Open access • Journal Article • DOI:10.1109/TASL.2010.2045943

Analysis of MLP-Based Hierarchical Phoneme Posterior Probability Estimator

Joel Praveen Pinto, Sivaram Garimella, Mathew Magimai-Doss, Hynek Hermansky ...+1 more authors

Institutions: Idiap Research Institute, Johns Hopkins University

Published on: 01 Feb 2011 - IEEE Transactions on Audio, Speech, and Language Processing (IEEE)

Topics: Multilayer perceptron, Posterior probability and TIMIT

Analysis of MLP Based Hierarchical Phoneme Posterior Probability Estimator

Joel Pinto, G.S.V.S. Sivaram, Mathew Magimai.-Doss, *Member, IEEE,*
Hynek Hermansky, *Fellow, IEEE,* and Hervé Bourlard, *Fellow, IEEE.*

***Abstract*—We analyze a simple hierarchical architecture consisting of two multilayer perceptron (MLP) classifiers in tandem to estimate the phonetic class conditional probabilities. In this hierarchical setup, the first MLP classifier is trained using standard acoustic features. The second MLP is trained using the posterior probabilities of phonemes estimated by the first, but with a long temporal context of around 150-230 ms. Through extensive phoneme recognition experiments, and the analysis of the trained second MLP using Volterra series, we show that (a) the hierarchical system yields higher phoneme recognition accuracies - an absolute improvement of 3.5% and 9.3% on TIMIT and CTS respectively - over the conventional single MLP based system, (b) there exists useful information in the temporal trajectories of the posterior feature space, spanning around 230 ms of context, (c) the second MLP learns the phonetic temporal patterns in the posterior features, which include the phonetic confusions at the output of the first MLP as well as the phonotactics of the language as observed in the training data, and (d) the second MLP classifier requires fewer number of parameters and can be trained using lesser amount of training data.**

***Index Terms*—Multilayer perceptrons, Volterra series, hierarchical systems, posterior probabilities.**

I. Introduction

MUltilayer perceptron (MLP) classifier based acoustic modeling is being extensively used in state-of-the-art automatic speech recognition (ASR) systems [1][2][3][4][5]. The MLP is typically trained using standard acoustic features such as mel frequency cepstral coefficients or perceptual linear predictive coefficients with a certain temporal context. A well trained MLP can estimate the posterior probabilities of the output classes, typically subword units of speech such as phonemes, conditioned on the input features [6][7].

MLP based acoustic modeling has certain benefits. Firstly, it obviates the need for strong assumptions on the statistics of the features and the parametric form of its density function. As a consequence, features with different distributions can be simply concatenated and applied at the input of the MLP to achieve feature combination [3]. Secondly, when trained on large amount of data, MLPs have been shown to be invariant to speaker characteristics [3] and environment specific information such as noise [8]. Thirdly, the output of the MLP are probabilities with useful properties (*e.g.,* positivity, summing to one), providing an efficient framework for multi-stream combination [9]. Lastly, the MLP can be trained efficiently and is scalable with large amount of data.

The phonetic class conditional probabilities estimated by the MLP are used in hidden Markov model (HMM) based ASR in different ways. In the hybrid HMM/MLP system [6], they are used as local emission scores in the HMM states. In the Tandem system [10], they are transformed by applying logarithm followed by Karhunen-Loeve transformation (KLT), and used as features to a standard HMM/GMM system. In a recent study [11], the estimated posterior probabilities are used directly as features in an HMM based system, where the state emission distribution is multinomial. Throughout this paper, whenever the phoneme posterior probabilities are used as local representation of speech in place of standard acoustic features, we refer to them as *posterior features*.

In the posterior feature space, each dimension corresponds to a phoneme. The posterior feature vector at a particular time instant is a point in the posterior feature space, representing the instantaneous soft-decision on the underlying phoneme. It carries useful information such as the probability mass assigned to the competing phonemes. The sequence of posterior feature vectors is a trajectory in the posterior feature space, and it can provide additional contextual information such as the evolution of the posterior features within a phoneme (sub-phonemic transition). Furthermore, a sufficiently long temporal context on the posterior features can also capture the transition to/from neighboring phonemes (sub-lexical transition).

The contextual information in the posterior features has been successfully exploited in ASR in our previous studies [12][13], where a second MLP classifier was trained on the posterior features with a temporal context of 150-230 ms. This hierarchical approach yielded higher phoneme recognition accuracies when compared to the conventional single MLP based approach. This paper is an extension to our previous work [12], and the main focus is on the analysis of the hierarchical system. We investigate the reasons for the effectiveness of the hierarchical system and attempt to understand the functionality (or working) of the second MLP classifier by analyzing its trained parameters.

As the second MLP is trained using posterior features with a certain temporal context, we can expect it to learn the phonetic-temporal patterns, mainly capturing the phonetic confusions at the output of the first classifier. However, as the MLP is a complex classifier with nonlinear activation functions, discovering the phonetic-temporal patterns learnt by the system for each phoneme is not straightforward. Moreover, as the MLP is trained using a discriminative criterion, these patterns cannot be simply derived from the confusion matrix of the first MLP classifier. In addition, confusion matrices do not capture any temporal information. To understand this information, one has to interpret the trained parameters (weights

and biases) of the second MLP classifier.

In this work, we address this issue by representing the second stage of the hierarchical system using Volterra series [14][15], thereby decomposing the trained nonlinear system into its linear, quadratic, and higher order parts. Furthermore, we analyze the linear part of the second MLP and interpret the phonetic-temporal patterns that are learned. In contrast, our previous study [12] utilized a single layer perceptron in place of the second MLP to facilitate easy analysis. While preliminary insights into the working of the system were obtained by plotting its weight matrix, the actual MLP that was used in ASR studies remained unanalyzed.

Other extensions to our previously published work include a study on the role of temporal context on the posterior features, and its effect on the performance of the hierarchical system. We also analyze some of the useful properties of posterior features such as lesser nonlinguistic variabilities and sparse representation, and discuss its influence on the complexity of the second MLP classifier and the amount of training data. Experiments are also performed on conversational telephone speech (CTS) to ascertain if the trends in results and analysis concur with those obtained on TIMIT.

Through extensive phoneme recognition studies and the analysis of second MLP in the hierarchical system using Volterra series, we show that (a) the hierarchical system yields higher phoneme recognition accuracies compared to a single MLP based system, (b) the posterior features contain useful contextual information spanning around 150-230 ms of temporal context (c) the second MLP in the hierarchical system learns the phonetic-temporal patterns in the posterior features, which includes the phonetic confusion patterns at the output of the first classifier and to a certain extent the phonotactics of the language as observed in the training data, and (d) the classifier at the second stage of the hierarchy requires fewer number of parameters and lesser amount of training data.

The rest of the paper is organized as follows: In Section II, we describe the MLP based hierarchical system and discuss its similarities/differences with previous works in the literature. In Section III, we describe the experimental setup and the results. In Section IV, we introduce Volterra series and discuss its application in the analysis of the second stage of the hierarchical system. Furthermore, we also interpret the linear Volterra kernels of the system in terms of the phonetic-temporal patterns. In Section V, we analyze the properties of the posterior features that contribute to the effectiveness of the hierarchical system. In Section VI, we discuss some of the less explored facets of the hierarchical approach.

II. Hierarchical Posterior Estimation

An MLP classifier with enough complexity and trained with sufficient amount of data can directly estimate the Bayesian *a posteriori* probabilities of the output classes, conditioned on the input features [7]. Consequently, the performance of ASR systems using MLP acoustic models can be improved using the following three broad strategies: (a) using richer acoustic features (b) increasing the capacity of the MLP (but this approach is often limited by the amount of training data [16]) and (c) using finer representation of output classes such as sub-phonemic states [12] [17].

In this work, we explore a way to post-process the output of the MLP (posterior probabilities of phonemes, conditioned on acoustic features) to obtain new *enhanced* estimates of the phonetic class conditional probabilities.

A. Motivation

An MLP trained on acoustic features gives a frame-level phoneme classification accuracy of around 60-70%. The errors in classification can be mainly attributed to the limitations in feature extraction and modeling techniques. Analysis of the associated phonetic confusion matrices show that there exists a consistent pattern in classification. For example, if the phoneme /iy/ (*e.g.,* b**ea**t) is misclassified, then it is more likely that vowels such as /ih/ (*e.g.,* b**i**t) or /eh/ (*e.g.,* b**e**t) is assigned a higher probability mass. This information in the distribution of the probability values could be exploited to correct the output of the MLP classifier.

The posterior features have lesser nonlinguistic variabilities such as speaker and environmental characteristics when compared to acoustic features. In addition, they have a simpler (or sparse) representation. As a consequence, we hypothesize that contextual information spanning longer time spans can be effectively learned in the posterior feature space. The useful contextual information could be the evolution of the posterior features within a phoneme (sub-phonemic level) and its transition into the neighboring phonemes (sub-lexical level).

There have been attempts in the recent past to model the contextual information in the posterior features in an hierarchical fashion by using classifiers such as conditional random field (CRF) [18][19] or MLP [12][20]. In this work, we further investigate the MLP based hierarchical system [12]. As shown in Fig. 1, the first MLP is trained in the conventional way using standard acoustic features. The second MLP is trained using posterior features estimated by the first MLP classifier with a long temporal context of around 150-230 ms.

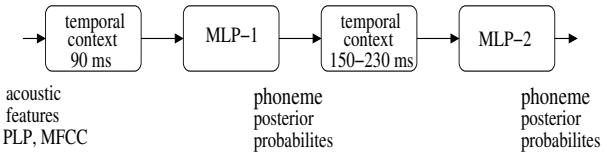


Fig. 1. Estimation of posterior probabilities of phonemes using an hierarchy of two MLPs. The second MLP is trained using the posterior probabilities of phonemes estimated by the first MLP.

B. Notations and Formalism

The following notations are used throughout this paper. $\mathbf{f_t}$ denotes the acoustic feature vector [1] at time t. A temporal context of $2d_1+1$ frames on the feature vector $\mathbf{f}_t$ is denoted by $\mathbf{f}_{t-d_1:t+d_1} = [\mathbf{f}'_{t-d_1}, \ldots \mathbf{f}'_t, \ldots \mathbf{f}'_{t+d_1}]'$. The first MLP classifier,

[1] All vectors are column vectors by default. Transpose is denoted by $'$

denoted by Θ_{mlp1}, estimates the posterior probability of each of the K phonetic classes $q_t = k, k = 1, 2, \ldots K$, conditioned on the acoustic features spanning $d_1 \approx 4$ frames around $\mathbf{f}_t$ as

$$x_k(t) = P\left(q_t = k \mid \mathbf{f}_{t-d_1:t+d_1},\ \Theta_{mlp1}\right),\ k = 1, \ldots K \quad (1)$$

The estimated posterior probabilities at time t are represented in a vectorial form as $\mathbf{x}_t = [x_1(t), x_2(t), \ldots x_k(t), \ldots x_K(t)]'$, and a temporal context of $2d_2 + 1$ frames on the posterior feature vector is denoted by $\mathbf{x}_{t-d_2:t+d_2}$. The second MLP, denoted by Θ_{mlp2}, estimates the posterior probabilities of phonemes conditioned on a temporal context $d_2 \approx 11$ on the posterior features estimated by the first MLP as

$$z_k(t) = P\left(q_t = k \mid \mathbf{x}_{t-d_2:t+d_2},\ \Theta_{mlp2}\right),\ k = 1, \ldots K \quad (2)$$

The output of the second MLP at time t is represented as $\mathbf{z}_t = [z_1(t), z_2(t), \ldots z_k(t), \ldots z_K(t)]'$. In later parts of this section, $\mathbf{f}_{1:T}$ and $\mathbf{x}_{1:T}$ denotes the sequence of acoustic and posterior feature vectors in the entire utterance, where T denotes the total number of frames.

In practice, the input features to the MLP are normalized to zero mean and unit variance. Feature normalization ensures that the operating region on the hidden activation function is in the linear region, leading to a faster convergence of the back propagation training algorithm [21]. In the case of the second MLP, as the features are posterior probabilities, mean and variance normalization is equivalent to taking scaled likelihoods as features (refer Appendix A for the proof). Hence, normalization of posterior features removes the effect of unigram phonetic class priors learned by the first MLP classifier. The priors are, however, again learned by the second MLP classifier.

C. Background

In this section, we review different approaches in MLP based acoustic modeling, that use hierarchical architectures to model the temporal information in the speech signal, and contrast them with the approach investigated in this paper. In all the discussed works, the first stage of the hierarchy is an MLP. The second stage of the hierarchy includes classifiers such as MLP, HMM, recurrent neural network (RNN), or CRF. The reviewed works are categorized into the following groups (*G1* to *G8*), mainly based on the application of temporal context on the posterior features and the type of classifier at the second stage of the hierarchy.

G1: Classifier Combination

Hierarchical architecture of MLPs have been previously studied in the TRAPS [22] and HATS [23] systems. At the first stage of the hierarchical system, separate MLP classifiers are trained for each of the critical bands. Temporal information in the acoustic features is exploited by using the log critical band energies spanning over a period of about second as input feature. At the second stage, an MLP is used to merge the outputs from the classifiers at the first stage of the hierarchy. In other words, the input to the second MLP classifier are the activations at the output (hidden in case of HATS) layer of the critical band specific MLPs, but without any temporal context. Independent processing of speech in subbands was originally inspired by Allen's interpretation [24] of Fletcher's work [25], indicating a similar mechanism in the human auditory system. Similar hierarchical architectures have also been studied in multiband ASR [26][27].

G2: Feature Combination

Multi-resolution relative spectra [28] features are obtained by filtering the log critical band energies using a bank of multi-resolution bandpass filters. These features are typically used in Tandem based ASR systems. In more recent studies [29][30], the multi-resolution filter bank is split into two groups - fast modulation filters (narrow bandwidth) and slow modulation filters (wider bandwidth) - and combined in a hierarchical fashion. At the first stage of the hierarchy, an MLP is trained with features obtained using fast modulation filters. The estimates of posterior probabilities from the first MLP (log + KLT), with a temporal context of 90 ms are appended to the features obtained using slow modulation filters, and used to train the second MLP classifier. ASR studies using this hierarchical system have shown to yield higher recognition accuracies. In this approach, the second MLP acts like a feature combiner.

G3: Hierarchy using HMM

Hierarchical structures have also been investigated in an attempt to integrate additional knowledge such as minimum duration of phonemes and transition probabilities between phonemes [31]. This knowledge is incorporated into an HMM model Θ_{hmm}. The posterior probabilities of phonemes estimated by the MLP model Θ_{mlp1} are used as emission scores in the HMM states. The new estimates of posterior probabilities are derived from the state occupancy probabilities $P(q_t = k|\mathbf{f}_{1:T}, \Theta_{mlp1}, \Theta_{hmm})$ estimated using the forward-backward algorithm. The new estimates of the posterior probabilities are conditioned on the entire acoustic observation sequence $\mathbf{f}_{1:T}$.

G4: Hierarchy using RNN

Recurrent neural networks (RNN) can also estimate the phonetic class conditional probabilities [32]. In a prior work [33], the hierarchical estimation of the phoneme posterior probabilities using an RNN was investigated. The first stage of the hierarchical system consists of an MLP trained using the power spectrum of the speech. Its output units represent the articulatory features corresponding to the phonemes. In the second stage, an RNN model Θ_{rnn} is trained on the articulatory features estimated by the MLP. In this case, at time t, the RNN estimates the posterior probabilities of the phonemes $P(q_t = k|\mathbf{x}_{1:t}, \Theta_{rnn})$, conditioned on the present and all the previously observed articulatory feature vectors $\mathbf{x}_{1:t}$.

G5: Hierarchy using CRF

There is a growing interest in CRF based models, especially linear chains (with first order Markovian assumption) for reasons such as discriminative training, relaxed conditional independence assumption, and ability to jointly model features

streams with different distributions [34]. In more recent works, CRFs have been investigated for hierarchical estimation of phoneme posterior probabilities [18][19]. At the first stage of the hierarchical system, an MLP estimates the posterior probabilities of phonemes using (1). In the second stage, the estimates of the posterior probabilities from the MLP $\mathbf{x}_{1:T}$ are used as features to the CRF model Θ_{crf}. The new estimates of the posterior probabilities of phonemes $P(q_t = k|\mathbf{x}_{1:T}, \Theta_{crf})$ are obtained using a framework similar to HMM based forward-backward algorithm.

The main difference between the CRF based hierarchical system and HMM based hierarchical system, discussed in *G3*, is in the way the estimates of posterior probabilities from the MLP are used. In the HMM based system, the posterior probabilities of phonemes are used as local acoustic scores in the HMM states, whereas in the CRF based system, they are used as features. In addition, the CRF based system also benefits from discriminative training.

G6: Hierarchy using MLP

In the proposed approach, the MLP at the second stage of the hierarchy yields a new estimate of posterior probabilities, conditioned on a window of the posterior features estimated by the first MLP, and the model Θ_{mlp2} representing the second MLP as $P(q_t = k|\mathbf{x}_{t-d_2:t+d_2}, \Theta_{mlp2})$.

This approach is similar in principle to the RNN based hierarchical approach *G4* and the CRF based hierarchical approach *G5*. The classifiers in the second stage of these systems are trained discriminatively using either posterior features or articulatory features. Apart from the modeling abilities of these classifiers, the main difference between these hierarchical systems is the temporal context on the posterior features. In the RNN based system, the new estimates of posterior probabilities are conditioned on all previously observed posterior feature vectors. In the CRF based approach, it is conditioned on the entire sequence of posterior features. Whereas in our approach, the temporal context on the posterior features is explicitly limited to be around 150-230 ms.

The works described in *G1-G3* are primarily motivated towards exploiting the temporal information in the acoustic features. Whereas in our work as well as *G4* and *G5*, the hierarchical system is motivated towards exploiting temporal information in the posterior features. In this work, the first MLP is trained using standard PLP features. However, it can be trained with any acoustic features, or the first stage can be entirely replaced with more sophisticated MLP based systems described in *G1-G2*. Table I gives a summary of the discussed approaches highlighting the differences in the temporal context and the nature of the second classifier in the hierarchy.

The proposed hierarchical framework can also be related to the following prior works in the literature

G7: Bottleneck Features

In bottleneck feature extraction [35], a five layer MLP with a bottleneck constriction at the middle (or compression) layer, is trained to classify phonemes. The linear activation values at the bottleneck layer are used as features in Tandem based speech recognition. The processing from the input to the compression layer can be likened to the first MLP in the hierarchical system, and the processing from the compression layer to the output layer can be likened to the second MLP.

Even though the architectures of both these systems seem to be similar, the motivation for these works and their application in speech recognition are different. In the bottleneck feature extraction, the objective is to obtain lower dimensional features (independent of the phonetic classes), which are more suitable to the ensuing HMM/GMM system. In the proposed hierarchical system, the first MLP transforms the acoustic features to posterior features with lesser undesirable variabilities such as speaker and environment characteristics. Consequently, the second MLP can exploit the temporal information in the posterior features spanning temporal contexts as long as 250 ms. The second MLP gives new estimates of phonetic class conditional probabilities.

TABLE I
SUMMARY OF THE HIERARCHICAL SYSTEMS EXPLOITING TEMPORAL INFORMATION. NOTATIONS INCLUDE: CLASSIFIER-1 (C1), CLASSIFIER-2 (C2), ACOUSTIC FEATURES (A), POSTERIOR FEATURES (P), POSTERIOR FEATURES TRANSFORMED USING log AND KLT (P_{tr}), LENGTH OF THE UTTERANCE (T).

system name	temporal context		C2 features	C2 type
	C1 (acoustic)	C2 (posterior)		
G1 [22][23]	long (1s)	nil	P	MLP
G2 [29][30]	long (1s)	90 ms	A+P_{tr}	MLP
G3 [31]	T	nil	-	HMM
G4 [33]	any	1:t	P	RNN
G5 [18][19]	any	T	P	CRF
G6 [12][20]	any	230 ms	P	MLP

G8:Frame-based MPE

The hierarchical system discussed in this work can be related to the frame based minimum phone error (fMPE) system [36]. In fMPE, a very high dimensional vector of posterior probabilities is obtained from Gaussian mixture models with a temporal context. The high dimensional posterior vector is projected to a lower dimensional feature space, and used as a correction to the input features such as PLP cepstral coefficients. The linear transformation matrix is trained using minimum phone error criterion [37].

In the MLP based hierarchical system, the high dimensional vector of posterior probabilities is obtained by stacking the output of the first MLP over a long temporal context. The second MLP acts as a nonlinear transform, and is trained using a minimum cross-entropy error criterion, which also achieves minimum phone error. Apart from the nonlinear transformation, the major difference between the two is that in fMPE, the transformed posterior vectors are used as a correction to the input features, but in the hierarchical system, they are used as new features to the ASR. Interestingly, fMPE has been shown to be a special case of semi-parametric trajectory model that models the trajectories of the acoustic features [38]. In our case, the second MLP learns the trajectories of the posterior features. This is discussed in Section IV-C.

III. Experiments and Results

A. Experimental Setup

The efficacy of the hierarchical system in estimating phoneme posterior probabilities is evaluated by performing speaker independent phoneme recognition experiments on TIMIT as well as CTS databases. We preferred phoneme recognition as it facilitates a detailed analysis of the results. Improvements in word recognition using the hierarchical approach have been reported in [20][39].

The TIMIT database consists of 4.3 hours (including 1.1 hours of NIST complete test set) of read speech, recorded in clean conditions. The 'sa' dialect sentences in the database are not included in the experiments. The database is hand-labeled using 61 phonetic symbols, which include the closures as well as the allophonic variations of certain phonemes. In our experiments, these phonetic symbols are mapped to the standard set of 39 phonemes [40] with an additional garbage class.[2]

The CTS setup used in the experiments consists of 277.7 hours speech defined as *ctstrain04*, which is a subset of *h5train03* data set defined at the Cambridge University for training the CU-HTK system for RT03 evaluation [41][42]. [3] The phonetic transcription of the speech - required for training the MLP as well as computing the accuracy of phoneme recognition - is obtained by Viterbi forced alignment. For this, we used off-the-shelf HMM/GMM acoustic models developed in [44] in conjunction with the UNISYN [45] pronunciation dictionary containing 45 phonemes. The dictionary, on an average, contains 1.015 pronunciations per word.

In all the experiments, the acoustic features are the first 13 PLP cepstral coefficients. These coefficients, after speaker specific mean and variance normalization, are appended to their delta and delta-delta derivatives, to obtain a 39 dimensional feature vector for every 10 ms. A three layered MLP with sigmoid nonlinearity at the hidden layer, and softmax nonlinearity at the output is used in all the experiments. The parameters of the MLP are optimized using minimum cross-entropy training criterion. Phoneme recognition is performed using hybrid HMM/MLP approach [6]. The sequence of phonemes is decoded by applying Viterbi algorithm, where each phoneme is represented by a strictly left-to-right, three-state HMM, thereby enforcing a minimum duration of 30 ms. The emission likelihood in each of the three states is the same, and is derived from the associated output of the MLP.

Table II shows the number of speakers and the amount of data in the training, cross-validation, and test sets of the two databases. On TIMIT, the train and test sets are according to the standard protocol. On CTS, the total data is split into train, CV, and test sets as shown in the table. The parameters of the MLP and the phoneme n-gram models are estimated on the training set. The cross-validation set is used to control the learning rate of the MLP. In addition, it is also used to optimize the the phoneme insertion penalty (and language model scaling factor, if phoneme n-gram models are used) of the decoder. All the results reported in this paper are on the test set, which is not seen in the entire training phase.

TABLE II
The number of speakers and the amount of data in the train, cross-validation (CV) and test sets of TIMIT and CTS.

	TIMIT			CTS		
	train	CV	test	train	CV	test
speech (hours)	2.6	0.6	1.1	232.0	36.3	9.4
speakers	375	87	168	4538	726	182

On CTS task, training an MLP with 232 hours of speech is computationally expensive.[4] In order to speed up the experiments to obtain various plots, the training data set is split randomly into two equal parts. The first MLP is trained with one half of the training data, and the second MLP is trained with the remaining half. The single MLP based system is, however, trained on the complete training data. On TIMIT, as the amount of training data is small, both the MLPs in the hierarchical system are trained on the full data.

The MLPs are trained using the Quicknet package [46]. The phoneme n-gram models are trained using the SRILM toolkit [47] and phoneme recognition is performed using the weighted finite state transducer based Juicer decoder [48].

B. Experimental Results

Table III shows the phoneme recognition accuracies obtained by hierarchical modeling (system S2) in comparison with the standard single MLP modeling (system S1). The single MLP system is trained using PLP features with a 90 ms context. The second MLP in the hierarchical system is trained using the output of the single MLP based system S1, with a temporal context of 230 ms. It can be seen that, by hierarchical modeling we obtain an absolute improvement of 3.5% in recognition accuracy on TIMIT, and 9.3% on CTS. To study the effect of increase in the model capacity on the recognition accuracies, we also compare these results to those obtained by a single MLP based system with the same number of parameters as in the hierarchical system (system S3). In this case, the improvement in the recognition accuracies is 2.5% and 8.3% respectively.

In Fig. 2, we compare the phoneme recognition accuracies obtained using hierarchical approach to those obtained using the single MLP approach for different values of the temporal context. In the case of hierarchical system, the first MLP is always trained with a temporal context of 90 ms on the acoustic features. As the temporal context on the posterior features at the second MLP is increased, the total number of parameters in the MLP is kept constant by appropriately

[2]Unlike in [40], the closures are merged with their corresponding bursts (*e.g.,* /bcl/,/b/→/b/). The garbage class handles frames with no labels, and the glottal stop /q/ and its closure /qcl/. The garbage and silence classes are excluded while evaluating the recognition accuracies.

[3]The *h5train03* setup consists of around 296 hours of speech from Switchboard-I [43], Switchboard Cellular, and Callhome English speech corpora, distributed by the Linguistic Data Consortium. For training the AMI RT05 system [44], the sentences containing words which do not occur in the dictionary were removed, resulting in 277.7 hours of *ctstrain04* data set.

[4]Using multi-threaded version of Quicknet [46] (with eight threads and bunch size of 2048), training an MLP of size $351 \times 5000 \times 45$ on 232 hours of speech takes roughly 72 hours to complete 8 epochs on a 2.4 GHz, AMD Opteron processor, with eight cores.

TABLE III
PHONEME RECOGNITION ACCURACIES OBTAINED BY USING HIERARCHICAL POSTERIOR ESTIMATION AS COMPARED TO THE STANDARD SINGLE MLP ON TIMIT AND CTS DATABASES.

	single MLP baseline (**S1**)	hierarchical two MLPs (**S2**)	single MLP same capacity (**S3**)
TIMIT	68.1	71.6	69.1
CTS	54.3	63.6	55.3

reducing the size of its hidden layer.[5] In the case of single MLP estimator, as the temporal context on the acoustic features is increased, the total number of parameters is kept constant, and equal to those in the hierarchical system (sum of the parameters in both the MLPs).

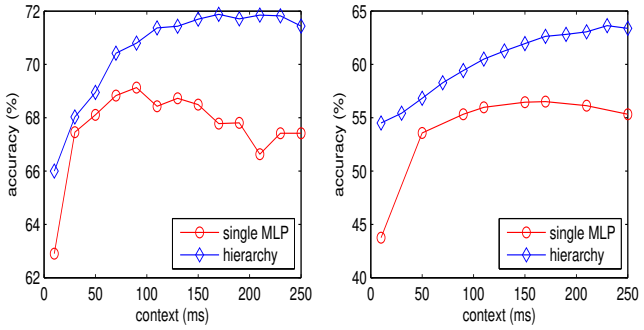


Fig. 2. (a) Phoneme recognition accuracy on TIMIT using an hierarchical setup as well as single MLP with the same number of parameters. In hierarchical system, the size of the first MLP is $351 \times 1000 \times 40$, and the size of the second MLP for 23 frame context is $920 \times 1083 \times 40$. (b) A similar plot on the CTS, where the size of the first MLP is $351 \times 5000 \times 45$, and the size of the second MLP for 23 frame context is $1035 \times 1334 \times 45$. Any two points in the plot correspond to systems with the same number of parameters, and can be calculated using.[5]

It can be seen from the figure that:

1) The hierarchical system consistently outperforms the single MLP based system with the same number of parameters for all values of context. As the context at the second MLP is increased, even though the number of hidden nodes is decreased, there is a steady increase in the recognition accuracies. Thus it can be concluded that improvement is due to the topology of two MLPs in tandem, and not merely due to the increase in overall model capacity.
2) In case of CTS, the recognition accuracies begin to saturate at around 230 ms of temporal context at the input of the second MLP. In case of TIMIT, the accuracies begins to saturate after 150 ms, but this could be due to the lack of sufficient training data. In both cases, the effective temporal context of 150-230 ms extends well beyond the typical duration of phonemes (50-70 ms), which suggests that the second MLP is integrating temporal information in the posteriors features corresponding to the neighboring phonemes as well.
3) A long temporal context is more effective when applied on the posterior features rather than on the acoustic features. On increasing the temporal context on the acoustic features at the input of the single MLP system, recognition accuracies peak for a context of around 90-110 ms, but are significantly lower when compared to the hierarchical system.

From the above discussion it is clear that the hierarchical system is useful as a phoneme posterior probability estimator, and that a long temporal context is more effective on the posterior features rather than on the acoustic features. As the second MLP is trained using posterior features, which represents the underlying sequence of phonemes, it is clear that the second MLP learns the phonetic-temporal patterns.

The following questions, however, remain unanswered: (a) what are the phonetic-temporal patterns learned for each phoneme ? (b) as the long temporal context extends beyond the typical duration of phonemes, has the second MLP also learned the phonotactics of the language ? and (c) why is the relatively longer temporal context more effective on the posterior features ?

The first two questions can be answered by analyzing the input-output relationship learned by the second MLP classifier. In this work, we use Volterra series for the analysis, and this is discussed in Section IV. The effectiveness of temporal context on the posterior features is discussed in Section V.

C. Second MLP as a Function

The second MLP can be viewed as a vector valued function $f_{mlp2}(.)$, which takes the estimates of posterior probabilities of phonemes from the first MLP denoted by $\mathbf{x}_{t-d_2:t+d_2}$ as its arguments, and gives a new estimate of the posterior probabilities of phonemes $\mathbf{z}_t$ as

$$\mathbf{z}_t = f_{mlp2}(\mathbf{x}_{t-d_2:t+d_2}). \tag{3}$$

In the second MLP classifier, let W denote the weight matrix connecting the input layer to the hidden layer, C denote the weight matrix connecting the hidden layer to the output, $\mathbf{b}_h$ and $\mathbf{b}_o$ denote the bias vectors at the hidden and output layers respectively, and $f_{soft}(.)$ and $f_{sigm}(.)$ denote the vector valued softmax and sigmoid functions at the output and the hidden layers of the MLP respectively. Then, equation (3) can be expressed as

$$\mathbf{z}_t = f_{soft}(\mathbf{y}_t), \tag{4}$$

where the vector $\mathbf{y}_t = [y^1(t), \ldots y^j(t), \ldots y^N(t)]'$ denotes the linear activation vector before the softmax nonlinearity at the output layer of the MLP, and is given by

$$\mathbf{y}_t = \mathbf{b}_o + C f_{sigm}(\mathbf{b}_h + W\mathbf{x}_{t-d_2:t+d_2}). \tag{5}$$

It is difficult to analyze or interpret the input-output relationship $(\mathbf{x}_t, \mathbf{z}_t)$ of the MLP, given by (4) and (5), due to the presence of nonlinear functions $f_{sigm}(.)$ and $f_{soft}(.)$. The output nonlinearity can be conveniently dropped from the analysis as parameters of the discriminatively trained MLP $\{W, \mathbf{b}_h, C, \mathbf{b}_o\}$ can still be interpreted from the input-output relationship $(\mathbf{x}_t, \mathbf{y}_t)$. This does not affect the interpretability

[5] If F denotes the dimensionality of the features, C denotes the temporal context, and H (and O) denote the size of the hidden (and output) layers, the number of parameters in the MLP is given by $C*F*H+H+H*O+O$.

as the output units are still phonemes, and the ordering of the estimates are not altered. The nonlinearity at the hidden layer, however, can still make the analysis of (5) difficult.

In our previous work [12], this problem was circumvented, but not solved, by using a single layer perceptron (SLP) in place of the second MLP in the hierarchical system. The SLP retained the same input-output architecture, training data, and optimization criterion as that of the MLP. The weights of the trained perceptron revealed the linear fit to the observed training data. However, the MLP classifier which was actually used in ASR studies was not analyzed.

In this work, we follow a more principled approach and represent the second stage of the hierarchical system using Volterra series. For this, we treat the multi-input $\mathbf{x}_t$, multi-output $\mathbf{y}_t$ system characterized by (5) as a nonlinear time-invariant system. Traditionally, in the literature, such systems have been analyzed using Volterra series [14][15]. By using Volterra series, the nonlinear system can be decomposed into its linear, quadratic, and higher order parts and analyzed.

At this stage, we digress from the discussion on hierarchical systems to present the theory of Volterra series. We also briefly discuss our earlier work on representing a cascade of finite impulse response (FIR) filter bank and an MLP using Volterra series [49][50]. The analysis of the hierarchical system using Volterra series is resumed from Section IV-C onwards.

IV. Volterra Series

A Volterra series is an infinite series which can model the input-output relationship of a nonlinear time-invariant system. As an illustration, we first discuss the Volterra series expansion for a single-input, single-output system.

A. Volterra Series: Single Input - Single Output System

If $x(t)$ is the input to a nonlinear system, and $y(t)$ its output, the Volterra series expansion for the system is given by

$$y(t) = \sum_{n=0}^{\infty} G_n\left[g_n, x(t)\right]$$

where, $\{G_n\}$ is the set of Volterra functionals, and $\{g_n\}$ is the set of Volterra kernels of the nonlinear system. The first three functionals in the Volterra series are given by

$$G_0\left[g_0, x(t)\right] = g_0,$$

$$G_1\left[g_1, x(t)\right] = \int_{\mathbb{R}} g_1(\tau)x(t-\tau)d\tau, \quad \text{and}$$

$$G_2\left[g_2, x(t)\right] = \int_{\mathbb{R}^2} g_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)d\tau_1 d\tau_2$$

Each term in the Volterra series is a multi-dimensional convolution between the input to the system and its Volterra kernels. The Volterra kernels $\{g_0, g_1, g_2 \ldots g_\infty\}$ completely characterize the nonlinear time-invariant system.

The first order Volterra functional G_1 is the linear convolutional integral, and its kernel g_1 is impulse response function, which characterizes the linear part of the nonlinear system. As a special case, if the system is linear, then the Volterra series reduces to order one, and its first order Volterra kernel gives the actual impulse response function of the system.

Volterra series has been extensively used in the analysis of biological systems [51]. It has also been used in the literature to analyze artificial neural networks in various fields of engineering. For example, in the analysis of neural networks used for velocity estimation in computer vision [52], analysis of perceptron based nonlinear noise filtering and beamforming [53], analysis of time-delay neural networks used to model the nonlinear behavior of electronic devices [54], etc.

B. Volterra series : Three Layered MLP

In recent works [49][50], we proposed a mathematical framework to apply Volterra series to a nonlinear time-invariant system comprising of an FIR filter bank, followed by a three layer MLP. This generic framework was developed to analyze MLP classifiers trained using standard acoustic features such as mel frequency cepstral coefficients (MFCC), along with the dynamic coefficients. In such cases, if the MLP is analyzed as a standalone system, then the functionality of the trained MLP is revealed in terms of input features (*e.g.,* cepstral patterns), which is difficult to analyze. However, in most cases, ASR features are obtained by processing an intermediate representation (*e.g.,* spectro-temporal) using a linear time-invariant system. For instance, in MFCC, the intermediate representation is the $\log$ energies in the mel critical bands, and the linear system consists of discrete cosine transformation matrix and the FIR filters that compute the dynamic cepstral features. By including the linear system in the analysis, the parameters of the trained MLP can be analyzed using more interpretable spectro-temporal patterns.

Application of Volterra series to the second stage of the hierarchical system forms a special case in this generic framework. The input to the system are the posterior features estimated by the first MLP. The temporal context on the posterior features can be viewed as being obtained using a bank of FIR filters with time-shifted Kronecker delta impulse response functions.

Fig. 3 is a block diagram of the system under analysis. Let $\mathbf{x}_t = [x_1(t), \ldots x_k(t), \ldots x_K(t)]'$ denote the input to the FIR filter bank, where K is the number of inputs. If L denotes the number of filters in the filter bank, and $h_l(t)$ denotes the impulse response of these filters, then the input features to the MLP is given by $\mathbf{u}_t = [u_{1,1}(t), \ldots u_{k,l}(t), \ldots u_{K,L}(t)]'$, where $u_{k,l}(t)$ is given by the convolution [6] between $x_k(t)$ and $h_l(t)$ as

$$u_{k,l}(t) = \int_{\tau} h_l(\tau)x_k(t-\tau)d\tau \tag{6}$$

Furthermore, let M and N denote the size of the hidden and output layers respectively, $w^i_{k,l}$ denote the weight connecting the node (k,l) in the input layer to the node i (with a bias b^i_h) in the hidden layer, c^j_i denote the weight connecting the hidden node i to the output node j (with a bias b^j_o), and $\phi(.)$ denote the nonlinear activation function at the hidden layer. The output of the system $\mathbf{y}_t = \left[y^1(t), \ldots y^j(t), \ldots y^N(t)\right]'$ is

[6]Even though the above system is a discrete-time system, continuous-time notations are used for clarity. This helps in distinguishing the integral operator in the convolution from the summation in the MLP function.

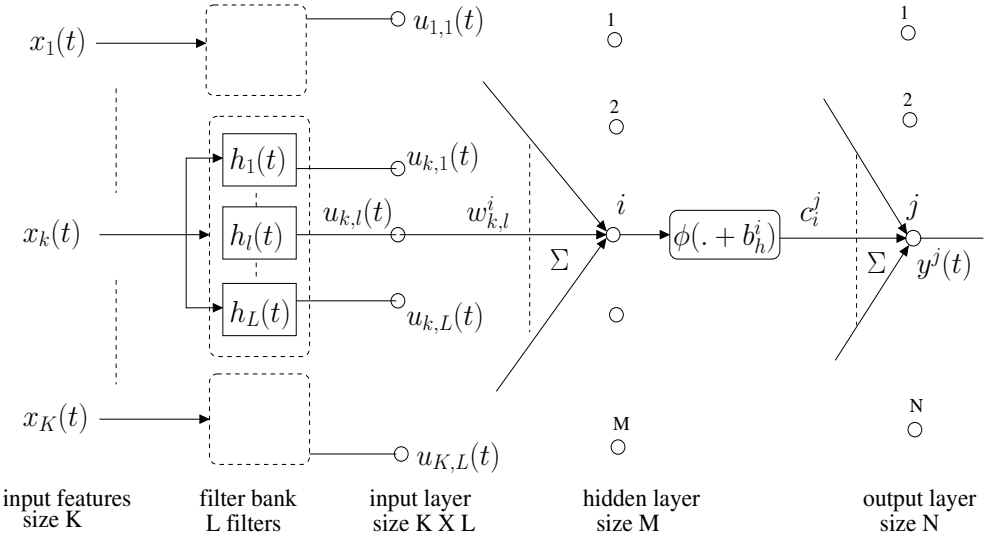


Fig. 3. Block schematic of the system analyzed using a Volterra series. It consists of an FIR filter bank followed by a three layer MLP.

the linear activation values before the output nonlinearity in the MLP, and is given by

$$y^j(t) = b^j_o + \sum_{i=1}^{M} c^j_i \phi\left(b^i_h + \sum_{k=1}^{K}\sum_{l=1}^{L} w^i_{k,l} u_{k,l}(t)\right). \quad (7)$$

The nonlinear time-invariant system characterized by (6) and (7) cannot be analyzed in its present parametric form due to the nonlinear function $\phi(.)$. However, if the nonlinear function can be expressed as a power series, then the same system can be alternatively represented using Volterra series as

$$y^j(t) = g^j_0 + \sum_{k_1=1}^{K}\int_{\tau_1} g^j_{k_1}(\tau_1)\, x_{k_1}(t-\tau_1)\, d\tau_1 + \sum_{k_1=1}^{K}\sum_{k_2=1}^{K}$$

$$\int_{\tau_1}\int_{\tau_2} g^j_{k_1 k_2}(\tau_1,\tau_2)\, x_{k_1}(t-\tau_1)\, x_{k_2}(t-\tau_2)\, d\tau_1 d\tau_2 + \ldots \quad (8)$$

and analyzed. In the above equation, g^j_0, $g^j_{k_1}(\tau_1)$, and $g^j_{k_1 k_2}(\tau_1,\tau_2)$ respectively denote the zeroth, first, and second order Volterra kernels of the trained MLP for the output class j. The variables $\tau_1, \tau_2 \ldots$ denote time, and $k_1, k_2 \ldots$ denote the components of the input vector $\mathbf{x}_t$. The first order Volterra kernel is the linear part of the nonlinear system, and can reveal the contribution of the input $x_{k_1}(t)$ to the output $y^j(t)$. Similarly, the second order Volterra kernels reveal the quadratic part of the nonlinear system.

The Volterra kernels of the system can be identified in terms of the impulse response of the FIR filter bank and the parameters of the trained MLP. Suppose that the nonlinear activation function $\phi(.)$ at the hidden node i with a bias b^i_h can be approximated as a polynomial expansion as

$$\phi(. + b^i_h) = a_{0,i} + a_{1,i}(.) + a_{2,i}(.)^2 + \ldots, \quad (9)$$

where, $a_{0,i}, a_{1,i}, a_{2,i} \ldots$ are the coefficients of the polynomial expansion. By substituting (6) and (9) in (7), and comparing the resulting equation to (8), the first three Volterra kernels are identified as

$$g^j_0 = b^j_o + \sum_{i=1}^{M} c^j_i\, a_{0,i} \quad (10)$$

$$g^j_{k_1}(\tau_1) = \sum_{i=1}^{M} c^j_i\, a_{1,i} \sum_{l_1=1}^{L} w^i_{k_1 l_1} h_{l_1}(\tau_1) \quad (11)$$

$$g^j_{k_1 k_2}(\tau_1,\tau_2) = \sum_{i=1}^{M} c^j_i a_{2,i} \sum_{l_1=1}^{L}\sum_{l_2=1}^{L} w^i_{k_1 l_1} w^i_{k_2 l_2} h_{l_1}(\tau_1) h_{l_2}(\tau_2) \quad (12)$$

The complete derivation of the Volterra kernels is described in [49]. Note that the bias at the hidden layer is captured in the polynomial coefficients and the bias at the output layer is incorporated into the zeroth order Volterra kernel. The identified Volterra kernels are in continuous-time notations. The corresponding discrete time kernels are obtained by using discrete-time expressions for the impulse response functions of the filter bank in (10)-(12).

Polynomial expansion: The key step in the analytical identification of the Volterra kernels is the polynomial approximation of the hidden nonlinearity. Polynomial expansion of saturating functions such as sigmoid or hyperbolic tangent are divergent if approximated for all possible values of the input $(-\infty, \infty)$. However, since the MLP is trained using posterior features, which are trained to be linearly separable as discussed in Section V-A3, and as a consequence of feature normalization, the operating point on the nonlinearity is in a relatively small region containing the linear part of the function. To estimate the polynomial coefficients, the operating region on the hidden nonlinearity is first identified using cross-validation data. The coefficients are subsequently optimized to minimize the least square error between the sigmoid function and its polynomial approximation in the operating region of the hidden nonlinearity, leaving a small percentage (1%) of its tail. The estimation of polynomial coefficients is described in detail in [49].

C. Application of Volterra Series

In this section, we compute the Volterra kernels for multi-input $\mathbf{x}_t$, multi-output $\mathbf{y}_t = [y^1(t), \ldots y^j(t), \ldots y^N(t)]'$ system characterized by (5). This system can be viewed as N parallel, multi-input, single-output, nonlinear, time-invariant systems, and represented by

$$y_t^j = b_o^j + C^j f_{sigm}\left(\mathbf{b}_h + W\mathbf{x}_{t-d_2:t+d_2}\right), \quad j = 1 \ldots N, \tag{13}$$

where, C^j denotes the weight row vector connecting the hidden layer to the output node j, and b_o^j the bias at the output node j. The system represented by (13) can be realized using the framework shown in Fig. 3, where the temporal context of $2d_2 + 1$ frames on the posterior features, denoted by $\mathbf{x}_{t-d_2:t+d_2}$, can be created by filtering $\mathbf{x}_t$ using a bank of $L = 2d_2 + 1$ FIR filters. The impulse response of the $2d_2 + 1$ tap FIR filter is given by

$$h_l(n) = \delta\left(n + l - \frac{L+1}{2}\right) \quad \begin{matrix} l &=& 1, 2 \ldots L \\ n &=& -d_2, \ldots 0, \ldots d_2 \end{matrix}$$

The Volterra kernels are computed in terms of the above impulse response functions and the weights of the trained MLP using the discrete-time versions of (10)-(12). In practice, due to feature normalization, $\mathbf{x}_t$ represents posterior features which are normalized to zero mean and unit variance.

In the remaining part of this section, we analyze trained second MLPs in the hierarchical system (see Table III for results) - one trained on TIMIT ($K = 40, L = 23, M = 1083, N = 40$), and the other trained on CTS ($K = 45, L = 23, M = 1334, N = 45$). Before analyzing the Volterra kernels, the accuracy of first and second order truncated Volterra series is evaluated. For this, we substitute the identified kernels in the synthesis equation (8) to obtain the linear activation values of phonemes. Approximate estimates of phoneme posterior probabilities are obtained by applying softmax nonlinearity, and subsequently used in phoneme recognition.

TABLE IV
PHONEME RECOGNITION ACCURACY OBTAINED BY LINEAR AND QUADRATIC APPROXIMATION OF THE MLP USING THE VOLTERRA SERIES.

model	series order	phoneme accuracy	
		TIMIT (%)	CTS (%)
linear	1	68.7	50.1
quadratic	2	70.1	54.9
MLP	∞	71.6	63.6

Table IV shows the phoneme recognition accuracies obtained by the first and second order Volterra series approximation of the second MLP classifier. In theory, the recognition accuracy obtained by the Volterra series approximation should approach asymptotically to the accuracy obtained by the direct evaluation of the MLP, as the order of the series is increased. However, the computation of the higher order Volterra kernels is computationally intensive and hence not practical.

It can be seen that on TIMIT, the phoneme recognition accuracy obtained by the first order Volterra approximation is only three percent lower compared to direct evaluation of the MLP function. In other words, the second (quadratic), third (cubic), and higher order parts contribute very little to nonlinear modeling ability of the second MLP. Hence, in this case, the linear Volterra kernels reveal most of the information learned by the nonlinear classifier.

In the case of a more complex CTS task, the phoneme recognition accuracy obtained using first order Volterra series is 13.5% lower compared to the direct evaluation of the MLP. This implies that second and higher order Volterra kernels contribute significantly to the modeling ability of the second MLP and that the linear Volterra kernels can only partially explain its functionality. The remaining information is complemented by the higher order Volterra kernels. In this work, we restrict the analysis to linear Volterra kernels.

D. Interpretation of the First Order Volterra Kernels

It is clear from (8) that the first order Volterra kernels reveal the linear part of the nonlinear system under analysis. Suppose that the second MLP is trained using a temporal context of 230 ms, then the Volterra kernel for phoneme $j = 1, 2 \ldots N$ at the output of the second MLP is given by $g_k^j(t)$, and reveals the contribution of each of the phonemes $k = 1, 2 \ldots K$ at the input of the MLP, in a window of $[t-11, \ldots t, \ldots t+11]$, which amounts to 230 ms of context. As the input to the second MLP is in terms of phonemes, the first order Volterra kernels can be interpreted as phonetic-temporal patterns. In our experiments, $N = K$ as both the MLPs in the hierarchical system are trained on the same phoneme set.

The phonetic-temporal patterns observed in the first order Volterra kernels can reveal two important aspects learned by the second MLP classifier: 1) the acoustic confusion among phonemes at the output of the first MLP classifier, and 2) the phonotactics of the language as observed in the training data. In the remaining part of this section, we discuss these aspects in detail.

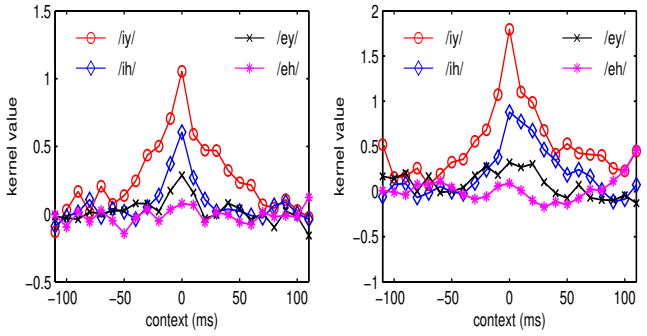


Fig. 4. (a) First order Volterra kernel of the phoneme /iy/ (*e.g.,* b**ea**t) obtained on TIMIT. (b) A similar plot on CTS database.

1) Volterra kernels revealing acoustic confusions among phonemes: Fig. 4 (a) and (b) are the plots of the first order Volterra kernel of the second MLP classifier for the vowel /iy/ (*e.g.,* b**ea**t) on TIMIT and CTS respectively. The figure shows the impulse response functions corresponding to the top four contributing phonemes at the input of the MLP. The impulse

response function corresponding to other phonemes are not plotted in the figure for clarity. The top contributing phonemes are selected based on the energy in their impulse response functions. It is not surprising that the maximum contribution is from the same phoneme /iy/ at the input. There are, however, positive contributions from other confusing vowels such as /ih/, /ey/, and /eh/.

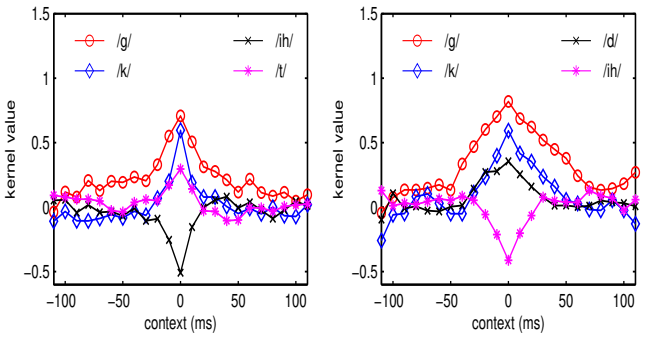


Fig. 5. First order Volterra kernel of the phoneme /g/ (*e.g.,* **g**oat) obtained on TIMIT. (b) A similar plot on CTS database.

Fig. 5 (a) and (b) are plots of first order Volterra kernel of the phoneme /g/ (*e.g.,* **g**oat) obtained on TIMIT and CTS databases respectively. It can be seen that the kernels show positive contributions from other confusing consonants such as /k/, /t/, and /d/. Moreover, the MLP has also learned to give negative weight to the vowel /ih/. This is due to the discriminative training of the MLP and this information is otherwise not intuitive. It suggests that the consonant /g/ is less likely to be confused with the vowel /ih/.

As both the input and output representations of the second MLP are in terms of phonemes, the first order Volterra kernel can be interpreted as a phonetic-temporal confusion patterns. However, unlike the standard phonetic confusion matrix, the first order Volterra kernels reveal the contribution of the input phonemes in a window of certain duration depending on the temporal context used. In Table V, we show the top three contributing phonemes at the center ($t = 0$) of the Volterra kernels for both TIMIT as well as CTS databases. These confusions patterns are compared to the standard confusion matrix, obtained by performing frame-level phoneme classification at the output of the first MLP. Only entries in the confusion matrix with values greater than 0.06 are shown in the table.

It can be seen from the table that the confusions at the center of the Volterra kernels match to a certain extent with standard phonetic confusion matrix derived from the posterior features. However, these confusion entries need not be the same because the Volterra kernels represent the discriminatively trained second MLP classifier, whereas the phonetic confusion matrix is a measure of the phonetic confusion in the posterior features, which are used to train the second MLP.

It is interesting to note that the ability of the second classifier (an MLP in our case) in the hierarchical setup to learn the acoustic confusion among phonemes at the output of the first MLP has also been observed in the CRF based hierarchical system [19] (refer system G5 in section II-C).

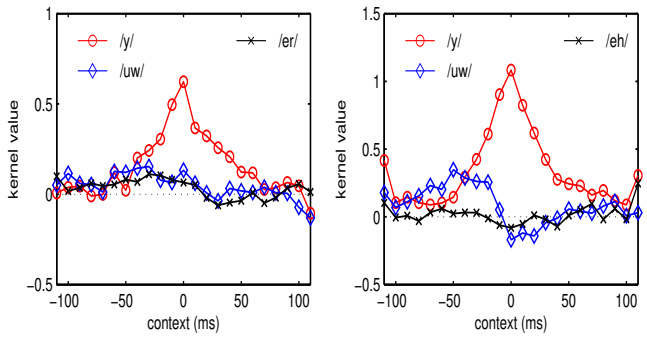
TABLE V
CONFUSING PHONEMES AT THE CENTER OF THE VOLTERRA KERNELS (TOP THREE) AS COMPARED TO THE PHONETIC CONFUSION MATRIX (VALUE > 0.06).

phonemes TIMIT	confusions Volterra	confusion matrix	phonemes CTS	confusions Volterra	confusion matrix
iy	ih, ey, eh	ih	iy	ih, eh, ey	ih, ey
ih	iy, eh, ae	ah	ih	iy, sil, eh	ax, iy
ey	ih, iy, ae	ih, iy	ey	ih, ay, eh	iy, ih
eh	ih, ae, ah	ih, ae, ah	eh	ah, ih, ey	ae, ih, ax, ah
			aa	ah, ay, ow	ah, ay, ao
ah	ih, ao, eh	ih, ao, ow	ah	ay, eh, l	ax, ow
			ax	axr, ah, m	ih, ah
			axr	r, ax, ih	r, ax
uw	ih, iy, w	ih, iy	uw	iy, ih, ow	iy, ax
uh	ih, ah, eh	ih, ah, ow, l, uw	uh	ih, s, ey	ax, ih
ae	ao, ah, aw	eh	ae	eh, ah, ay	eh
ao	ae, ay, ah		ao	aa, l, w	aa, ow
aw	ao, ah, ae	ao, ae	aw	ah, ay, eh	ae, ow, ah, aa, eh, ay
ay	ao, ah, ey	ao	ay	ah, eh, aa	ah
ow	ah, ao, l	l, ah, ao	ow	ah, l, ao	ah, l
oy	ao, ih, ay	ao, ey	oy	r, w, ay	w, l, ao, ow
y	iy, ih, oy	iy, uw, ih	y	iy, ae, ch	iy, sil
w	l, uh, oy	l	w	l, r, ao	
l	ao, ah, ow	ow, ao	l	ah, el, w	ow
			el	l, ow, ao	l, ow, ax
r	er, ae, ao	er	r	axr, iy, w	axr
er	r, ih, ah	r	er	r, axr, ih	r, axr
hh	sil, k, p	sil	hh	s, ae, dh	sil
m	n, p, b	n	m	n, ng, w	n, sil
			em	n, ah, m, en	m, ah, sil, n, ax
n	m, dx, dh	m	n	m, ng, en	d
			en	n, m, ng	n, ax, d, m
ng	n, m, uw	n	ng	n, m, iy	n
p	t, b, k		p	k, t, f	t, sil
t	d, p, k	d, k	t	d, k, m	sil
k	sil, t, p	t	k	sil, p, t	sil, t
b	p, d, m	p	b	p, dh, w	dh
d	t, dx, k	t	d	t, sil, s	t, n, sil
g	k, d, t	k, d	g	k, d, dh	k
dx	d, n, dh	d			
f	p, s, sil		f	s, sil, k	s, sil
th	s, t, f	f, t	th	s, sil, f	s, t, sil
s	z, sh, f	z	s	f, sh, z	sil, z
sh	s, z, jh	s	sh	s, f, ch	s, ch
v	f, b, m		v	sil, f, z	ax
dh	t, th, d	sil	dh	y, b, g	t, d
z	s, sh, th	s	z	s, sil, f	s, sil
			zh	iy, ih, z	z, sh, uw
ch	s, jh, sh	sh, t, jh, s	ch	t, s, k	t, s, sh
jh	s, z, sh	ch, sh	jh	ch, d, y	t, d, ch

2) Volterra kernels revealing the phonotactics of the language: A closer look at the first order Volterra kernels reveals that the MLP has also learned the phonotactics the language. In the ensuing discussions, the following notations are used. $P(p1^{+}|p2) = P(p_{n+1} = p1|p_n = p2)$ denotes the probability that phoneme /p1/ follows /p2/, and is typically used using n-gram statistical language modeling. In contrast, $P(p1^{-}|p2) = P(p_{n-1} = p1|p_n = p2)$ denotes the probability that phoneme /p1/ precedes /p2/. To estimate this language model, the sequence of phonemes in the training data are reversed, and bigram statistics are estimated.

Fig. 6 (a) is a plot of the first order Volterra kernel of the phoneme /y/ on TIMIT, showing the contributions of two phonemes /uw/ and /er/ that are most likely to follow /y/. It can be seen that the corresponding kernels have higher value to the left of the origin as compared to the right. This is because $P(uw^{+}|y) = 0.52 \gg P(uw^{-}|y) = 0.04$. As Volterra kernels are impulse response functions, the corresponding matched filters are obtained by time-reversing the kernels about their origin $t = 0$.

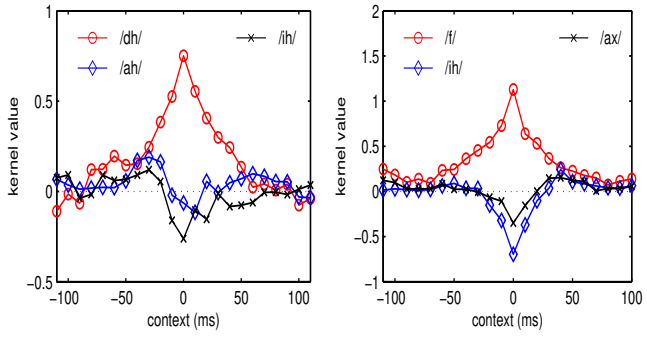
Fig. 6 (b) is a plot of the Volterra kernel of phoneme /y/ on CTS, showing the impulse response functions of phonemes /uw/ and /eh/, that are most likely to follow /y/. It can be



(a) (b)

Fig. 6. (a) The Volterra kernel of phoneme /y/ on TIMIT. $P(uw^+|y) = 0.52$, $P(uw^-|y) = 0.04$, $P(er^+|y) = 0.16$, and $P(er^-|y) = 0.03$. (b) The Volterra kernel of phoneme /y/ on CTS. $P(uw^+|y) = 0.54$, $P(uw^-|y) = 0.04$, $P(eh^+|y) = 0.30$, and $P(eh^-|y) = 0.001$.

seen that the kernel for /uw/ is consistent with the bigram language model probabilities, but in case of /eh/, there is no such agreement as the kernel is close to zero for all values of the context.



(a) (b)

Fig. 7. (a) The Volterra kernel of phoneme /dh/ on TIMIT. $P(ih^+|dh) = 0.34$, $P(ih^-|dh) = 0.04$, $P(ah^+|dh) = 0.29$, and $P(ah^-|dh) = 0.11$ (b) The Volterra kernel of phoneme /f/ on CTS. $P(ih^+|f) = 0.07$, $P(ih^-|f) = 0.17$, $P(ax^+|f) = 0.05$, and $P(ax^-|f) = 0.10$.

Fig. 7 (a) is the plot of the impulse response functions of phonemes /dh/, /ah/, and /ih/ in the first order Volterra kernel of phoneme /dh/ (*e.g.,* **th**is) on TIMIT. It can be seen that the impulse response functions of phonemes /ih/ and /ah/ have higher weight to the left of origin as compared to the right. This is because the pairs of phonemes /dh//ah/ and /dh//ih/ occur more frequently in the training data than the pairs /ah//dh/ and /ih//dh/.

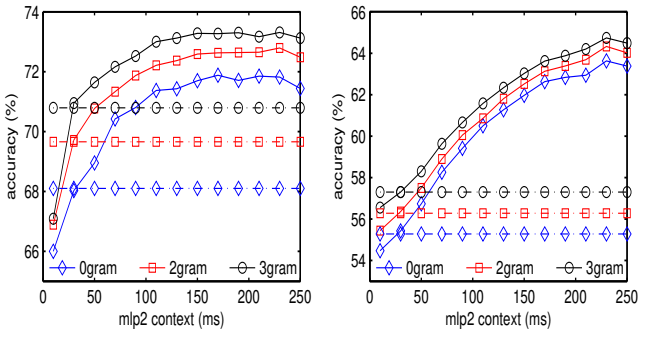
In Fig. 7 (b), we plot the impulse response functions of phonemes /f/, /ih/, and /ax/ in the first order Volterra kernel of phoneme /f/ (*e.g.,* **f**ar) on CTS. Phonemes /ih/, and /ax/ are the two most likely phonemes to precede /f/ and as a consequence, their impulse response functions have higher values to the right of the origin. Moreover, it can also be seen that at the origin, the impulse response functions of /ih/ and /ah/ have negative weights, which suggests that these vowels are not confusable with the consonant /f/. It should be noted that the Volterra kernels reveal the properties of the discriminatively trained MLP. Hence, they need not always be consistent with the bigram probabilities between phonemes (derived from simple counts) in all cases.

The interpretations that can be drawn by analyzing the linear Volterra kernels are summarized below. If $g_1^1(\tau)$ and $g_2^1(\tau)$ are the impulse response functions (indicating the contributions) of phonemes /p1/ and /p2/ respectively in the Volterra kernel of the phoneme /p1/. The function $g_1^1(\tau)$ will always have a positive peak at the origin $\tau = 0$. Depending on the shape of the function $g_2^1(\tau)$, the interpretations could be as follows: (a) a positive peak at the origin indicates the acoustic confusion between the phonemes, (b) a negative valley at the origin indicates the anti-confusion due to the discriminative training of the MLP, and (c) a peak which is shifted away from the origin reveals the phonotactics implicitly learned by the MLP. Moreover, the Volterra kernels can also reveal the effective temporal duration learned by the system.

E. Decoding with Language Models

First order Volterra analysis of the hierarchical system reveals that, apart from the acoustic confusions, the second MLP has also implicitly captured the phonotactics of the language. However, it is not clear if the implicitly learned phonotactics has indeed contributed towards the increase in the recognition accuracies of the hierarchical system. To ascertain this, we performed phoneme recognition by explicitly using phoneme n-gram models.

Fig. 8 (a) and (b) are plots of the phoneme recognition accuracies on TIMIT and CTS respectively, obtained by decoding with nogram (loop of phonemes with equal transition probabilities), bigram and trigram phoneme language models. The accuracies are shown for temporal context at the second MLP ranging from 10ms to 250ms. As the input context is increased, the total number of parameters of the second MLP is kept constant by appropriately modifying the size of the hidden layer. The horizontal dotted lines in the plot indicate the recognition accuracies obtained by a single MLP based system using different language models. It can be seen from the figure that recognition accuracies increase by explicitly using bigram and trigram models. This improvement is observed for all values of the temporal context on the posterior features, but the gain in the accuracies decreases with the increase in context.



(a) (b)

Fig. 8. (a) Phoneme recognition accuracies on TIMIT using nogram, bigram, and trigram phoneme language models. The horizontal lines show the accuracy of the first MLP using language models. (b) A similar plot on CTS database.

To illustrate this, in Fig. 9 we plot the relative gain in the recognition accuracies obtained on CTS by decoding with bigram and trigram language models over no language model, as a function of the temporal context at the input of the second MLP classifier. It can be seen that the gain in accuracy obtained by explicitly using a phoneme n-gram model decreases with the increase in the temporal context. This is because, with increase in the temporal context, the

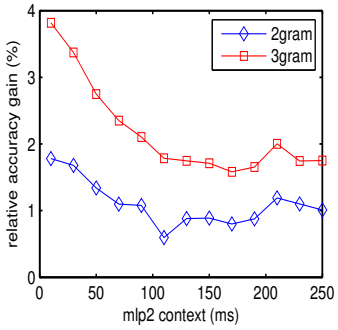


Fig. 9. Relative gain in recognition accuracy on CTS database obtained by decoding with bigram and trigram language model as compared to no language model for different values of the temporal context at the input of the second MLP.

second MLP is able to learn the phonotactics more effectively, and gain in accuracy by introducing explicit language models reduces. This further supports the observations from the linear Volterra kernels. However, even with 230 ms context, the MLP has only partially learned the phonotactics and we still obtain 1-2% improvement in accuracies by using bigram/trigram language models in decoding.

To summarize briefly, we showed in this section that the second MLP classifier in the hierarchical system learns the phonetic-temporal patterns (acoustic confusions among phonemes and and the phonotactics of the language) in the posterior features spanning a temporal context of 150-230 ms. In the following section, we discuss the important properties of the posterior features that enabled the second MLP to effectively learn these patterns.

V. Modeling flexibility of posterior features

In this section, we discuss the important properties of posterior features such as (a) lesser nonlinguistic variabilities when compared to the acoustic features, (b) sparse distribution, and (c) linear separability in the posterior feature space. We also discuss the consequence of these properties on the complexity of the second MLP classifier and the amount of training data.

A. Characteristics of Posterior Features

1) Variability in posterior features: The acoustic features are known to exhibit a high degree of nonlinguistic variabilities such as speaker and environmental (*e.g.,* noise, channel) characteristics. The first MLP classifier can be interpreted as a discriminatively trained nonlinear transformation from the acoustic feature space to the posterior feature space. It has been shown that a well trained (large population of speakers, and different conditions) MLP classifier can achieve invariance to speaker [3] as well as environmental [8] characteristics. Moreover, it has also been shown that the effect of coarticulation is less severe on the posterior features when compared to the acoustic features [55][56].

In other words, the posterior features are soft-decisions on the underlying sequence of phonemes (*i.e.,* the linguistic message), and have much lesser nonlinguistic variabilities when compared to acoustic features.

2) Sparseness in the posterior features: The posterior features represent the probabilities of the phonetic classes conditioned on the acoustic features, and hence sum up to unity at any given time instant. In addition, they are also sparsely distributed in the posterior feature space. To illustrate this, in Table VI, we show the average number of components (or phonemes) in the posterior feature vector that capture 90, 95, and 99% of the probability mass value. It can be seen that on TIMIT, on an average, 3.6 phonemes capture 95% of the probability mass value. The other phonemes share the remaining 5% of the probability mass. On CTS, on an average 6.2 phonemes capture 95% of the probability mass value, indicating the more complex nature of the task.

TABLE VI
AVERAGE NUMBER OF COMPONENTS (PHONEMES) IN THE POSTERIOR FEATURE VECTOR THAT CAPTURE 90, 95, AND 99% OF THE PROBABILITY MASS IN THE POSTERIOR PROBABILITIES OF PHONEMES ESTIMATED BY THE FIRST MLP.

	probability mass value		
	>90%	>95%	>99%
TIMIT (max 40)	2.7	3.6	6.6
CTS (max 45)	4.4	6.2	11.3

The sparse distribution of the posterior features has been previously studied in [3], where the authors termed the posterior features as more *regular* compared to the standard acoustic features. It was argued that sparse distribution was one of the favorable properties of posterior features.

3) Linear separability: The model parameters of the first MLP are optimized to minimize the cross entropy between the estimated posterior probability vectors and the output target vectors, which are typically in the hard-target format. In other words, if K denotes the number of phonemes, the hard target vector $l_{p_i} \in \mathbb{R}^K$ for the phoneme $p_i, \quad i = 1, 2 \ldots K$ is given by $l_{p_i}(k) = \delta(k - i)$. The target vectors are, therefore, at the simplex of the K dimensional space, which makes them linearly separable. Hence, a well trained model attempts to achieve linear separability in the estimated posterior features.

The properties of posterior features discussed in this section can influence the choice of the second MLP classifier in the following ways:

1) Since the posterior features are trained to be linearly separable and have a sparse distribution, a simpler classifier (in terms of model capacity) may be sufficient at the second stage of the hierarchy. We validate this hypothesis in Section V-B.
2) Since the posterior features have lesser variability, the second MLP could be trained with lesser amount of training data. We test this hypothesis in Section V-C.

B. Complexity of the Second MLP

In this section, we study the effect of the model capacity (in terms of the number of parameters) of the second MLP in the hierarchical system on the phoneme recognition accuracies. Fig. 10 is a plot the phoneme recognition accuracies obtained by using the hierarchical approach, as a function of the number of parameters in the second MLP classifier (relative to the number of parameters in the first MLP). The number of parameters is controlled by reducing the size of the hidden layer until it equals the size of the input layer. On both TIMIT as well as CTS, the second MLP is trained using a temporal context of 230 ms. The horizontal dotted lines in the plot indicate the recognition accuracies obtained by using the output of the first MLP classifier.

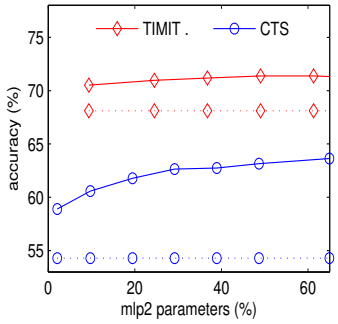


Fig. 10. Phoneme recognition accuracies as a function of the number of parameters in the second MLP classifier (relative to the number of parameters in the first MLP classifier, which has a size of $351 \times 1000 \times 40$ on TIMIT, and a size of $351 \times 5000 \times 45$ on CTS). In both cases, a temporal context of 230 ms is applied at the input of the second MLP, and the horizontal lines indicate the recognition accuracies obtained by using a single MLP system.

It can be seen from the figure that on both TIMIT as well as CTS, the recognition accuracies drop with the reduction in the number of parameters, and the drop in accuracy is more significant in the case of CTS. Nonetheless, the hierarchical system still outperforms the single MLP based system on both the tasks. The second MLP with just 20% of the parameters in the first MLP can still yield significantly higher recognition accuracies over the single MLP based system.

As an extreme case, a single layer perceptron (SLP) is used as a second classifier in the hierarchical system. It can be seen from Table VII that even a linear classifier in the second stage of the hierarchy can yield higher recognition accuracies (2.3% and 1.1% respectively on TIMIT and CTS respectively) when compared to the baseline system.

TABLE VII
PHONEME RECOGNITION ACCURACIES OBTAINED BY HIERARCHICAL POSTERIOR ESTIMATION USING MULTILAYER AND SINGLE LAYERED PERCEPTRON (SLP) CLASSIFIERS.

experiment	no hierarchy(%)	MLP hierarchy(%)	SLP hierarchy (%)
TIMIT	68.1	71.6	70.4
CTS	54.3	63.6	55.4

It can be recalled from Table IV that, on TIMIT, the phoneme recognition accuracy obtained by first order Volterra series approximation was only three percent lower compared to the accuracy obtained by directly evaluating the MLP, indicating the linear separable nature of the posterior features. Therefore, at the second stage of the hierarchy, an MLP classifier with fewer number of parameters (mildly nonlinear) is sufficient. On CTS, however, it can be seen that there is a 13.5% drop in recognition accuracy by approximating the MLP using first order Volterra series, which indicates that the posterior features estimated by the first MLP are not as linearly separable as those in TIMIT. This explains the higher drop in recognition accuracies with the reduction in the number of parameters on CTS task.

C. Size of Training Data

In this section, we study the effect of the amount of data required to train the second MLP in the hierarchical system on the phoneme recognition accuracies. In Fig. 11, we plot the phoneme recognition accuracies obtained by using the hierarchical approach as a function of the amount of training data used to train the second MLP classifier (relative to the amount of training data used to train the first MLP classifier). The amount of training data is controlled by randomly dropping the sentences in the training set. It can be seen that even with 80% reduction in the training data, the hierarchical system yields higher recognition accuracies when compared to the baseline system.

In this work, in order to speed up the training time on the CTS task, the training data was split into two halves, and the two MLPs in the hierarchical system were trained on the disjoint data sets. By training the hierarchical system using the above strategy, where the MLPs have sizes $351 \times 5000 \times 45$ and $1035 \times 1334 \times 45$, we obtained a recognition accuracy of 63.6%. However, only a slight improvement in recognition accuracy, about 0.7%, is obtained by training both the MLPs in the hierarchical system on the full 232 hours of data. Moreover, the training strategy for the hierarchical system - same training set or disjoint sets - did not affect the recognition accuracies.

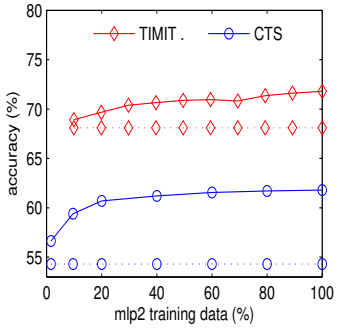


Fig. 11. Phoneme recognition accuracies as a function of the amount of data used to train the second MLP. 100% data corresponds to 153 minutes on TIMIT, and 116 hours on CTS. An MLP with fewer number of hidden nodes (200 on TIMIT and 400 on CTS) is used. In both cases, a temporal context of 230 ms is applied at the input of the second MLP. The horizontal lines indicate the accuracies obtained by using a single MLP based system.

VI. Discussion

We investigated a simple hierarchical system consisting of two MLP classifiers in tandem. The second MLP is trained using the posterior features estimated by the first with a temporal context of 150-230 ms. The effectiveness of the hierarchical system is a consequence of the long temporal context being applied in the posterior features, which are trained to be linearly separable, possess a sparse distribution, and lesser nonlinguistic variabilities.

A similar hierarchical system was previously studied in [19], where a CRF was used at the second stage of the system. It was argued that the system learns the phonetic confusion patterns in the posterior features estimated by the MLP. The findings from the present study further strengthens this argument. We show using Volterra analysis that the second MLP indeed learns the phonetic-temporal patterns which capture the phonetic confusions at the output of the first MLP. In addition, we also showed that it learns the phonotactics of the language as observed in the training data.

In the following subsections, we discuss some of the interesting aspects of the MLP based hierarchical system and possible future directions.

A. Choice of the Subword Units

In this work, the second MLP is trained on posterior features where each dimension corresponds to a phoneme. Further improvements in recognition accuracies have been observed using posterior features corresponding to the sub-phonemic states, *e.g.,* three states per phoneme [12]. A Volterra analysis of the second MLP classifier in such a scenario would reveal the phonetic-temporal patterns in the sub-phonemic posterior feature space.

The second MLP could also be trained using posterior features corresponding to the articulatory or phonological attributes of phonemes, *e.g.,* place and manner of articulation. A Volterra analysis of the second MLP in this case would reveal the articulatory-temporal patterns that are learned for each of the phonemes. Similar hierarchical systems have been previously studied where the second classifier is a RNN [33] or a CRF model [18].

B. Choice of the classifiers

In this work, the second MLP classifier in the hierarchical system is trained using posterior probabilities of phonemes conditioned on acoustic features, which are estimated by an MLP. In general, however, these phonetic class conditional probabilities could be estimated using other statistical models as well. For example, in an earlier work, the posterior probabilities of phonemes were estimated using a GMM, and similar improvements in recognition accuracies were observed [57].

The basic idea is to transform the acoustic features into posterior features corresponding to linguistically meaningful units such as phonemes, sub-phonemic states, or articulatory attributes using any classifier. In the posterior feature space, the temporal information spanning durations as long as 250 ms can be effectively learned.

C. The Second MLP as a Matched Filter

In the hierarchical system discussed in this paper, the second MLP can be viewed as a discriminatively trained nonlinear matched filter. Matched filters have been investigated previously in phoneme spotting in [58], where the matched filter for each phoneme was derived independently by averaging its phoneme posterior trajectory. The width of the matched filter implicitly captured the duration of the phoneme. The phoneme posteriors are multiplied with their respective matched filters and peaks are picked to spot phonemes.

D. Choice of the Databases

Experiments were performed on two databases (TIMIT and CTS), mainly to confirm the effectiveness of the hierarchical system in different data conditions. The results on the two tasks also exhibit certain differences. Firstly, the improvement in recognition accuracies obtained using the hierarchical approach is much higher on CTS, about 9.3%, when compared to 3.5% on TIMIT. Secondly, on TIMIT, the recognition accuracy obtained by first order Volterra series approximation is just 3% lower to that obtained by a direct evaluation of the MLP. In contrast, this difference is about 13.5% on CTS.

The TIMIT and CTS tasks differ in three aspects namely, the channel conditions (microphone versus telephone), the speaking styles (read speech versus conversational), and the labeling strategy (hand labeling versus forced alignment). It is not clear from the present study how the speaking style or the labeling strategy affects the hierarchical system as the experimental conditions differ in more than one respect.

These aspects can be studied using carefully designed experiments. For example, the impact of speaking styles on the hierarchical system can be studied by using two different databases which differ only in the speaking style, with all other relevant factors (the channel conditions, the labeling strategy, etc) the same. In such a scenario, the differences in the Volterra kernels of the second MLP for the two systems will bring out the impact of speaking style.

E. MLP based Hierarchical system for Adaptation

A potential application of the MLP based hierarchical system is in task adaptation. At the first stage of the hierarchical system, a well trained MLP available off-the-shelf could be used. The second MLP is trained on the posterior features estimated for the target task (adaptation data). It has already been observed that the second MLP in the hierarchy requires fewer number of parameters and can be trained using lesser amount of data, making it an ideal case for adaptation, especially in scenarios where the training data is limited.

VII. Summary and Conclusions

We investigated a simple hierarchical architecture for estimating the posterior probabilities of phonemes. The system consisted of two MLP classifiers in tandem. The first MLP is trained on PLP features, with a temporal context of 90 ms. The second MLP is trained on the posterior probabilities of phonemes (posterior features) estimated by the first classifier,

but with a relatively longer temporal context of around 150-230 ms. The hierarchical system yielded an absolute improvement of 3.5% and 9.3% over the conventional single MLP based system on TIMIT and CTS databases respectively.

The posterior features are endowed with two important properties. Firstly, they are trained to be linearly separable and possess a sparse distribution. Secondly, the posterior features carry very little information on the undesirable nonlinguistic variabilities such as speaker and noise characteristics. In other words, the posterior features represent the soft-decisions on the underlying sequence of phonemes, and are much simpler to classify. Consequently, the second MLP classifier can effectively learn the contextual information present in the temporal trajectories of the posterior features, spanning about 230 ms of context.

In order to unearth the phonetic-temporal patterns learned by the second MLP classifier, we applied Volterra series to model the second stage in the hierarchical system, and analyzed its first order Volterra kernels (linear part of the nonlinear system). The analysis of the linear Volterra kernels showed that the second MLP has effectively captured the phonetic confusion patterns at the output of the first classifier, as well as the phonotactics of the language, as observed in the training data.

Furthermore, we demonstrated that a simpler MLP with fewer number of parameters is sufficient at the second stage in the hierarchy, and that it can be trained using lesser amount of training data. We attribute this to the salient properties of the posterior features such as lesser nonlinguistic variabilities, sparse distribution, and linear separability.

APPENDIX A
NORMALIZATION OF POSTERIOR FEATURES

The expression for the posterior features is given by (1). In the following derivation, we drop the subscript for time t and simplify the notations by denoting the event $q_t = k$ by simply q_k. The model for the first MLP is denoted by Θ. Subsequently, (1) reduces to $x_k = P(q_k \mid \mathbf{f}, \Theta)$, where q_k denotes the phoneme, $\mathbf{f}$ denotes the input feature vector. The mean of the component k in the posterior feature vector is given by

$$\begin{aligned} m_k &= E_{\mathbf{f}}\left[x_k\right] \\ &= E_{\mathbf{f}}\left[P(q_k \mid \mathbf{f}, \Theta)\right] \\ &= \int p(\mathbf{f}) P(q_k \mid \mathbf{f}, \Theta)\ d\mathbf{f} \\ &= P(q_k \mid \Theta) \end{aligned} \tag{14}$$

Hence, the sample mean of the posterior features is an estimate of the prior probability of the phonemes q_k. In the above simplification, the property $p(\mathbf{f} \mid \Theta) = p(\mathbf{f})$ is exploited. The mean and variance of the posterior features are related as

$$\begin{aligned} \sigma_k^2 + m_k^2 &= E_{\mathbf{f}}\left[(x_k)^2\right] \\ &= \int p(\mathbf{f})\ \frac{p(\mathbf{f} \mid q_k, \Theta) P(q_k \mid \Theta)}{p(\mathbf{f} \mid \Theta)}\ x_k\ d\mathbf{f} \\ &= P(q_k \mid \Theta) \int p(\mathbf{f} \mid q_k, \Theta)\ x_k\ d\mathbf{f} \\ &= P(q_k \mid \Theta)\ E_{\mathbf{f}|q_k}\left[x_k\right] \end{aligned} \tag{15}$$

The conditional expectation in the above expression can be estimated as the average posterior probability of a phoneme obtained using data belonging to that particular phoneme only. If $\hat{x}_k$ denotes the scaled likelihood of the phoneme q_k, and given by

$$\hat{x}_k = \frac{x_k}{m_k} = \frac{P(q_k \mid \mathbf{f}, \Theta)}{P(q_k \mid \Theta)},$$

(15) can be expressed using (14) as

$$\frac{\sigma_k^2}{m_k^2} + 1 = E_{\mathbf{f}|q_k}\left[\hat{x}_k\right] \tag{16}$$

The posterior feature vector component, normalized to zero mean and unit variance $\hat{\hat{x}}_k$ can be be simplified using (16) as

$$\hat{\hat{x}}_k = \frac{x_k - m_k}{\sigma_k} = \frac{\hat{x}_k - 1}{\left[E_{\mathbf{f}|q_k}\left[\hat{x}_k\right] - 1\right]^{\frac{1}{2}}} \tag{17}$$

From (17), it is clear that mean and variance normalization on the posterior features is equivalent to taking scaled likelihoods as features. In other words, by taking scaled likelihoods as features and normalizing them to zero mean and unit variance would yield the same features as in (17). The only difference is that in the latter, the prior probabilities are estimated by normalizing the relative frequency of the phonetic labels in the training data. In the above formulation, the priors are estimated using the MLP model. In effect, by normalizing the posterior feature to zero mean and unit variance, the effect of priors in them are removed.

ACKNOWLEDGEMENTS

This work was supported by the Swiss National Science Foundation under the Indo-Swiss joint research program on keyword spotting (KEYSPOT), the Swiss National Center for Competence in Research (NCCR) under the Interactive Multimodal Information Management (IM2) project, and by the DARPA GALE program. Any opinions, findings and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of funding agencies. The authors thank Prof. B. Yegnanarayana from IIIT-Hyderabad, India and Dr. S. R. M. Prasanna from IIT-Guwahati, India for the fruitful discussions during the initial part of this work.

REFERENCES

[1] N. Morgan *et al.*, "Pushing the Envelope - Aside," *IEEE Signal Process. Magazine*, vol. 22, no. 5, pp. 81–88, 2005.

[2] A. Stolcke *et al.*, "Recent Innovations in Speech-to-Text Transcription at SRI-ICSI-UW," *IEEE Trans. Audio. Speech. Language. Process.*, vol. 14, no. 5, pp. 1729–1744, 2006.

[3] Q. Zhu, B. Chen, N. Morgan, and A. Stolcke, "On Using MLP Features in LVCSR," *Proc. of Interspeech*, pp. 921–924, 2004.

[4] P. Fousek, L. Lamel, and J.-L. Gauvain, "Transcribing Broadcast Data using MLP Features," *Proc. of Interspeech*, pp. 1433–1436, 2008.

[5] J. Park, F. Diehl, M. Gales, M. Tomalin, and P. Woodland, "Training and Adapting MLP Features for Arabic Speech Recognition," *Proc. of IEEE Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 4461–4464, 2009.

[6] H. Bourlard and N. Morgan, *Connectionist Speech Recognition – A Hybrid Approach*. Kluwer Academic Publishers, 1994.

[7] M. Richard and R. Lippmann, "Neural Network Classifiers Estimate Bayesian *a posteriori* Probabilities," *Neural Computation*, vol. 3, pp. 461–483, 1991.

[8] S. Ikbal, "Nonlinear Feature Transformations for Noise Robust Speech Recognition," Ph.D. dissertation, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, 2004.

[9] H. Misra, "Multi-stream Processing for Noise Robust Speech Recognition," Ph.D. dissertation, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, 2006.

[10] H. Hermansky, D. Ellis, and S. Sharma, "Tandem Connectionist Feature Extraction for Conventional HMM Systems," *Proc. of IEEE Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 1635–1638, 2000.

[11] G. Aradilla, "Acoustic Models for Posterior Features in Speech Recognition," Ph.D. dissertation, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, 2008.

[12] J. Pinto, B. Yegnanarayana, H. Hermansky, and M. Magimai.-Doss, "Exploiting Contextual Information for Improved Phoneme Recognition," *Proc. of IEEE Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 4449–4452, 2008.

[13] J. Pinto, S. Prasanna, B. Yegnanarayana, and H. Hermansky, "Significance of Contextual Information in Phoneme Recognition," Idiap Research Institute, Tech. Rep. 28, 2007.

[14] S. Boyd, L. O. Chua, and C. A. Desoer, "Analytical Foundations of Volterra Series," *IMA Journal of Mathematical Control and Information*, vol. 1, pp. 243–282, 1984.

[15] V. Volterra, *Theory of Functionals and of Integro-Differential Equations*. Dover, New York, 1930.

[16] D. Ellis and N. Morgan, "Size Matters: An Empirical Study of Neural Network Training for Large Vocabulary Continuous Speech Recognition," *Proc. of IEEE Conf. Acoust. Speech Signal Process. (ICASSP)*, vol. 2, pp. 1013–1016, 1999.

[17] P. Schwarz, P. Matejka, and J. Cernocky, "Hierarchical Structures of Neural Networks for Phoneme Recognition," *Proc. of IEEE Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 325–328, 2006.

[18] J. Morris and Fosler-Lussier, "Conditional Random Fields for Integrating Local Discriminative Classifiers," *IEEE Trans. Audio. Speech. Language. Process.*, vol. 16, no. 3, pp. 617–628, 2008.

[19] E. Fosler-Lussier and J. Morris, "CRANDEM Systems: Conditional Random Field Acoustic Models for Hidden Markov Models," *Proc. of IEEE Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 4049–4052, 2008.

[20] H. Ketabdar and H. Bourlard, "Hierarchical Integration of Phonetic and Lexical Knowledge in Phone Posterior Estimation," *Proc. of IEEE Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 4065–4068, 2008.

[21] Y. LeCun, L. Bottou, G. Orr, and K.-R. Muller, "Efficient BackProp," in *Neural Networks: Tricks of the Trade*, ser. Lecture Notes in Computer Science, G. Orr and K.-R. Muller, Eds. Springer-Verlag, 1998, no. 1524, pp. 9–50.

[22] H. Hermansky and S. Sharma, "Temporal Patterns (TRAPs) in ASR of Noisy Speech," *Proc. of IEEE Conf. Acoust. Speech Signal Process. (ICASSP)*, vol. 1, pp. 289–292, 1999.

[23] B. Chen, S. Chang, and S. Sivadas, "Learning Discriminative Temporal Patterns in Speech: Development of Novel TRAPS-like Classifiers," *Proc. of ICSLP*, pp. 429–432, 2001.

[24] J. Allen, "How do Humans Process and Recognize Speech?" *IEEE Trans. Speech. Audio. Process.*, vol. 2, pp. 567–577, 1994.

[25] H. Fletcher, *Speech and Hearing in Communication*, ASA edition ed. Acoustical Society of America, 1995.

[26] H. Bourlard and S. Dupont, "A New ASR Approach based on Independent Processing and Recombination of Partial Frequency Bands," *Proc. of ICSLP*, pp. 422–425, 1996.

[27] S. Tibrewala and H. Hermansky, "Sub-Band Based Recognition of Noisy Speech," *Proc. of IEEE Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 1255–1258, 1997.

[28] H. Hermansky and P. Fousek, "Multi-Resolution RASTA Filtering for Tandem based ASR," *Proc. of Interspeech*, pp. 361–364, 2005.

[29] F. Valente and H. Hermansky, "Hierarchical and Parallel Processing of Modulation Spectrum for ASR Applications," *Proc. of IEEE Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 4165–4168, 2008.

[30] ——, "On the Combination of Auditory and Modulation Frequency Channels for ASR Applications," *Proc. of Interspeech*, pp. 2242–2245, 2008.

[31] H. Ketabdar, J. Vepa, S. Bengio, and H. Bourlard, "Using More Informative Posterior Probabilities for Speech Recognition," *Proc. of IEEE Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 29–32, 2006.

[32] A. Robinson, "An Application of Recurrent Nets to Phone Probability Estimation," *IEEE Trans. Neural Networks*, vol. 5, no. 2, pp. 298–305, 1994.

[33] S. U. Khan, G. Sharma, and P. Rao, "Speech Recognition using Neural Networks," *IEEE Conference on Industrial Technology*, vol. 2, pp. 432–437, 2000.

[34] Y. Abdel-Haleem, "Conditional Random Fields for Continuous Speech Recognition," Ph.D. dissertation, University of Sheffield, 2006.

[35] F. Grezl, M. Karafiat, S. Kontar, and J. Cernosky, "Probabilistic and Bottleneck Features for LVCSR of Meetings," *Proc. of IEEE Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 757–760, 2007.

[36] D. Povey *et al.*, "FMPE: Discriminatively Trained Features for Speech Recognition," *Proc. of IEEE Conf. Acoust. Speech Signal Process. (ICASSP)*, vol. 1, pp. 961–964, 2005.

[37] D. Povey and P. Woodland, "Minimum Phone Error and I-Smoothing for Improved Discriminative Training," *Proc. of IEEE Conf. Acoust. Speech Signal Process. (ICASSP)*, vol. 1, pp. 105–108, 2002.

[38] K. Sim and M. Gales, "Discriminative Semi-parametric Trajectory Models for Speech Recognition," *Computer Speech and Language*, vol. 21, no. 4, pp. 669–687, 2007.

[39] J. Pinto, M. Magimai.-Doss, H., and H. Bourlard, "MLP Based Hierarchical System for Task Adaptation in ASR," *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2009.

[40] K.-F. Lee and H.-W. Hon, "Speaker-Independent Phone Recognition using Hidden Markov Models," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, no. 11, pp. 1641–1648, 1989.

[41] G. Evermann *et al.*, "Development of the 2003 CU-HTK Conversational Telephone Speech Transcription System," *Proc. of IEEE Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 249–252, 2004.

[42] P. C. Woodland *et al.*, "The CU-HTK English CTS System," *Proc. of the Rich Transcription Workshop*, 2003. [Online]. Available: http://www.itl.nist.gov/iad/mig/tests/rt/2003-spring/presentations/cts-slides.2up.letter.pdf

[43] J. J. Godfrey, E. C. Holliman, and J. Mcdaniel, "SWITCHBOARD: Telephone Speech Corpus for Research and Development," *Proc. of IEEE Conf. Acoust. Speech Signal Process. (ICASSP)*, vol. 1, pp. 517–520, 1992.

[44] T. Hain *et al.*, "The Development of AMI System for Transcription of Speech in Meetings," in *Machine learning for Multimodal Interaction: 2nd International Workshop, Revised Selected Papers*, S. Renals and S. Bengio, Eds. Springer-Verlag, 2005, no. 3869, pp. 344–356.

[45] S. Fitt, "Documentation and User Guide to UNISYN Lexicon and Post-lexical Rules," Center for Speech Technology Research, University of Edinburgh, Tech. Rep., 2000.

[46] "The ICSI Quicknet Software Package." [Online]. Available: http://www.icsi.berkeley.edu/Speech/qn.html

[47] "SRILM - The SRI Language Modeling Toolkit." [Online]. Available: http://www.speech.sri.com/projects/srilm

[48] D. Moore *et al.*, "Juicer: A Weighted Finite State Transducer Speech Decoder," in *Machine learning for Multimodal Interaction: 3rd International Workshop, Revised Selected Papers*, S. Renals, S. Bengio, and J. Fiscus, Eds. Springer-Verlag, 2006, no. 4299, pp. 285–296.

[49] J. Pinto, G. Sivaram, H. Hermansky, and M. Magimai.-Doss, "Volterra Series for Analyzing MLP Based Phoneme Posterior Estimator," Idiap Research Institute, Tech. Rep. 69, 2008.

[50] ——, "Volterra Series for Analyzing MLP Based Phoneme Posterior Estimator," *Proc. of IEEE Conf. Acoust. Speech Signal Process. (ICASSP)*, 2009.

[51] V. Marmarelis, *Nonlinear Dynamic Modeling of Physiological Systems*. Wiley, 2004.

[52] W. Gray and B. Nabet, "Volterra Series Analysis and Synthesis of a Neural Network for Velocity Estimation," *IEEE Trans. Systems. Man. and Cybernetics*, vol. 29, pp. 190–197, 1999.

[53] W. Knecht, "Nonlinear Noise Filtering and Beamforming using the Perceptron and its Volterra Approximation," *IEEE Trans. on Speech and Audio Process.*, vol. 2, no. 1, pp. 55–62, 1994.

[54] G. Stegmayer, "Volterra Series and Neural Networks to model an Electronic Device Nonlinear Behavior," *Proc. of IEEE Conf. Neural Networks*, vol. 4, pp. 2907–2910, 2004.

[55] D. Ellis, R. Singh, and Sivadas, "Tandem Acoustic Modeling in Large-vocabulary Recognition," *Proc. of IEEE Conf. Acoust. Speech Signal Process. (ICASSP)*, vol. 1, pp. 517–520, 2001.
[56] S. Sivadas and H. Hermansky, "Hierarchical Tandem Feature Extraction," *Proc. of IEEE Conf. Acoust. Speech Signal Process. (ICASSP)*, vol. 1, pp. 809–812, 2002.
[57] J. Pinto and H. Hermansky, "Combining Evidence from a Generative and a Discriminative Model in Phoneme Recognition," *Proc. of Interspeech*, pp. 2414–2417, 2008.
[58] M. Lehtonen, P. Fousek, and H. Hermansky, "Hierarchical approach for spotting keywords," Idiap Research Institute, Tech. Rep. 41, 2005.

Joel Pinto is a Research Assistant at Idiap Research Institute and a PhD candidate at Ecole Polytechnique Fedéralé du Lausanne (EPFL) Switzerland. He received the M.E. degree in Electrical Engineering from the Indian Institute of Science, Bangalore, India in 2003 and the B.E. degree in Electronics and Communication Engineering from the Manipal Institute of Technology, India in 2001.

From Feb. 2003 to Aug. 2005, he was with Hewlett Packard Labs India, Bangalore working on automatic speech recognition for spoken dialog systems. His research interests include acoustic modeling for speech recognition, signal processing, and machine learning.

Sivaram Garimella received the M.E. degree in Signal Processing in 2006 from the Indian Institute of Science, Bangalore. He is currently a graduate student in the Department of Electrical and Computer Engineering and a research assistant in the Center for Language and Speech Processing (CLSP) at Johns Hopkins University, Baltimore.

From September 2006 to August 2007, he was a software engineer at Muvee Technologies, Singapore, where he worked on video transitions and effects. He was a research assistant in IDIAP research institute, Switzerland, during September 2007 and January 2009, where he worked on automatic speech recognition. His research interests include feature extraction for automatic speech recognition and machine learning.

Mathew Magimai.-Doss (S '03, M'05) received the B.E. in Instrumentation and Control Engineering from the University of Madras, India in 1996; the M.S. by Research in Computer Science and Engineering from the Indian Institute of Technology, Madras, India in 1999; the PreDoctoral diploma and the Docteur ès Sciences (PhD) from École Polytechnique de Fédérale Lausanne (EPFL), Switzerland in 2000 and 2005, respectively. From April 2006 till March 2007, he was a postdoctoral fellow at International Computer Science Institute, Berkeley, USA. Since April 2007, he has been working as a research scientist at Idiap Research Institute, Martigny, Switzerland . His research interests include speech processing, automatic speech and speaker recognition, statistical pattern recognition, and artificial neural networks.

Hynek Hermansky is a Full Professor of the Electrical and Computer Engineering at the Johns Hopkins University in Baltimore, Maryland. He is also a Professor at the Brno University of Technology, Czech Republic, an Adjunct Professor at the Oregon Health and Sciences University, Portland, Oregon, and an External Fellow at the International Computer Science Institute at Berkeley, California. He is a Fellow of IEEE for Invention and development of perceptually-based speech processing methods, is in charge of plenary sessions at the upcoming 2011 ICASSP in Prague, was the Technical Chair at the 1998 ICASSP in Seattle and an Associate Editor for IEEE Transaction on Speech and Audio. Further, he is Member of the Editorial Board of Speech Communication, holds 6 US patents and authored or co-authored over 200 papers in reviewed journals and conference proceedings. He has been working in speech processing for over 30 years, previously as a Director of Research at the IDIAP Research Institute, Martigny and an Adjunct Professor at the Swiss Federal Institute of Technology in Lausanne, Switzerland, a Professor and Director of the Center for Information Processing at OHSU Portland, Oregon, a Senior Member of Research Staff at U S WEST Advanced Technologies in Boulder, Colorado, a Research Engineer at Panasonic Technologies in Santa Barbara, California, and a Research Fellow at the University of Tokyo. He holds Dr.Eng. Degree from the University of Tokyo, and Dipl. Ing. Degree from Brno University of Technology, Czech Republic. His main research interests are in acoustic processing for speech recognition.

Hervé Bourlard received the Electrical and Computer Science Engineering degree and the Ph.D. degree in Applied Sciences both from "Faculté Polytechnique de Mons", Mons, Belgium. After having been a member of the Scientific Staff at the Philips Research Laboratory of Brussels and an R&D Manager at L&H SpeechProducts, he is now Director of the Idiap Research Institute (www.idiap.ch), Full Professor at the Swiss Federal Institute of Technology at Lausanne (EPFL), and Director of a National Center of Competence in Research in "Interactive Multimodal Information Management" (IM2, www.im2.ch). Having spent (since 1988) several long-term and short-term visits (initially as a Guest Scientist) at the International Computer Science Institute (ICSI, www.icsi.berkeley.edu) in Berkeley, CA, he is now a member of the ICSI Board of Trustees.

His main interests are in signal processing, statistical pattern classification, multi-channel processing, artificial neural networks, and applied mathematics, with applications to speech and natural language modeling, speech and speaker recognition, computer vision, and multimodal processing.

H. Bourlard is the author/coauthor/editor of 4 books and over 250 reviewed papers (including one IEEE paper award) and book chapters. He is an IEEE Fellow for "contributions in the fields of statistical speech recognition and neural networks". He is (or has been) a member of the program/scientific committees of numerous international conferences (e.g., General Chairman of IEEE Workshop on Neural Networks for Signal Processing 2002, Co-Technical Chairman of ICASSP 2002, General Chairman of Interspeech 2003) and on the editorial board of several journals (e.g., past co-Editor-in-Chief of "Speech Communication").

Over the last 20 years, Hervé Bourlard has initiated and coordinated numerous large international research projects, as well as multiple collaborative projects with industries. He is an appointed expert for the European Commission and, from 2002 to 2007, was also part of the European Information Society Technology Advisory Group (ISTAG).