INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Unsupervised Raw Waveform Representation Learning for ASR

Purvi Agrawal, Sriram Ganapathy

Learning and Extraction of Acoustic Patterns (LEAP) Lab, Dept. of Electrical Engg.,
Indian Institute of Science, Bengaluru-560012, India

(purvia, sriramg)@iisc.ac.in

Abstract

In this paper, we propose a deep representation learning approach using the raw speech waveform in an unsupervised learning paradigm. The first layer of the proposed deep model performs acoustic filtering while the subsequent layer performs modulation filtering. The acoustic filterbank is implemented using cosine-modulated Gaussian filters whose parameters are learned. The modulation filtering is performed on log transformed outputs of the first layer and this is achieved using a skip connection based architecture. The outputs from this two layer filtering are fed to the variational autoencoder model. All the model parameters including the filtering layers are learned using the VAE cost function. We employ the learned representations (second layer outputs) in a speech recognition task. Experiments are conducted on Aurora-4 (additive noise with channel artifact) and CHiME-3 (additive noise with reverberation) databases. In these experiments, the learned representations from the proposed framework provide significant improvements in ASR results over the baseline filterbank features and other robust front-ends (average relative improvements of 16% and 6% in word error rate over baseline features on clean and multi-condition training, respectively on Aurora-4 dataset, and 21% over the baseline features on CHiME-3 database).

Index Terms: Unsupervised representation learning, raw speech waveform, convolutional variational autoencoder, cosine-modulated Gaussian filterbank, speech recognition.

1. Introduction

Even though the performance of automatic speech recognition (ASR) systems have improved significantly with the success of deep neural networks (DNN), the performance degradation in mismatched train and test condition is still a challenging task to overcome [1]. It can be partly overcome by obtaining robust speech representation where representations are less susceptible to noise and reverberation. This paper focuses on methods for unsupervised learning of robust speech representation.

Features for speech processing applications are prominently based on properties of human auditory processing. For speech recognition features, traditional approaches like mel filterbank and gammatone filterbank [2, 3] approximate the early part of human hearing. Recently, with the advent of neural networks, feature learning from data has been actively pursued [4–6]. In supervised data-driven approach, the underlying model can automatically discover features needed for the objective at hand from the raw signals, e.g. detection or classification. Several works like [5, 7, 8] have specifically incorporated the learning of acoustic mel-like filters using convolution layer in the initial layers of network. However, these approaches are highly dependent on the amount of labeled training data. Also, many of the prior works use mel initialization. In this paper, we hypothesize that representation learning can be efficiently performed even without labelled data.

A prior work on unsupervised representation learning derives acoustic filterbank using restricted Boltzmann machine (RBM) [9,10]. These works employ a large number of learnable parameters (for eg. 128×80 parameters for 80 filters of 128 tap each with [10] approach). To overcome this, recent efforts have introduced parametric filter learning, like Gaussian filters [11], and Sinc filters [12]. The parametric approaches have an advantage over a standard convolutional layer as the number of free parameters are lesser. However, these works also train the network in supervised manner with the task of phoneme classification for ASR. In this paper, we propose a parametric filter learning approach in an unsupervised manner, which is being attempted for the first time to the best of our knowledge.

This work proposes a deep unsupervised representation learning method directly from raw speech waveform. In particular, the representation learning is carried out as a two-layer process. First, an acoustic filterbank is learnt from the raw waveforms using the first convolutional layer in CVAE. We use cosine-modulated Gaussian functions as acoustic filters with center frequency and bandwidth as the learnable parameters and random initialization as the starting point. The convolution is carried out in time domain, and the output of the layer is pooled and log transformed to obtain time-frequency representation. The next layer learns the spectral and temporal modulation filters from the obtained representations [13]. The filtered spectrogram is then used as feature for ASR. The ASR experiments are performed on Aurora-4 (additive noise with channel artifact) and CHiME-3 challenge (additive noise with reverberation) databases. The proposed approach provides significant improvements in terms of WER over various other noise robust front-ends.

The rest of the paper is organized as follows. Sec. 2.1 describes VAE theory in brief, followed by description of the CVAE model for learning acoustic filterbank, and modulation filter learning in Sec. 2.2 and 2.3, respectively. Sec. 3 describes the ASR experiments with the various front-ends followed by the results. We conclude with a summary in Sec. 5.

2. Filterbank learning using CVAE

2.1. Variational Autoencoder (VAE)

The VAE differs from a standard AE where the VAE model assumes that the samples of latent representation can be drawn from a standard normal distribution, i.e $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ [14]. If we assume an observation vector $\boldsymbol{x}$, a latent vector $\boldsymbol{z}$ and a set of parameters θ for the decoder network, the aim of the VAE network is to maximize the probability of each $\boldsymbol{x}$ in the training set under the generative process. The model involves two steps: (1) $\boldsymbol{z}$ is generated from prior distribution $p(\boldsymbol{z})$; (2) a value $\boldsymbol{x}$ is generated from conditional distribution $p_\theta(\boldsymbol{x}|\boldsymbol{z})$. The VAE framework employs a variational lower bound method [14] in

http://dx.doi.org/10.21437/Interspeech.2019-2652

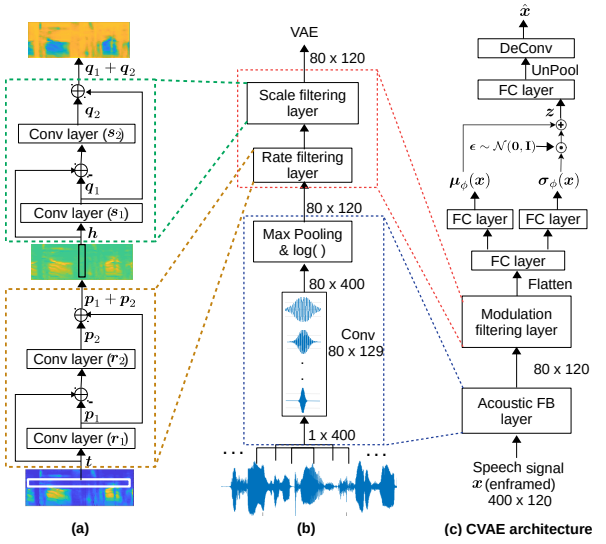

Figure 1: *Block diagram of CVAE architecture in (c) to learn acoustic filters in Acoustic FB layer, and modulation filters in Modulation filtering layer. (a) shows expanded modulation filtering layer, (b) shows expanded acoustic FB layer.*

which a new function $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ (probabilistic encoder with encoder parameters ϕ) is introduced that can take value of $\boldsymbol{x}$ and give a distribution over $\boldsymbol{z}$ values. In other words, the function $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ approximates the true posterior distribution $p_\theta(\boldsymbol{z}|\boldsymbol{x})$. The encoder and decoder parameters, ϕ and θ, respectively, are trained by maximizing the 'variational' lower bound $\mathcal{L}(\theta, \phi; \boldsymbol{x})$ of the marginal likelihood $\log p_\theta(\boldsymbol{x})$, given as

$$\mathcal{L}(\theta, \phi; \boldsymbol{x}) = -D_{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})] + \mathbb{E}_{\boldsymbol{z}|\boldsymbol{x}\sim q_\phi}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] \quad (1)$$

In the implementation of VAE, the distributions $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ and $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ are assumed to be Gaussian. VAE uses a "reparameterization trick" to move the sampling to an input layer. Given the parameters of the encoder network, $\boldsymbol{\mu}_\phi(\boldsymbol{x})$ and $\boldsymbol{\sigma}^2_\phi(\boldsymbol{x})$ which are the mean and variance parameters of $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ - we can sample from $\mathcal{N}(\boldsymbol{\mu}_\phi(\boldsymbol{x}), diag(\boldsymbol{\sigma}^2{}_\phi(\boldsymbol{x})))$ by first sampling $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, then computing $\boldsymbol{z} = \boldsymbol{\mu}_\phi(\boldsymbol{x}) + diag(\boldsymbol{\sigma}_\phi(\boldsymbol{x}))\boldsymbol{\epsilon}$ (shown schematically in Fig. 1(c)).

2.2. Acoustic filterbank learning

This section describes the acoustic filterbank learning using CVAE. The use of CVAE is motivated by the goal of learning filterbank (FB) in an unsupervised manner. The kernels (first layer weights) of the deep CVAE trained using raw input are interpreted as the acoustic filters learned from the data.

The acoustic FB layer is expanded in Fig.1(b). First, we take a frame of length 400 samples from raw waveform (corresponding to 25 ms of signal sampled at 16 kHz), and convolve the raw waveform with the set of 80 filters. The kernels (filters) of the convolution layer are modeled using cosine-modulated Gaussian function as:

$$w_n(t) = \cos 2\pi\mu_n t \times \exp\left(-t^2/2\sigma_n^2\right) \quad (2)$$

where $w_n(t)$ is the n-th filter impulse response at time t, μ_n is the frequency of the nth filter's cosine function, which represents the mean of the corresponding Gaussian function in frequency domain (center frequency), and σ_n is the variance of the n-th filter tied to the mean as $\sigma_n = 1/\mu_n$. The number of filter taps (length of filter impulse response) is fixed to $N = 129$,

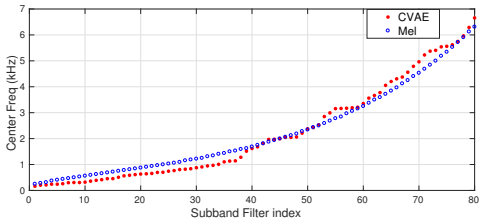

Figure 2: *Comparison of center frequency of filterbank learnt using CVAE with center frequencies of mel filterbank.*

corresponding to 8 ms which has been found to be sufficient to capture temporal variations of speech signal [15]. This approach to FB learning generates filters with a smooth frequency response and allows the filters to be separable. In order to preserve the positive range of frequency values ($0 - 8$kHz) for μ, we choose the μ to be the sigmoid of a real number (λ) which is scaled, i.e. $\mu = 8000\times\text{sigmoid}(\lambda)$ and λ is iteratively updated.

The output of the acoustic FB layer has 80 feature maps, corresponding to convolution of each input frame with each of the 80 filters. Max pooling is applied to the convolved output with pooling kernel of size 8 and pool stride of 4, followed by sigmoid nonlinearity, sum and logarithmic compression, thereby producing 1×80 sized frame level feature vector. We then shift the window (120 times) around the raw waveform by a hop size of 10ms and repeat this convolution to produce patches of size 80×120.

2.3. Modulation filter learning

The second layer of the encoder is the modulation filtering layer as shown in Fig. 1(c). As outlined in its expanded block (Fig. 1(a)), the modulation filtering layer comprises of rate filtering layer followed by scale filtering layer, each of which employs a skip connection based filter learning architecture [13]. The filters of the convolution layers in rate/scale filtering layer trained using spectrogram trajectories (of previous layer output) are interpreted as the modulation filters.

For rate (scale) filtering layer, the inputs are the temporal (spectral) trajectories of the time-frequency representation obtained from the previous layer output (80×120). The dimension of the 1-D trajectory for rate filtering, denoted as $\boldsymbol{t}$, is 1×120 (equivalent to 1.2 s of speech), and for scale filter learning it is 80×1 (corresponding to all 80 frequency bands). The kernel size is 1×5 in each convolution layer. Let the output of its first convolution layer be $\boldsymbol{p}_1$, where $\boldsymbol{p}_1 = \boldsymbol{t} * \boldsymbol{r}_1$ for rate filter learning. In order to learn multiple irredundant filters, we remove the contribution of the learnt kernel from the input $\boldsymbol{t}$ using skip connection and feed the $tanh$ of the residual $(\boldsymbol{t} - \boldsymbol{p}_1)$ to the next convolutional layer. The next layer (also having one kernel) then learns the modulation characteristics from the residual and generates output $\boldsymbol{p}_2 = (\boldsymbol{t}-\boldsymbol{p}_1)*\boldsymbol{r}_2$. We add the two filtered (hidden) representations and apply non-linear activation. This is carried out for all the temporal trajectories of the input time-frequency representation and is reshaped to a time-frequency representation again. Next, the spectral trajectory $\boldsymbol{h}$ of dimension 80×1 is fed to the modulation scale filtering layer, where a similar skip-connection based architecture is employed to learn two scale filters $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$. The resultant output representation is then flattened to be fed to the fully-connected (FC) layer of 120 nodes in the CVAE model (Fig. 1(c)). The latent vector $\boldsymbol{z}$ is calculated from the encoder output as discussed in Section 2.1 and the network is trained with the objective of minimizing total loss function calculated as:

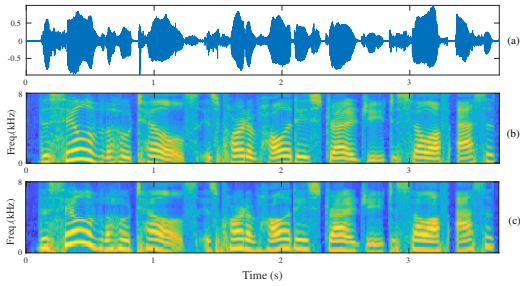

Figure 3: *(a) Speech signal, (b) log mel spectrogram (c) spectrogram using learnt cosine-modulated Gaussian filterbank.*

Table 1: *ASR performance comparison for time-frequency representations with different acoustic filterbanks.*

Cond	MFB	CRBM [10]	CVAE-Acoustic
	Clean Training (Multi condition Training)		
A	3.4 (4.2)	3.4 (3.6)	3.1 (3.8)
B	18.9 (7.8)	23.0 (8.4)	18.7 (7.8)
C	15.3 (8.4)	20.1 (7.2)	14.0 (8.0)
D	35.2 (18.5)	40.0 (19.4)	36.0 (18.8)
Avg.	**24.7** (**12.1**)	28.7 (12.7)	**24.6** (**12.2**)

$$E_{Total} = \alpha E_{MSE} + \beta E_{MSE-acoustic} + \gamma E_{Latent} \quad (3)$$

where E_{MSE} is the mean squared error between input $\boldsymbol{x}$ and the reconstructed output $\hat{\boldsymbol{x}}$, $E_{MSE-acoustic}$ is the mean squared error between the acoustic FB layer output (time-frequency representation) and the reconstructed time-frequency representation in decoder (before deconvolution layer), and E_{Latent} is the latent loss of encoder (KL divergence of $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ with unit Gaussian distribution $p(\boldsymbol{z})$). The values of $\alpha = 0.01$, $\beta = 1.0$ and $\gamma = 0.01$ are used in our experiments. The decoder has one fully connected layer of 120 nodes, followed by an unpooling operation and deconvolution (for reversing the operations performed by the acoustic FB layer of the encoder).

2.4. Filter characteristics

The acoustic filters in the acoustic FB layer and the filters in the modulation filtering layer are iteratively updated using the gradients of the total loss function and Adam optimizer [16]. The CVAE is trained using data of different databases separately. We begin with random initialization of the filters and allow the CVAE to learn filter characteristics from data.

Fig. 2 shows the the center frequency (μ_n values sorted in ascending order) of the acoustic filters using clean Aurora-4 database (details of the Aurora-4 dataset are given in Sec. 3) and this is compared with the center frequency of the mel filterbank. As can be observed, the proposed filterbank also has nonlinear relationship between center frequencies and the filter index with more number of filters in lower frequencies compared to higher frequencies, similar to traditional acoustic filterbanks [2,3]. The time-frequency representation obtained from the learnt filters is shown in Fig. 3(c). The log mel spectrogram is also plotted for reference in Fig. 3(b). It can be observed that the obtained representation preserves all information such as formant contours, voiced and unvoiced sounds, even when filters are learnt with a fully unsupervised objective.

2.5. Feature extraction for ASR

The features for ASR are derived by filtering the raw speech waveforms with learnt acoustic filterbank, followed by filtering the time-frequency representation using modulation filters learnt from the proposed CVAE architecture. In this work, we select the rate filter ($\boldsymbol{r}_2$) with bandpass characteristic as it has been observed earlier to be important for ASR task [13, 17, 18], while both the scale filters are used. Hence, the obtained time-frequency representation from the acoustic FB layer is filtered using filters $(\boldsymbol{r}_2, \boldsymbol{s}_1)$ and $(\boldsymbol{r}_2, \boldsymbol{s}_2)$ separately (80 dimensional each) and are concatenated as features for ASR.

Table 2: *Word error rate (%) in Aurora-4 database for clean training condition with various feature extraction schemes.*

Cond	MFB	PFB	ETS	RAS	LDA	MHE	Prop
A. Clean with same Mic							
Clean	3.4	3.3	3.2	3.5	3.7	3.5	**2.9**
B: Noisy with same Mic							
Airport	21.9	18.3	15.0	19.3	23.2	19.5	14.1
Babble	19.6	16.0	15.5	19.9	21.0	17.7	15.0
Car	8.0	6.2	9.8	7.9	8.7	7.9	6.3
Rest.	24.9	22.9	20.5	23.0	27.0	23.2	18.4
Street	19.5	17.8	19.5	18.7	20.8	18.1	15.3
Train	19.8	16.3	17.4	19.4	20.1	17.9	17.2
Avg.	18.9	16.2	16.3	18.0	20.1	17.4	**14.4**
C: Clean with diff. Mic							
Clean	15.3	**11.7**	14.5	16.0	15.9	14.6	12.9
D: Noisy with diff. Mic							
Airport	40.1	36.4	31.4	39.2	40.4	38.7	32.9
Babble	37.3	34.2	32.1	38.5	36.8	36.8	33.3
Car	24.9	21.5	24.9	24.8	25.9	25.8	21.1
Rest.	39.6	39.0	35.4	39.1	41.0	39.3	34.0
Street	35.7	34.1	35.0	35.8	37.0	35.8	31.9
Train	35.6	31.8	33.2	36.4	36.7	35.9	33.5
Avg.	35.2	32.8	32.0	35.6	36.3	35.4	**31.9**
Avg. of all conditions							
Avg.	24.7	22.1	21.9	24.4	25.6	23.9	**20.6**

3. Experiments and results

The speech recognition Kaldi toolkit [19] is used for building the ASR on two datasets, Aurora-4 and CHiME-3 respectively. A deep belief network- deep neural network (DBN-DNN) with 4 hidden layers having 21 frames of input temporal context and a sigmoid nonlinearity is discriminatively trained using the training data and a tri-gram language model is used in the ASR decoding. For each dataset, we compare the ASR performance of the proposed approach of filtered representation (Prop) with traditional mel filterbank energy (MFB) features, power normalized filterbank energy (PFB) features [20], advanced ETSI front-end (ETS) [21], RASTA features (RAS) [22], LDA based features (LDA) [23], and MHEC features (MHE) [24]. In particular, the RASTA features (RAS) and LDA features are included as they both perform modulation filtering in the temporal domain using a knowledge driven filter and a supervised data driven filter, respectively.

3.1. Aurora-4 ASR

The WSJ Aurora-4 corpus is used for conducting ASR experiments. This database consists of continuous read speech recordings of 5000 words corpus, recorded under clean and noisy conditions (street, train, car, babble, restaurant, and airport) at $10 - 20$ dB SNR. The training data has two sets of 7138 clean and multi condition recordings (84 speakers) respectively. The validation data has two sets of 1206 recordings for clean and multi condition setup. The test data has 330 recordings (8 speakers) for each of the 14 clean and noise conditions. The test data is classified into group A - clean data, B - noisy data, C - clean data with channel distortion, and D - noisy data with channel distortion.

Table 3: *Word error rate (%) in Aurora-4 database for multi-condition training with various feature extraction schemes.*

Cond	MFB	PFB	ETS	RAS	LDA	MHE	Prop
A. Clean with same Mic							
Clean	4.2	4.1	4.5	4.6	4.7	4.0	**3.5**
B: Noisy with same Mic							
Airport	7.5	7.9	8.0	8.1	10.1	8.2	6.5
Babble	7.7	7.9	7.9	8.7	9.9	8.6	7.0
Car	4.7	4.9	5.6	5.0	5.8	4.9	4.4
Rest.	9.8	10.2	11.0	11.0	12.6	11.1	9.1
Street	8.6	8.8	10.0	9.0	10.6	8.8	8.1
Train	8.7	8.3	9.3	9.1	10.6	8.4	8.7
Avg.	7.8	8.0	8.6	8.5	9.9	8.3	**7.3**
C: Clean with diff. Mic							
Clean	8.4	7.8	8.0	9.7	10.0	8.1	**7.4**
D: Noisy with diff. Mic							
Airport	19.7	20.9	18.5	20.1	22.3	20.8	18.2
Babble	20.3	20.9	19.3	20.0	22.5	21.3	18.9
Car	11.8	13.1	14.1	12.5	14.5	12.8	11.3
Rest.	21.7	23.7	21.8	23.1	25.2	23.1	20.6
Street	19.1	20.0	19.4	18.9	21.2	20.5	18.2
Train	18.3	19.6	19.6	19.9	21.6	18.9	17.8
Avg.	18.5	19.7	18.8	19.1	21.2	19.6	**17.5**
Avg. of all conditions							
Avg.	12.1	12.7	12.6	12.8	14.4	12.8	**11.4**

As an initial experiment on Aurora-4 dataset, we compare the ASR performance of the time-frequency representation obtained using different acoustic filterbanks in Table 1. The acoustic FB layer output (CVAE-Acoustic) of the proposed model is compared with MFB and the acoustic FB output learnt in an unsupervised manner from CRBM [10]. It can be observed that CVAE-Acoustic features perform similar to MFB features in both training conditions, under all test conditions and is significantly better than the previous approach to filter learning.

The ASR performance for the proposed (Prop) features (joint acoustic and modulation filtering) in clean and multi-condition training condition is shown in Table 2 and Table 3 for each of the 14 test conditions, respectively. The ASR results are also separately reported for different noisy conditions. As seen in these results, most of the noise robust front-ends do not improve over the baseline mel filterbank (MFB) performance in multi-condition training. The proposed feature extraction scheme provides significant improvements in ASR performance over the baseline system (average relative improvements of 16% over MFB in clean training, and 6% in multi condition training). Furthermore, the improvements in ASR performance are consistently seen across all the noisy test conditions.

3.2. CHiME-3 ASR

The CHiME-3 corpus for ASR contains multi-microphone tablet device recordings from everyday environments, released as a part of 3rd CHiME challenge [25]. Four varied environments are present, cafe (CAF), street junction (STR), public transport (BUS) and pedestrian area (PED). For each environment, two types of noisy speech data are present, real and simulated. The real data consists of 6-channel recordings of sentences from the WSJ0 corpus spoken in the environments listed above. The simulated data was constructed by artificially mixing clean utterances with environment noises. The training data has 1600 (real) noisy recordings and 7138 simulated noisy utterances. We use the beamformed audio for filter learning using CVAE, and for ASR training and testing. The development (dev) and evaluation (eval) data consists of the 410 and 330 utterances respectively. For each set, the sentences are read by four different talkers in the four CHiME-3 environments. This results in 1640 (410×4) and 1320 (330×4) real development and evaluation utterances in total. Identically-sized, simulated dev and eval sets are made by mixing recordings captured in the recording booth with the environmental noise recordings.

Table 4: *Word error rate (%) in CHiME-3 Challenge database for multi-condition training (real+simulated) with test data from simulated and real noisy environments.*

Test Cond	MFB	PFB	RAS	MHE	CVAE-Acoustic	Prop
Sim_dev	14.3	13.7	14.6	14.4	14.2	**12.3**
Real_dev	11.6	12.0	11.8	12.0	11.5	**10.0**
Avg.	13.0	12.9	13.2	13.2	12.8	**11.1**
Sim_eval	25.5	25.1	23.1	26.4	26.1	**19.7**
Real_eval	22.6	23.0	21.6	22.9	22.5	**18.6**
Avg.	24.1	24.1	22.4	24.7	24.3	**19.1**

Table 5: *WER (%) for each noise condition in CHiME-3 dataset with the baseline features and the proposed feature extraction.*

Cond.	Dev Data				Eval Data			
	Sim		Real		Sim		Real	
	MFB	Prop	MFB	Prop	MFB	Prop	MFB	Prop
BUS	12.6	**10.4**	14.2	**11.8**	18.3	**13.6**	29.2	**23.1**
CAF	17.0	**15.6**	11.4	**10.0**	26.3	**21.4**	23.7	**19.1**
PED	12.0	**9.7**	8.5	**7.2**	29.1	**21.5**	21.1	**17.9**
STR	15.7	**13.3**	12.3	**11.1**	28.3	**22.4**	16.4	**14.4**

The results for the CHiME-3 dataset are reported in Table 4. The CVAE-Acoustic features perform similar to MFB features in ASR. However, the proposed (Prop) approach to feature extraction (joint acoustic and modulation filtering) provides significant improvements over the baseline system as well as the other noise robust front-ends considered here. On the average, the proposed approach provides relative improvements of 15% over MFB features in the dev set and 21% in the eval set. The detailed results on different noises in CHiME-3 are reported in Table 5. For all the noise conditions in CHiME-3 in simulated and real environments, the proposed approach shows significant improvements over the baseline MFB features. In the eval dataset, the relative improvements over the baseline features for most of the noise conditions are above 20%.

4. Acknowledgements

This work was partly funded by grants from the Department of Atomic Energy (DAE) project (DAEO0205), the Ministry of Human Resource and Development (MHRD), Government of India, and the Pratiksha Trust, Indian Institute of Science.

5. Summary

The major contribution of the work are as follows:

- Proposed a CVAE architecture with the initial two layers of convolutions for speech representation learning from raw waveform with unsupervised learning objective.
- The first layer of convolutions performs acoustic FB learning which is shown to have nonlinear frequency resolution similar to mel FB. In ASR tasks, the proposed acoustic filters perform similar to mel filters and improve over previous unsupervised FB learning method.
- The second layer performs modulation filtering. The features based on joint acoustic and modulation filtering are used for ASR.
- Significant improvements in multiple datasets over baseline features using representations from the proposed CVAE model.

6. References

[1] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, 2014.

[2] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[3] R. Patterson, I. Nimmo-Smith, J. Holdsworth, and P. Rice, "An efficient auditory filterbank based on the gammatone function," in *a meeting of the IOC Speech Group on Auditory Modelling at RSRE*, vol. 2, no. 7, 1987.

[4] D. Palaz, R. Collobert, and M. M. Doss, "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks," *arXiv preprint : 1304.1018*, 2013.

[5] T. N. Sainath, B. Kingsbury, A. R. Mohamed, and B. Ramabhadran, "Learning filter banks within a deep neural network framework," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2013, pp. 297–302.

[6] Z. Tüske, P. Golik, R. Schlüter, and H. Ney, "Acoustic modeling with deep neural networks using raw time signal for LVCSR," in *Fifteenth annual conference of the international speech communication association*, 2014, pp. 890–894.

[7] Y. Hoshen, R. J. Weiss, and K. W. Wilson, "Speech acoustic modeling from raw multichannel waveforms," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4624–4628.

[8] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform CLDNNs," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015, pp. 1–5.

[9] N. Jaitly and G. Hinton, "Learning a better representation of speech soundwaves using restricted boltzmann machines," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 5884–5887.

[10] H. B. Sailor and H. A. Patil, "Filterbank learning using convolutional restricted boltzmann machine for speech recognition." in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5895–5899.

[11] H. Seki, K. Yamamoto, and S. Nakagawa, "A deep neural network integrated with filterbank learning for speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5480–5484.

[12] M. Ravanelli and Y. Bengio, "Interpretable convolutional filters with SincNet," *arXiv preprint arXiv:1811.09725*, 2018.

[13] P. Agrawal and S. Ganapathy, "Deep variational filter learning models for speech recognition," in *(to appear) IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[14] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[15] M. S. Lewicki, "Efficient coding of natural sounds," *Nature neuroscience*, vol. 5, no. 4, p. 356, 2002.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[17] H. Hermansky and N. Morgan, "Rasta processing of speech," *IEEE transactions on speech and audio processing*, vol. 2, no. 4, pp. 578–589, 1994.

[18] P. Agrawal and S. Ganapathy, "Unsupervised modulation filter learning for noise-robust speech recognition," *Journal of the Acoustical Society of America*, vol. 142, no. 3, pp. 1686–1692, 2017.

[19] D. Povey *et al.*, "The KALDI speech recognition toolkit," in *IEEE ASRU*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[20] C. Kim and R. M. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4101–4104.

[21] E. ETSI, "202 050 v1. 1.1 stq; distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms," *ETSI ES*, vol. 202, no. 050, p. v1, 2002.

[22] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 578–589, 1994.

[23] S. van Vuuren and H. Hermansky, "Data-driven design of RASTA-like filters." in *Eurospeech*, vol. 1, 1997, pp. 1607–1610.

[24] S. O. Sadjadi, T. Hasan, and J. H. Hansen, "Mean Hilbert envelope coefficients (MHEC) for robust speaker recognition," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[25] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015, pp. 504–511.