [10]. For both training and testing, percentage number of the correctly recognized phonemes is reported.

Following experiments were performed with each of the two MFCC-MLPs. The differences among experiments are due to the ways the parameters of the MFCC-MLP is adjusted during training. Across the training procedures for all these experiments, Alpha-net and PROB-MLP are made to have the same architecture and the adaptation strategy.

<table>
<thead>
<tr>
<th>Experiment name</th>
<th>Recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO_MFCC_ADP</td>
<td>72.54</td>
</tr>
<tr>
<td>FULL_MFCC_ADP</td>
<td>90.56</td>
</tr>
<tr>
<td>FRQ_MFCC_ADP</td>
<td>82.20</td>
</tr>
<tr>
<td>FB_MFCC_ADP</td>
<td>81.43</td>
</tr>
<tr>
<td>REC_MFCC_ADP</td>
<td>83.53</td>
</tr>
</tbody>
</table>

Table 1: Training and Testing results for MFCC-CAS-MLP as the preprocessor.

Exp1 (NO\_MFCC\_ADP): The MFCC-MLP is kept non-adaptive in this experiment. This is done by freezing the parameters of this part of the system, at their initial values.

Exp2 (FULL\_MFCC\_ADP): In this experiment, all possible parameters of the MFCC-MLP are adapted during training. However some constraints are imposed on the adaptation, so that a physical interpretation is possible even after training. In the first layer, each weight is updated indirectly through the two constituent quantities, namely the windowing coefficient $w(n)$ and the corresponding Hartley/Fourier angle (see eqns. 6 and 7). In the second layer again (see eqn. 2), all the filter coefficients are indirectly adapted, through a set of free auxiliary variables. As mentioned earlier, filter coefficients are obtained by sampling a set of Gaussian shaped filters. Each of these filters is characterized by three free variables, namely the center frequency (mean), the bandwidth (variance) and the height, which are the auxiliary quantities those allowed to be adjusted. This is similar to the approach in [1]. Finally, the whole cosine is adjusted as a single weight entity, without any constraints, in the third layer (see eqn. 3).

Exp3 (FRQ\_MFCC\_ADP): The procedure here is almost identical to that in FULL\_MFCC\_ADP, except for a difference in the first layer. In this case the Hartley/Fourier frequency index is adjusted, instead of the Hartley/Fourier angle. This causes a huge reduction of the number of free parameters of the system.

Exp4 (FB\_MFCC\_ADP): The difference of this experiment from FRQ\_MFCC\_ADP is that no parameter of the Hartley/Fourier transform layer is updated. That is, only the Filter bank layer and the Cosine transform layer are adapted during training.

Exp5 (REC\_MFCC\_ADP): This is an extended version of FRQ\_MFCC\_ADP, where the Hartley/Fourier transform layer is modified by adding a local feedback path to each of the neurons. Details of this recurrent neural architecture and training algorithms are described in [11]. The feedback paths of the net give some kind of memory to the system, making the current feature vector depend not only on the current speech frame, but also on the previous ones. Therefore the network represents a mapping from a space formed by concatenated speech frames up to now, to the feature space. This is obviously a more general mapping than that of the original system.

In all these experiments, training algorithm converges nicely, to an acceptable minimum during the 100 epochs. The resulting recognition rates, for both training and testing are given in tables 1 and 2.
### Table 2: Training and Testing results for MFCC-FT-MLP as the preprocessor

<table>
<thead>
<tr>
<th>Experiment name</th>
<th>%Recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO-MFCC-ADP</td>
<td>81.21</td>
</tr>
<tr>
<td>FULL-MFCC-ADP</td>
<td>92.36</td>
</tr>
<tr>
<td>FRQ-MFCC-ADP</td>
<td>84.42</td>
</tr>
<tr>
<td>FB-MFCC-ADP</td>
<td>83.37</td>
</tr>
<tr>
<td>REC-MFCC-ADP</td>
<td>86.53</td>
</tr>
</tbody>
</table>

### 5. DISCUSSION

A major interest is to compare the performances of systems with optimized preprocessors against those with conventional, non-optimized preprocessors (i.e. NO-MFCC-ADP with the others). As we can see, all systems with adjustable preprocessing perform better than NO-MFCC-ADP, in both cases, MFCC-CAS-MLP and MFCC-FT-MLP. However full adaptation of the preprocessor (FULL-MFCC-MLP) gives only slight improvements in testing, though it’s recognition rate is very high in training. This implies a poor generalization ability, which is due to it’s very large number of free parameters. In FRQ-MFCC-ADP, we have a significantly lesser number of free parameters and this brings in a high improvement in recognition rate over FULL-MFCC-ADP in testing.

Another noticeable point is that adaptation of only the filter bank and Cosine transform stages in training (FB-MFCC-ADP), gives almost as good results as in controlled adaptation of all layers in MFCC-MLP (e.g. FRQ-MFCC-ADP). A closer look on results however reveals that FRQ-MFCC-ADP is superior in training. Therefore it is clear that FB-MFCC-ADP has an advantage in generalization ability, which is due to somewhat lesser number of free parameters it contains than does FRQ-MFCC-ADP. Thus we cannot rigidly conclude that optimization of the Hartley/Fourier transform layer does not bring some useful information into the system. For example, it may be possible to achieve better results, using a bigger data set in training.

Among the optimizable preprocessing systems, the one which gives the best performance is, as expected, REC-MFCC-ADP. The additional memory introduced by the recurrent loops has helped in this case to improve the recognition rates.

An overall comparison of Fourier transform based systems (those with MFCC-FT-MLP) with Hartley transform based systems (those with MFCC-CAS-MLP) leads to the conclusion that FT based systems perform better. This is not an unexpected result, because FT is less phase sensitive than the Hartley transform. Therefore an FT based front end can offer a higher shift invariance, and hence better recognition rates.

### 6. CONCLUDING REMARKS

Optimizable preprocessing relates directly to one of the most fundamental problems of speech recognition, namely what features of the speech signal are the best wrt the discrimination of different sound classes. This kind of approach is specially suited for noisy speech recognition [2], and for cases where the language model is poor. Therefore it will be interesting to use the approach presented here on a such kind of problem. Even though our starting point in this paper was MFCC calculation, optimizable preprocessing schemes rooted in other philosophies are possible. This is another avenue for further work along the direction of optimizable preprocessing.

### 7. REFERENCES


