IDIAP RESEARCH REPORT

VOLTERRA SERIES FOR ANALYZING MLP BASED PHONEME POSTERIOR PROBABILITY ESTIMATOR

Joel Praveen Pinto G. S. V. S. Sivaram
Hynek Hermansky Mathew Magimai-Doss

Idiap-RR-69-2008

OCTOBER 2008

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

VOLTERRA SERIES FOR ANALYZING MLP BASED PHONEME POSTERIOR ESTIMATOR

Joel Pinto [1,2], *G.S.V.S. Sivaram* [1,2], *H. Hermansky* [1,2], *M. Magimai.-Doss* [1]

[1] Idiap Research Institute, Martigny, Switzerland
[2] École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
{jpinto,sgarimel,hynek,mathew}@idiap.ch

ABSTRACT

We present a framework to apply Volterra series to analyze multilayered perceptrons trained to estimate the posterior probabilities of phonemes in automatic speech recognition. The identified Volterra kernels reveal the spectro-temporal patterns that are learned by the trained system for each phoneme. To demonstrate the applicability of Volterra series, we analyze a multilayered perceptron trained using Mel filter bank energy features and analyze its first order Volterra kernels.

Index Terms— Volterra series, multilayered perceptrons, speech recognition

1. INTRODUCTION

Multilayered perceptron (MLP) based acoustic modeling is being extensively used in the state-of-the-art automatic speech recognition (ASR) [1][2]. The MLP is trained as a phoneme classifier, and estimates the posterior probabilities of the phonemes conditioned on the input features. The estimates of posterior probabilities are used in ASR typically as local acoustic scores in hybrid hidden Markov model (HMM) - artificial neural network system [3] or as features (after logarithm and principal component analysis transformation) to a standard HMM - Gaussian mixture model system [4]. Due to its usage in the latter approach, the estimated posterior probabilities are also known as phoneme posterior features.

MLP based acoustic modeling has been shown to improve recognition accuracies in ASR. However, once trained, the MLP is typically not further analyzed. The estimated posterior probabilities are typically evaluated using (i) frame-level phoneme classification accuracy (ii) phonetic confusion matrix (iii) mutual information between the estimated posterior probabilities and its ground truth phonetic labels or (iv) the final speech recognition accuracy. While the above metrics indicate the goodness of the phoneme posterior estimates, none of them reveal any information on the spectro-temporal patterns that the trained system has learned. An understanding of the properties of speech learned by the trained system could eventually lead to improvements in the present approaches.

One way to analyze the trained system is to treat it as a nonlinear black-box and present white Gaussian noise as input. The characteristics of the unknown system can be measured by cross-correlating the input white noise and the output of the system [5]. Such an approach is typically used to analyze unknown biological systems [6][7]. However, the three layered MLP based phoneme posterior estimator, which is typically used in ASR is simple enough and its trained model parameters are already known. In this work we focus on the analytical analysis of the trained system.

We formulate a framework to apply Volterra series [8] to analyze the trained MLPs. It is important to incorporate the feature extraction into this analysis because the identified Volterra kernels can then be interpreted as spectro-temporal patterns. The combined system is nonlinear and time-invariant, where the finite impulse response (FIR) filters used in feature extraction introduce memory and the activation functions in the MLP introduce nonlinearity. Volterra series has been used to model recurrent neural networks to analyze nonlinear properties of electronic devices [9]. The contributions of our work include (i) formulation of a framework to apply Volterra series to analyze MLPs estimating posterior probability of phonemes (ii) analytical identification of the Volterra kernels, (iii) addressing the effect of feature mean and variance normalization, (iv) as an example, application of Volterra series to analyze an MLP trained using Mel filter bank energies, and (v) discussion on the application of Volterra series to analyze MLPs trained using MFCC and MRASTA features.

2. PHONEME POSTERIOR ESTIMATOR

Fig. 1(a) is the block schematic of a typical phoneme posterior probability estimator, showing the feature extraction as well as the MLP classifier. This generic block schematic is applicable to typical speech recognition features such as Mel filter-bank cepstral coefficients (MFCCs), Mel filter-bank (MFB) energies, and multi-resolution relative spectra (MRASTA) features.

2.1. Auditory analysis

Auditory analysis is a common stage across almost all feature extraction techniques. Short time Fourier analysis is performed on speech signal with an analysis window of typically 25 ms and a frame shift of 10 ms. Auditory filters that are equally spaced in Mel or Bark frequency scale are applied on the Fourier magnitude spectrum, and log energies in the auditory channels are computed.

2.2. Feature specific LTI system

The trajectories of the log energies from the auditory analysis are then processed by a linear time-invariant (LTI) system whose impulse response is decided by the feature extraction being used. For Mel frequency cepstral coefficients (MFCC), this system consists of discrete cosine transform (DCT), the FIR filters required to compute the delta and delta-delta coefficients, and the filters creating a temporal context of features. In the case of multi-resolution relative spectra (MRASTA) features [10], the LTI system consists of a bank of zero mean filters whose shape is that of either first or second derivative of a Gaussian function. For Mel filter bank energy (MFB) features, the system consists of bank of time shifted Kronecker delta functions required to create a temporal context of features. Since the filters in this block operate on the trajectories of the auditory spectrum, this stage can be interpreted as filtering the modulation spectrum of the speech in the sub bands. From a mathematical perspective, the difference between the above feature extraction techniques is in the impulse response of the LTI system.

2.3. Feature Normalization

The input features to the MLP are normalized to zero mean and unit variance so that the operating point on the hidden activation function is in the linear region, leading to a faster convergence of the back propagation training algorithm [11]. Feature normalization also addresses to a certain extent the mismatch caused when the MLP is trained and tested on different data.

2.4. Multilayered perceptron

A three layered MLP is typically used in posterior feature extraction. The normalized features presented at the input layer of the MLP are projected to a higher dimensional hidden layer with sigmoid or hyperbolic tangent activation function. The output node of the MLP represents the basic modeling unit of speech such as a phoneme. Softmax nonlinearity is applied at the output layer and the model parameters are optimized using minimum cross entropy error criterion. It has been shown that MLP classifier with sufficient capacity, and trained on enough data estimates the Bayesian *a posteriori* probability of the output class, conditioned on the input features [12].

Feature extraction techniques are typically motivated by the perceptual or production properties of speech. But the parameters of the MLP classifier are optimized to achieve minimum phoneme classification error using the derived features on a cross-validation data. It is not obvious what spectro-temporal properties of speech the combined system has learned in the form of its trained weights. In this work, we use Volterra series to obtain insights about the phoneme posterior estimator.

3. VOLTERRA SERIES

An LTI system can be completely characterized by its impulse response function. Volterra series is an infinite series which can be used to express the input-output relationship in a nonlinear time-invariant system. Each term in the series is a multi-dimensional convolution between the input to the system and its Volterra kernels. The identified Volterra kernels completely characterize the nonlinear system. If $x(t)$ is the input to a nonlinear system and $y(t)$ its output, Volterra series expansion for the system can be expressed as

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

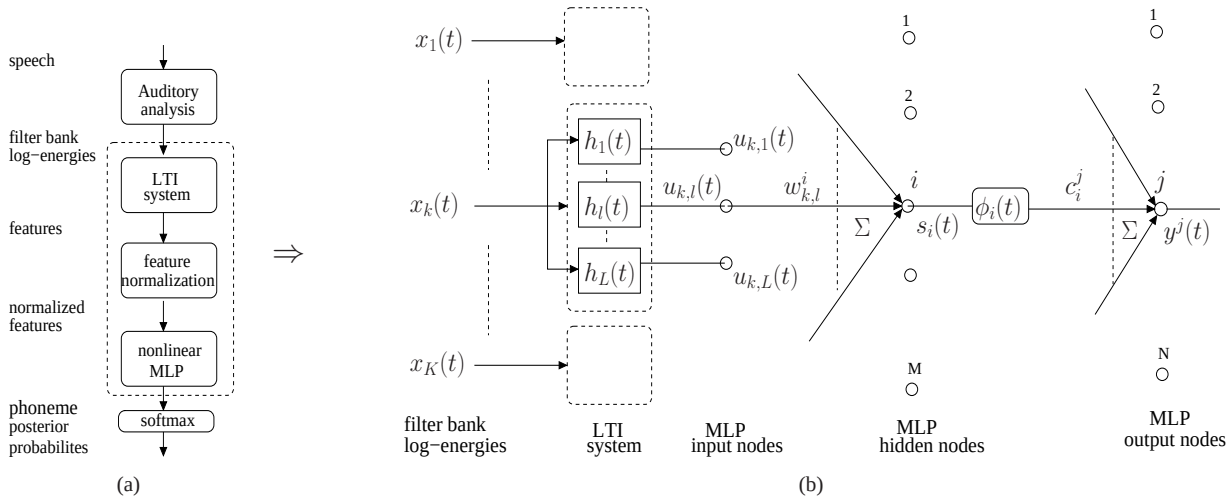


Fig. 1. *(a) Estimation of posterior probabilities of phonemes using an MLP. (b) Part of the system that is analyzed using Volterra series.*

where, $\{G_n\}$ is the set of Volterra functionals, and $\{g_n\}$ is the set of the Volterra kernels for the nonlinear system. The zeroth order Volterra functional is given by $G_0\left[g_0, x(t)\right] = g_0$; the first and second order functionals are given by

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

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

The first order Volterra functional $G_1\left[g_1, x(t)\right]$ is the linear convolutional integral, and its kernel $g_1(t)$ is the most familiar time-domain description of an LTI system (*i.e.* impulse response function). In the following section, we present a mathematical framework to apply Volterra series expansion to a three layered MLP estimating the posterior probabilities of phonemes.

The canonical structure of the Volterra series provides an intuitive framework to identify the linear, quadratic, and higher order components of the nonlinear system. When the system is unknown, *e.g.* biological systems, the nonlinear system is modeled using an alternative representation known as Wiener series expansion [13], whose functionals are orthogonal with respect to white Gaussian noise. The Wiener kernels are estimated using cross-correlation based methods [5]. The Volterra kernels are subsequently computed from the Wiener kernels. In the posterior feature extraction using a three layered MLP, we are able to identify the Volterra kernels analytically as discussed in the following section.

3.1. Volterra Kernel Identification: Three layered MLP

Fig. 1(b) shows a part of the phoneme posterior estimator that is modeled using Volterra series. It is a multi-input, multi-output, nonlinear time-invariant system comprising of an LTI filter bank followed by the MLP. The input to the system are log energies from the auditory analysis $x_k(t),\ k = 1, 2 \ldots K$, where K is the number of auditory channels. The output of the system is the accumulated sum $y^j(t),\ j = 1, 2 \ldots N$ before the output nonlinearity, where N is the number of output nodes in the MLP. The Volterra series expansion of such a system can be expressed as

$$y^j\left(t\right) = g_0^j \ + \ \sum_{k_1=1}^{K} \int_{\tau_1} g_{k_1}^j\left(\tau_1\right) x_{k_1}\left(t-\tau_1\right) d\tau_1 \ + \ \sum_{k_1=1}^{K}\sum_{k_2=1}^{K}$$

$$\int_{\tau_1}\int_{\tau_2} g_{k_1 k_2}^j\left(\tau_1, \tau_2\right) x_{k_1}\left(t-\tau_1\right) x_{k_2}\left(t-\tau_2\right) d\tau_1 d\tau_2 + \ldots \tag{1}$$

where, the terms $g_0^j,\ g_{k_1}^j\left(\tau_1\right),\ g_{k_1 k_2}^j\left(\tau_1, \tau_2\right)$ are the zeroth, first, and second order Volterra kernels respectively of the phoneme j. The variables $\tau_1, \tau_2 \ldots$ denote time, and $k_1, k_2 \ldots$ denote the frequencies on the Mel or Bark scale. We identify the above Volterra kernels in terms of the impulse response of the LTI system and the parameters of the MLP.

Even though the above system is a discrete-time system, we use continuous-time notations through out this paper for clarity. The LTI system, which is a part of feature extraction consists of a bank of L FIR filters, each with an impulse response of $h_l(t),\ l = 1, 2 \ldots L$. The component of the feature vector $u_{k,l}(t)$ is obtained by convolving the input $x_k(t)$ with the impulse response $h_l(t)$, and given by

$$u_{k,l}(t) = \int_{\tau} h_l(\tau)\, x_k(t-\tau)\, d\tau. \tag{2}$$

The MLP consists of $K \times L$ input nodes which is same as the dimension of the feature vector, M hidden nodes, and N output nodes. The input $s_i(t)$ to the hidden nonlinearity function $\phi_i(.)$ is the linear combination of the input features $u_{k,l}(t)$ weighted by the MLP weights from the input to the hidden layer $w^i_{k,l}$, and given by

$$s_i(t) = \sum_{k=1}^{K}\sum_{l=1}^{L} w^i_{k,l} u_{k,l}(t). \tag{3}$$

Here, we assume that features presented to the MLP are not normalized. Kernel identification for normalized features is discussed in section 3.2. The accumulated sum at the j^{th} output node is the linear combination of the outputs at the hidden layer and the weights connecting the hidden and the output layer of the MLP, and given by

$$y^j(t) = \sum_{i=1}^{M} c^j_i\ \phi_i(s_i(t)). \tag{4}$$

$\phi_i(.) = \phi(h_i + .)$ is the nonlinearity at the i^{th} hidden node, where h_i is the bias and $\phi(.)$ is the nonlinear activation function (sigmoid, hyperbolic tangent). To derive the Volterra kernels, $\phi_i(.)$ is approximated using a polynomial expansion of the form

$$\phi_i(s_i(t)) = a_{0,i} + a_{1,i} s_i(t) + a_{2,i} s_i(t)^2 + \ldots, \tag{5}$$

where the coefficients $a_{0,i}, a_{1,i} \ldots$ are scalar constants. Polynomial expansion of the nonlinearity and the estimation of the coefficients is discussed in section 3.3. By substituting (5) in (4), we obtain

$$\begin{aligned} y^j(t) &= \sum_{i=1}^{M} c^j_i\ \left[a_{0,i} + a_{1,i}\ s_i(t) + a_{2,i}\ s_i(t)^2 + \ldots\right]. \\ &= \sum_{i=1}^{M} c^j_i\ a_{0,i} + \sum_{i=1}^{M} c^j_i\ a_{1,i}\ s_i(t) + \sum_{i=1}^{M} c^j_i\ a_{2,i}\ s_i(t)^2 + \ldots \end{aligned} \tag{6}$$

By substituting (2) and (3) in (6) we obtain

$$y^j(t) = \sum_{i=1}^{M} c^j_i\ a_{0,i} + \sum_{i=1}^{M} c^j_i\ a_{1,i} \sum_{k_1=1}^{K}\sum_{l_1=1}^{L} w^i_{k_1 l_1} \int_{\tau_1} h_{l_1}(\tau_1)\, x_{k_1}(t-\tau_1)\, d\tau_1 +$$

$$\sum_{i=1}^{M} c^j_i a_{2,i} \sum_{k_1=1}^{K}\sum_{l_1=1}^{L}\sum_{k_2=1}^{K}\sum_{l_2=1}^{L} w^i_{k_1 l_1} w^i_{k_2 l_2} \int_{\tau_1}\int_{\tau_2} h_{l_1}(\tau_1)\, h_{l_2}(\tau_2)\, x_{k_1}(t-\tau_1)\, x_{k_2}(t-\tau_2)\, d\tau_1 d\tau_2 + \ldots$$

By exchanging summation and integration, and rearranging terms in the above equation, we obtain

$$y^j(t) = \left[\sum_{i=1}^{M} c^j_i\ a_{0,i}\right] + \sum_{k_1=1}^{K} \int_{\tau_1} \left[\sum_{i=1}^{M} c^j_i\ a_{1,i} \sum_{l_1=1}^{L} w^i_{k_1 l_1} h_{l_1}(\tau_1)\right] x_{k_1}(t-\tau_1)\, d\tau_1 +$$

$$\sum_{k_1=1}^{K}\sum_{k_2=1}^{K} \int_{\tau_1}\int_{\tau_2} \left[\sum_{i=1}^{M} c^j_i a_{2,i} \sum_{l_1=1}^{L}\sum_{l_2=1}^{L} w^i_{k_1 l_1} w^i_{k_2 l_2} h_{l_1}(\tau_1)\, h_{l_2}(\tau_2)\right] x_{k_1}(t-\tau_1)\, x_{k_2}(t-\tau_2)\, d\tau_1 d\tau_2 + \ldots$$

Volterra kernels are identified by comparing the above equation to the Volterra series expansion in (1). The first three Volterra kernels are given by

$$g_0^j = \sum_{i=1}^{M} c_i^j \; a_{0,i} \tag{7}$$

$$g_{k_1}^j (\tau_1) = \sum_{i=1}^{M} c_i^j \; a_{1,i} \sum_{l_1=1}^{L} w_{k_1 l_1}^i h_{l_1} (\tau_1) \tag{8}$$

$$g_{k_1 k_2}^j (\tau_1, \tau_2) = \sum_{i=1}^{M} c_i^j a_{2,i} \sum_{l_1=1}^{L} \sum_{l_2=1}^{L} w_{k_1 l_1}^i w_{k_2 l_2}^i h_{l_1} (\tau_1) h_{l_2} (\tau_2) \tag{9}$$

The identified Volterra kernels are functions of the impulse responses of the filters in the LTI system, and the parameters of the functions are determined by the weights of the MLP. Specifically, the first order Volterra kernel is a linear combination of the impulse responses of the filters. In the following section, we identify the Volterra kernels when the MLP is trained using features that are normalized to zero mean and unit variance as discussed in section 2.3.

3.2. Volterra Kernel Identification: Feature Normalization

Suppose that the feature vector component $u_{k,l}(t)$ has a mean $\mu_{k,l}$ and a standard deviation $\sigma_{k,l}$. The mean and standard deviation are estimated on the training data. The MLP is trained using features normalized to zero mean and unit variance, and given by

$$\hat{u}_{k,l} (t) = \frac{u_{k,l} (t) - \mu_{k,l}}{\sigma_{k,l}} \tag{10}$$

By substituting the normalized feature component in (3), we obtain

$$\begin{aligned} s_i (t) &= \sum_{k=1}^{K} \sum_{l=1}^{L} w_{k,l}^i \; \hat{u}_{k,l}(t) \\ &= \sum_{k=1}^{K} \sum_{l=1}^{L} w_{k,l}^i \frac{u_{k,l} (t) - \mu_{k,l}}{\sigma_{k,l}} \\ &= \hat{s}_i(t) \; - \; \Delta_i \end{aligned} \tag{11}$$

where,

$$\hat{s}_i(t) = \sum_{k=1}^{K} \sum_{l=1}^{L} \hat{w}_{k,l}^i \; u_{k,l}(t), \tag{12}$$

$$\hat{w}_{k,l}^i = \frac{w_{k,l}^i}{\sigma_{k,l}}, \quad and \tag{13}$$

$$\Delta_i = \sum_{k=1}^{K} \sum_{l=1}^{L} w_{k,l}^i \; \frac{\mu_{k,l}}{\sigma_{k,l}}. \tag{14}$$

The feature normalization can be incorporated into the parameters of the MLP by appropriately modifying the hidden bias and the weights connecting the input and hidden layers of the MLP. The output at the j^{th} output node can be written from (6) as

$$y^j (t) = \sum_{n=0}^{\infty} \sum_{i=1}^{M} \; c_i^j \; a_{n,i} \; \left(\hat{s}_i(t) - \Delta_i \right)^n . \tag{15}$$

By using binomial series expansion (15) can be rewritten as

$$y^j(t) = \sum_{n=0}^{\infty} \sum_{i=1}^{M} c_i^j \, a_{n,i} \sum_{r=0}^{n} \binom{n}{r} \, (s_i(t))^r \, (-\Delta_i)^{n-r} \tag{16}$$

By collecting the polynomial terms $(\hat{s}_i(t))^r$, (16) can be rewritten as

$$y^j(t) = \sum_{r=0}^{\infty} \sum_{i=1}^{M} c_i^j \, \hat{a}_{r,i} \, (\hat{s}_i(t))^r, \quad where \tag{17}$$

$$\hat{a}_{r,i} = \sum_{n=r}^{\infty} \binom{n}{r} a_{n,i} \quad (-\Delta_i)^{n-r} \tag{18}$$

Volterra kernels are identified by following the simplifications described in section 3.1 except for using (17) instead of (6). The first three Volterra kernels are identified as

$$g_0^j = \sum_{i=1}^{M} c_i^j \hat{a}_{0,i} \tag{19}$$

$$g_{k_1}^j(\tau_1) = \sum_{i=1}^{M} c_i^j \, \hat{a}_{1,i} \sum_{l_1=1}^{L} \hat{w}_{k_1 l_1}^i h_{l_1}(\tau_1) \tag{20}$$

$$g_{k_1 k_2}^j(\tau_1, \tau_2) = \sum_{i=1}^{M} c_i^j \hat{a}_{2,i} \sum_{l_1=1}^{L} \sum_{l_2=1}^{L} \hat{w}_{k_1 l_1}^i \hat{w}_{k_2 l_2}^i h_{l_1}(\tau_1) \, h_{l_2}(\tau_2) \tag{21}$$

It can be seen that the Volterra kernels are of the same mathematical form as those corresponding to unnormalized features, but the weights and the coefficients of polynomial expansion are appropriately modified by the mean and variance of the features. The new weights connecting the input and hidden layer of the MLP $\hat{w}_{k,l}^i$ is given by (13), and the new polynomial coefficients are given by (18). For example, the zeroth, first, and second order polynomial coefficients are given by

$$\hat{a}_{0,i} = a_{0,i} - \Delta_i a_{1,i} + \Delta_i^2 a_{2,i} - \Delta_i^3 a_{3,i} + \ldots$$

$$\hat{a}_{1,i} = a_{1,i} - 2\Delta_i a_{2,i} + 3\Delta_i^2 a_{3,i} - 4\Delta_i^3 a_{4,i} + \ldots$$

$$\hat{a}_{2,i} = a_{2,i} - 3\Delta_i a_{3,i} + 6\Delta_i^2 a_{4,i} - 10\Delta_i^3 a_{5,i} + \ldots$$

It can be seen from the expression for Volterra kernels (19)-(21) and (18) that due to feature normalization, the Volterra kernels are also in the form of an infinite series. However, in practice the order of the Volterra series as well as the Volterra kernels is decided by the order of the polynomial approximation of the hidden nonlinearity. Moreover, the coefficients $a_{n,i}$ in (18) approach towards zero as the order n is increased. In cases where the features are zero mean but non-unit variance [10], the polynomial coefficients remain unchanged as $\Delta_i = 0$, but MLP weights are appropriately scaled by the feature variance.

3.3. Polynomial expansion of the activation function

A key aspect in the derivation of the Volterra kernels is the polynomial expansion (5) of the nonlinearity at the hidden nodes. Polynomial expansion of activation functions such as sigmoid is divergent if approximated for all possible values of the input $(-\infty, \infty)$. However, 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.

Fig. 2(a) shows the histogram of the input (which includes the bias) to the sigmoid function at a hidden node, and is obtained on the cross-validation data. We fit a polynomial function of certain order in the range of values observed in the histogram, leaving out a small fraction on the tail. The coefficients of the polynomial are optimized to minimize the least mean square error between the sigmoid function and its polynomial approximation in the region of interest. Fig. 2(b) is the plot of the sigmoid activation function and its polynomial approximation. Since the hidden bias is incorporated in the polynomial expansion, the estimated coefficients are different for each hidden node.

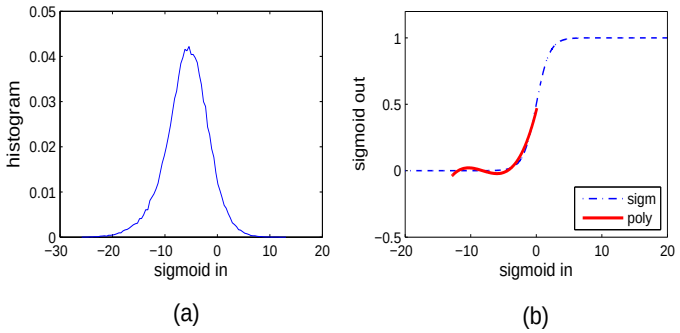


Fig. 2. *(a) Histogram of the input to the sigmoid at an hidden node. (b) Sigmoid function and its 3^{rd} order polynomial expansion.*

3.4. Interpretation of Volterra kernels

The first order Volterra kernel $g_k^j(t)$ (t denotes time, k denotes frequency, and j denotes the phoneme) is the linear transfer function of the posterior feature extraction system. The time-reversed linear kernel can be interpreted as a matched filter capturing the spectro-temporal patterns learned by the system. The second order kernel $g_{k_1 k_2}^j(t_1, t_2)$ for the phoneme j reveals the correlations across different frequency bands (k_1, k_2) at different times (t_1, t_2). Similarly the higher order Volterra kernels reveal the higher order correlations in the nonlinear system.

4. VOLTERRA ANALYSIS ON MFB FEATURES

To demonstrate the applicability of Volterra series expansion, we analyze a posterior feature extraction system, where the MLP trained on the standard TIMIT database using using Mel filter bank energy (MFB) features. The log-energies from the 26 auditory channels are presented to the MLP with a context of 170 ms. Hence the LTI system in Fig. 1 is a bank of 17 FIR filters with shifted Kronecker delta impulse response functions. The input layer of the MLP consists of 442 nodes, the hidden layer consists of 1000 nodes, and the output layer consists of 39 nodes corresponding to the number of phonemes. The training set consists of 153 minutes (375 speakers), cross-validation set consists of 34 minutes (87 speakers), and test set consists of 68 minutes (168 speakers) of speech.

The Volterra kernels are derived using (20). We fit a polynomial function of order 3 to the hidden nonlinearity, leaving out 5% of the points on the tail of the histogram. The identified kernels are applied in the Volterra series (1) to estimate the phoneme posterior probabilities. The estimated probabilities are evaluated by applying them in phoneme classification or isolated phoneme recognition experiments. Phoneme classification facilitates accurate analysis of the results as insertions and deletions are avoided. However, the trends observed in phoneme classification are also observed in phoneme recognition experiments. Viterbi algorithm is applied on the phoneme posterior probabilities with a minimum duration of three states per phoneme [3]. Table 1 shows the phoneme classification accuracy obtained by using linear and quadratic approximation of the MLP using Volterra series. The accuracy obtained using Volterra series should converge to the accuracy obtained using the MLP as the order of the series is increased.

model	series order	accuracy (%)
linear	1	38.2
quadratic	2	43.7
MLP	∞	77.9

Table 1. *Phoneme classification accuracy obtained by linear and quadratic approximation of the MLP using Volterra series.*

Fig. 3 shows the first order Volterra kernel for phonemes /iy/ (*e.g.* b**ea**t) and /eh/ (*e.g.* b**e**t). It can be seen that in the case of phoneme /iy/, the system has learned to emphasize 200-300 Hz frequency band which corresponds to its first formant. In case of /eh/, the system has learned to emphasize slightly higher frequency region of 400-500 Hz, which corresponds to its first formant. Fig. 4 shows the first order Volterra kernel for the phonemes /s/ (*e.g.* **s**ee) and /z/ (*e.g.* **z**oo). It can be seen that for

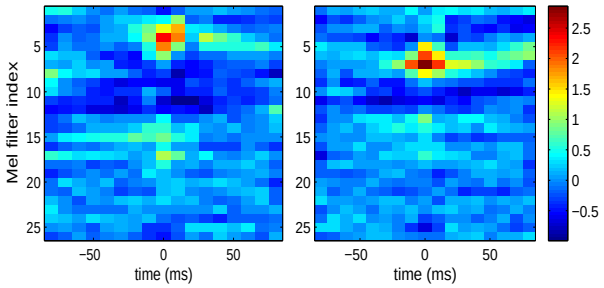


Fig. 3. *Linear Volterra kernels for phonemes /iy/ (left) and /eh/ (right)*

both these phonemes, the system has learned to emphasize the higher frequency regions. However, the unvoiced phoneme /s/ is distinguished from the voiced phoneme /z/ by the lack of energy in its low frequency region.

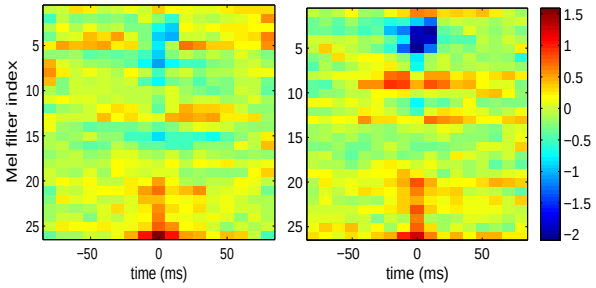


Fig. 4. *Linear Volterra kernels for phonemes /z/ (left) and /s/ (right)*

5. APPLICATION TO MRASTA AND MFCC FEATURES

In this section, we discuss the identification of the Volterra kernels when the MLP is trained using MRASTA, and the standard MFCC features. It can be recalled that to apply Volterra analysis, the system should be in the form given by Fig. 1, where the MLP is preceded by a FIR filter bank.

MRASTA features

In MRASTA feature extraction [10], the trajectories of log-energies from the auditory filters are processed by an LTI system consisting of a bank of FIR filters. The impulse response function of the filters are of the shape of either first or second derivative of a Gaussian function as shown in Fig. 5. The variance of the Gaussian function controls the time resolution of the filters. A typical implementation of MRASTA filter bank consists of eight first and second derivatives of the Gaussian function with standard deviation varying between 8 ms to 130 ms. The Volterra kernels are derived using (19)-(21). The features are zero mean since the Gaussian derivative functions are zero mean, and hence $\Delta_i = 0$ for the i^{th} hidden node in (18). The weights connecting the input and hidden layer are scaled by the variance of the feature component as (13).

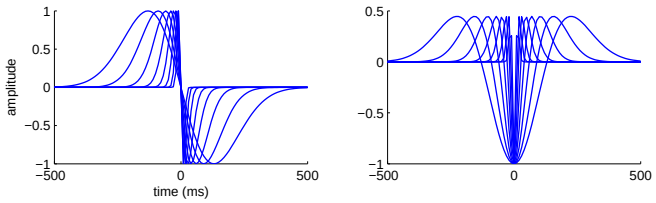
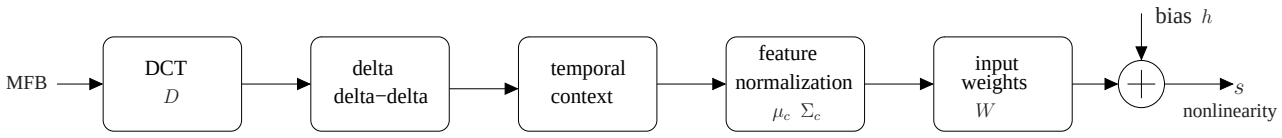


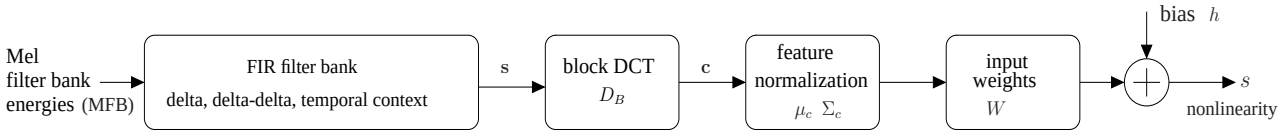
Fig. 5. *Impulse response of MRASTA filters.*

MFCC features

We discuss the MFCC features in its most generic form, where dynamic cepstral coefficients are also appended to the static coefficients, and the concatenated features are presented to the MLP with a certain (typically 90 ms) temporal context. The framework discussed in section 3.1 cannot be directly applied to MFCC features because the discrete cosine transform (DCT) operation mixes the energies from the auditory channels. However, the DCT operation can be incorporated in the MLP by appropriately modifying its parameters as shown in Fig. 6. This rearrangement does not change the functionality of the overall system in any way, but brings it to a form suitable for applying Volterra series expansion discussed so far.



(a) Standard computation of MFCC features and the MLP.



(b) The DCT operation is applied after FIR filters.

(c) The DCT coefficients, cepstral feature mean and variance is incorporated into the MLP parameters

Fig. 6. *Rearrangement of MFCC feature extraction to apply Volterra series analysis.*

In the following description, the size of the vectors and matrices are given in square brackets next to the variable name. Let N_f denote the number of Mel auditory filters and N_c denote the number of static cepstral coefficients derived using a DCT matrix D $[N_c \times N_f]$. As shown in Fig. 6(b), the dynamic cepstral coefficients (static, delta, delta-delta), and a temporal context of N_t frames are computed using a single filter bank comprising of $3N_t$ filters[1]. The impulse response for deriving static cepstral coefficient is a simple Kronecker delta function. The typical impulse response functions for deriving delta, and delta-delta coefficients are shown in Fig. 7. A temporal context of N_t frames at the input of the MLP is created using an FIR filter bank with time shifted Kronecker delta functions as impulse response. As shown in Fig. 6(b), the cosine transform can

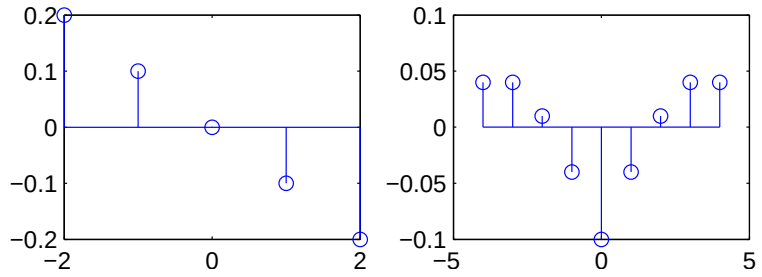


Fig. 7. *Typical impulse response for delta, and delta-delta feature extraction in the standard HTK implementation.*

applied after the filter bank[2] using a block diagonal transformation matrix D_B $[3N_tN_c \times 3N_tN_f]$, obtained by repeating the original DCT matrix $3N_t$ times along the diagonal. The concatenated cepstral feature vector $\mathbf{c}$ $[3N_tN_c \times 1]$ is computed from the filter bank output vector $\mathbf{s}$ $[3N_tN_f \times 1]$ as

$$\mathbf{c} = D_B\,\mathbf{s} \tag{22}$$

[1]The impulse response of a cascade of two LTI systems with impulse responses $h_a(t)$ and $h_b(t)$ is given by the convolution $h_{ab}(t) = h_a(t) * h_b(t)$.
[2]The cosine transform is applied along the frequency and the filter bank is applied along time. Hence the operations can be interchanged.

Furthermore, the block DCT matrix and the mean and variance of the cepstral features are incorporated into the MLP by appropriately modifying the bias at the hidden layer and the weights connecting the input and the hidden layer of the MLP. For this, we denote the mean of the concatenated MFCC cepstral feature vector as μ_c $[3N_tN_c \times 1]$, the diagonal matrix containing the feature variance as Σ_c $[3N_tN_c \times 3N_tN_c]$, and the weight matrix between the input and hidden layer (with M hidden nodes) of the MLP as W $[M \times 3N_tN_c]$. If we denote the bias vector at the hidden layer as h $[M \times 1]$, then the input to the nonlinearity at the hidden layer $\mathbf{s}$ $[M \times 1]$ is given by

$$s = h + W\Sigma^{-\frac{1}{2}}(c - \mu_c) \tag{23}$$

$$= \left(h - W\Sigma^{-\frac{1}{2}}\mu_c\right) + W\Sigma^{-\frac{1}{2}}D_B s$$

$$= h' + W's \tag{24}$$

where,

$$W' = W\Sigma^{-\frac{1}{2}}D_B, \tag{25}$$

$$h' = h - \Delta, \quad and \tag{26}$$

$$\Delta = W\Sigma^{-\frac{1}{2}}\mu_c. \tag{27}$$

It can be seen that the input to the activation function at the hidden layer can be computed either using (23) as shown in Fig. 6(a) or (24) as shown in Fig. 6(c). However, in the latter, the FIR filters are applied directly on the log-energies from the auditory filter bank as required by the Volterra analysis framework shown in Fig. 1. Volterra analysis is applied on the modified MLP parameters which are given by (25) and (26). The mean of the MFCC feature vector is reflected in the correction to the hidden bias $\Delta = [\Delta_1 \ldots \Delta_i \ldots \Delta_M]$, and is given by (27). The correction to the hidden bias Δ_i is used to modify the coefficients of the polynomial expansion of the sigmoid function at the i^{th} hidden node using (18). The modified MLP weights and the coefficients of the polynomial expansion are used to derive the Volterra kernels using (19)-(21).

6. SUMMARY AND CONCLUSION

The main objective of this work was to provide a framework to apply Volterra series to analyze MLP based phoneme posterior probability estimation. We include a part of the feature extraction (LTI system following the auditory analysis) in the analysis framework because the Volterra kernels can be interpreted as spectro-temporal patterns.

In this work, the linear Volterra kernels are interpreted as spectro-temporal patterns. The second order kernels could reveal useful correlations across different frequency channels at different time instants. The spectro-temporal patterns given by the Volterra kernels may not be consistent with the existing acoustic phonetic knowledge of phonemes in all aspects. This is because the Volterra kernels can only reveal the information learned by the MLP to discriminate among phonemes.

Future work includes a detailed analysis into the spectro-temporal properties learned by the system for different feature extraction techniques. Volterra analysis can also be used to compare posterior feature extraction systems that differ in the amount of the training data or the number of hidden nodes in the MLP.

7. ACKNOWLEDGEMENTS

This work was supported by the Swiss national science foundation under the Indo-Swiss joint research program on keyword spotting (KEYSPOT) as well as the Swiss National Center for Competence in Research (NCCR) under the Interactive Multimodal Information Management (IM2) project.

8. REFERENCES

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

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

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

[4] H. Hermansky, D.P.W. Ellis, and S. Sharma, "Tandem Connectionist Feature Extraction for Conventional HMM Systems," *Proc. of ICASSP*, pp. 1635–1638, 2000.

[5] Y.W. Lee and M. Schetzen, "Measurement of Wiener Kernels of a Non-linear System by Cross-correlation," *International Journal of Control*, vol. 2, pp. 237–254, 1965.

[6] D.J. Klein, D.A. Depireux, J.Z. Simon, and S.A Shamma, "Robust Spectrotemporal Reverse Correlation for the Auditory System: Optimizing Stimulus Design," *Journal of Computational Neuroscience*, 2000.

[7] Marmarelis P.Z. and Naka K.-I, "Identification of Multi-Input Biological Systems," *IEEE Trans. Biomedical Engg.*, vol. 21, no. 2, 1974.

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

[9] 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.

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

[11] Y. LeCun, L. Bottou, G.B. Orr, and K.-R Muller, "Effecient BackProp," *Neural Networks: Tricks of the Trade*, 1998.

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

[13] N. Wiener, *Nonlinear Problems in Random Theory*, MIT Press, 1966.