Interspeech 2018
2-6 September 2018, Hyderabad

Comparison of Unsupervised Modulation Filter Learning Methods for ASR

Purvi Agrawal and Sriram Ganapathy

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

(purvia, sriramg)@iisc.ac.in

Abstract

The widespread deployment of automatic speech recognition (ASR) system in consumer centric applications such as voice interaction and voice search demands the need for noise robustness in such systems. One approach to this problem is to achieve the desired robustness in speech representations used in the ASR. Motivated from studies on robust human speech recognition, we analyse the unsupervised data-driven temporal modulation filter learning for robust feature extraction. In this paper, we compare various unsupervised models for data driven filter learning like convolutional autoencoder (CAE), generative adversarial network (GAN) and convolutional restricted Boltzmann machine (CRBM). The unsupervised models are designed to learn a set of filters from long temporal trajectories of speech sub-band energy. The filters learnt from these models are used for modulation filtering of the input spectrogram before the ASR training. The ASR experiments are performed on Wall Street Journal (WSJ) Aurora-4 database with clean and multi condition training setup. The experimental results obtained from the modulation filtered representations shows considerable robustness to noise, channel distortions and reverberant conditions compared to other feature extraction methods. Among the three approaches compared in this paper, the GAN approach provides the most consistent improvements in ASR accuracy in different training scenarios.

Index Terms: Unsupervised learning, data-driven modulation filtering, generative adversarial network, convolutional autoencoder, robust speech recognition.

1. Introduction

The robustness of speech recognition systems to noise and reverberations continues to be a challenging task inspite of recent advances in its performance. On the other hand, robustness to many of these environmental artifacts is remarkable in the human auditory system. This may be primarily attributed to the spectro-temporal filtering performed by cortical neurons in human auditory system [1, 2, 3].

Several works in the past have incorporated the knowledge of spectro-temporal modulation filtering for ASR. These approaches define a series of the spectral modulation (scale), temporal modulation (rate), and spectro-temporal modulation filtering operations on the speech spectrogram. For the ASR application, the use of temporal modulations such as RASTA filtering [4], TRAPS [5] and HATS [6] have been well studied. The Gabor filtering approach attempts to filter the spectro-temporal modulations jointly [7, 8]. An approach to separable spectro-temporal Gabor filter bank features is proposed in [9]. The temporal modulation filtering using the linear discriminant analysis (LDA) is a supervised data driven approach [10]. Recently, we have also analysed unsupervised rate-scale filtering using a convolutional restricted Boltzmann machine (CRBM) [11].

This work analyses and compares various unsupervised data-driven modulation filtering approaches. The framework of unsupervised learning can be divided into distribution learning, representation learning or clustering methods. An autoencoder (AE) is a neural network which aims at representation learning at the hidden layers by mapping the input to the output using mean square error cost [12]. A convolutional autoencoder (CAE) incorporates convolutional layers in an AE [13, 14]. In this work, we explore the use of CAE for temporal modulation filter learning. A second approach for representation learning using conditional generative adversarial network (cGAN) attempts to modify the CAE approach with an additional adversarial cost function [15]. Here, a second network is trained in parallel to access and give feedback to CAE about how good its input to output mapping is, by analysing the input and generated outputs as pair. We explore the use of cGAN for modulation filter learning in this work.

We also use a distribution learning method for unsupervised modeling with the convolutional restricted Boltzmann machine (CRBM). The CRBM learns a binary hidden layer by maximizing likelihood of Boltzmann distribution [16, 17]. All the three models provide 1-D temporal modulation filter learning schemes which can be used in ASR. The kernel of the first convolutional layer of these models are interpreted as modulation filter, that captures modulations derived from large amount of unsupervised speech spectrogram data. After learning a filter, the projection of the input spectrogram on the learnt filter is removed and the residual spectrogram is then used in the same model framework for learning subsequent filters. In this approach, we do not apply any prior knowledge of the perceptual modulation filtering studies in auditory processing and allow the data to learn the temporal modulation content present in it.

The ASR experiments are performed on the Wall street Journal (WSJ) Aurora-4 database using a deep neural network (DNN) acoustic model. The results from the experiments indicate that the features derived from the learnt filters provide significant improvements over other noise robust front-ends. We also compare the ASR performance of the three filter learning models with respect to the derived filtered features. Further, we investigate the performance of the filtered features in reverberant conditions and in a semi-supervised setting where availability of labeled data is limited.

The rest of the paper is organized as follows. Sec. 2 describes the data driven models for learning temporal modulation filters, followed by multiple filter learning criteria. Sec. 3 describes the ASR experiments with the various front-ends followed by the results. We conclude with a summary in Sec. 4.

2. Modulation filter learning

This section describes three generative models used to capture the temporal modulation characteristics from spectrogram data. All the models use temporal trajectories of 1.5 s length derived

10.21437/Interspeech.2018-1972

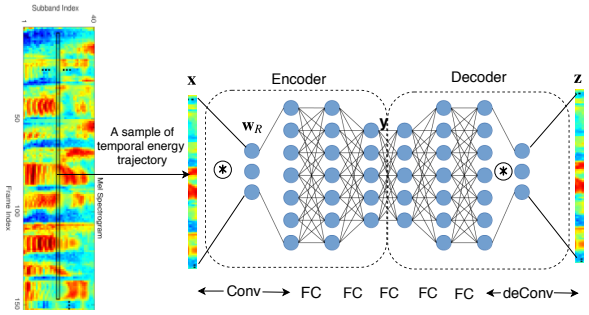

Figure 1: *Block diagram of temporal modulation filter learning using CAE from spectrograms.*

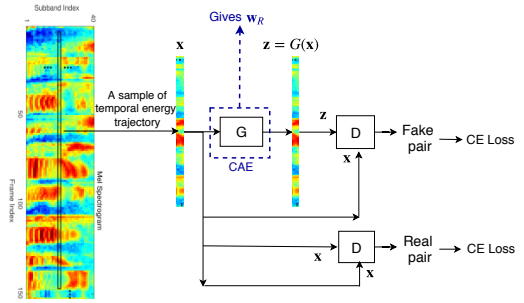

Figure 2: *Block diagram of temporal modulation filter learning using GAN - training G in an adversarial framework.*

from speech spectrograms (25 ms frame length with shift of 10 ms containing 80 mel-bands).

2.1. Convolutional Autoencoder (CAE)

Autoencoder (AE) is a neural network that can convert high-dimensional data to low-dimensional codes using encoder, and a decoder block to reconstruct the data back [12]. The encoder performs the deterministic mapping $f(\theta)$ from a n-dimensional input vector $\mathbf{x}$ into a hidden (encoded) representation $\mathbf{y}$ as:

$$f_\theta(x) = \mathbf{y} = \mathbf{s}(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{1}$$

with parameters $\theta = \{\mathbf{W}, \mathbf{b}\}$, where $\mathbf{W}$ is a weight matrix, $\mathbf{b}$ is a bias vector and s is the nonlinearity. The resulting encoded representation $\mathbf{y}$ is then mapped back (decoded) to a reconstructed d-dimensional vector $\mathbf{z}$ in the input space as:

$$g_{\theta'}(\mathbf{y}) = \mathbf{z} = \mathbf{s}(\mathbf{W}'\mathbf{y} + \mathbf{b}') \tag{2}$$

with $\theta' = \{\mathbf{W}', \mathbf{b}'\}$. Here, $\mathbf{z}$ is interpreted in probabilistic terms as the parameters of a distribution $p(X|Z = \mathbf{z})$ that may generate $\mathbf{x}$ with high probability. This requires the minimization of reconstruction error with respect to loss $L(\mathbf{x}, \mathbf{z}) = -\log \mathbf{p}(\mathbf{x}|\mathbf{z})$. For real-valued $\mathbf{x}$, this leads to

$$L(\mathbf{x}, \mathbf{z}) \approx ||\mathbf{x} - \mathbf{z}||^2. \tag{3}$$

While fully connected layers learn to reconstruct and identify the features of each signal as a whole, convolutional neural networks (CNNs) learn the mapping to the targets using feature maps locally [18, 13]. The CNNs require supervised training data and typically operate on smaller contextual windows (11 frames). A Convolutional autoencoder (CAE) replaces the fully connected layer with the convolutional layer that is able to learn local patterns by shared weights of connections [13, 14]. In our work, CAE is used to capture the temporal modulations of the speech spectrogram data. The architecture of CAE used is shown in Fig. 1. In order to analyse the effect of filtering, only one convolutional layer in encoder (Conv) and one convolutional layer in decoder (deConv) is used. The number of kernels in first Conv layer and last deConv layer is also restricted to one. The kernel (filter) to be learnt is marked $\mathbf{w_R}$ in the Fig. 1. The other layers are fully connected (FC) layers. The network is trained to reduce the mean square error (MSE) of the temporal trajectories till the performance saturates. To learn multiple non-overlapping filters (corresponding to different modulation characteristics), a sequential filter learning criteria is followed as explained in Section 2.4 [11].

2.2. Generative Adversarial Network (GAN)

In our work, CAE is also trained in an adversarial manner called as generative adversarial network (GAN). The GANs are unsupervised generative models that learn to produce realistic samples of input data via an adversarial learning. It consists of two models (usually neural networks) that are trained simultaneously: a generator G that captures the data distribution, and a discriminator D that estimates the probability that a sample came from the training data rather than G [15]. The training procedure for G is to maximize the probability of D making a mistake. The generator $G(n; \theta_G)$ is learned by mapping noise $\mathbf{n}$ to data space $\mathbf{x}$, where G is a differentiable function represented by a multilayer perceptron with parameters θ_G.

The discriminator $D(\mathbf{x}; \theta_D)$ is a second network that outputs a scalar $D(\mathbf{x})$ representing the probability that $\mathbf{x}$ is a true data point and not a model generated sample. We train D to maximize the probability of assigning the correct label to both training examples and samples from G. We simultaneously train G to minimize $\log(1 - D(G(\mathbf{n})))$. In other words, D and G play the following two-player minimax game with value function $V(G, D)$:

$$\min_G \max_D V(G, D) = E_{\mathbf{x}}[\log D(\mathbf{x})] + E_{\mathbf{n}}[\log(1 - D(G(\mathbf{n})))].$$

In contrast, conditional GANs (cGAN) [19] learn a mapping from observed sample $\mathbf{x}$ and random noise vector $\mathbf{n}$, to $\mathbf{z}$, $G : \{\mathbf{x}, \mathbf{n}\} \rightarrow \mathbf{z}$. In particular, the D observes both real and generated samples as a pair, with the task of detecting whether it is real pair or a fake pair. The objective of a cGAN can be expressed as:

$$\min_G \max_D V(D, G) = E_{\mathbf{x},\mathbf{z}}[\log D(\mathbf{x}, \mathbf{z})] + E_{\mathbf{x},\mathbf{n}}[\log(1 - D(\mathbf{x}, G(\mathbf{x}, \mathbf{n})))].$$

This paper uses cGAN to learn modulation characteristics of temporal spectrogram trajectories with aim of reconstructing $\mathbf{z} = \mathbf{x}$, i.e. identity mapping with G. The block diagram for filter learning with GAN is shown in Fig. 2. The CAE described in previous section is used as G with the same architecture and D is a classifier with two classes as real and fake pair. The D observes the pair of generated trajectory ($\mathbf{z}$) and the actual trajectory ($\mathbf{x}$) as fake pair, while the pair of actual trajectory ($\mathbf{x}$ and $\mathbf{x}$) as the real pair. The network is trained to minimize the cross-entropy (CE) loss of D and the MSE loss of G, and the learnt kernel $\mathbf{w_R}$ of G is interpreted as the modulation filter.

2.3. Convolutional Restricted Boltzmann Machine

Another generative model that tries to learn the input data distribution by maximizing the data likelihood is the restricted Boltzmann machine (RBM) [20]. A RBM is a two-layer, undirected graphical model with a set of binary hidden units $\mathbf{h}$ (as output layer), a set of (binary or real-valued) visible units $\mathbf{v}$ (as

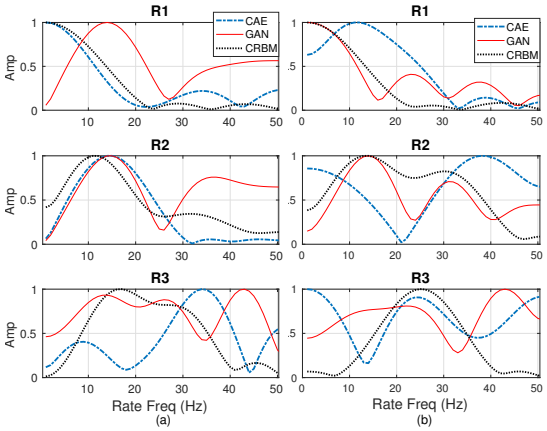

Figure 3: *1-D temporal modulation (rate) filters learnt from (a) clean WSJ mel spectrogram (b) multi condition WSJ mel spectrogram with residual approach.*

input layer), and symmetric connections between these two layers represented by a weight matrix $\mathbf{W}$. The energy function of the Gaussian RBM is given as:

$$E(\mathbf{v}, \mathbf{h}, \theta) = -\sum_{i,j} \mathbf{v}_i \mathbf{W}_{ij} \mathbf{h}_j - \sum_i {\mathbf{v}_i}^2 - \sum_j \mathbf{c}_j \mathbf{h}_j \quad (4)$$

where i and j are indices that iterate over visible and hidden units, respectively, model parameters are $\theta = (\mathbf{W}, \mathbf{b}, \mathbf{c})$, with **b** and **c** being the bias at visible and hidden layer, respectively. We use the contrastive divergence (CD) learning algorithm for RBM training [16] using gradient ascent based optimization procedure. With regard to visible-hidden weights, the one-step CD (Gibbs sampler) followed by weight update is given as:

$$\triangle_{\mathbf{W}_{ij}} J(\mathbf{W}, \mathbf{b}, \mathbf{c}; \mathbf{v}) = < \mathbf{v}_i \mathbf{h}_j >_{data} - < \mathbf{v}_i \mathbf{h}_j >_{model},$$
$$\mathbf{W}' = \mathbf{W} + \eta(\triangle_{\mathbf{W}} J), \quad (5)$$

where J is the log likelihood defined as the exponential of negative of E_1, $< . >$ denotes the expectation under the distribution specified by the subscript, $\mathbf{v}_i$ and $\mathbf{h}_j$ are the ith and jth elements of visible and hidden layer, respectively, $\mathbf{W}'$ is the updated $\mathbf{W}$ matrix, and η is the learning rate.

A convolutional operation can be added to RBM by weight sharing, reconstructing and identifying the features of the signal locally [17, 21, 22]. In this work, the temporal trajectories of mel spectrogram are given as input to CRBM. The details of filter learning using CRBM is discussed in [11].

2.4. Multiple filter learning

For learning multiple filters that are less redundant [23], we use the following approach. After an initial filter is learnt (R1), we remove the contribution of learnt rate component (R1) from the original spectrogram by subtracting the original spectrogram from the rate filtered spectrogram. This residual is fed back to generative model for learning next filter (R2). This method, similar to matching pursuit (MP) algorithm [24], allows us to learn irredundant set of filters. Hence, the learnt temporal filters serve as a dictionary to decompose the input data (temporal trajectories of sub-band energy from spectrogram in our case). We begin with random initialization of weights and allow the different unsupervised models to learn modulation characteristics from data. The normalized magnitude response of the filters learnt from the three filter learning schemes discussed in Sec 2 is shown in Figure 3 in clean and multi condition training setup.

In our analysis, we find that the first filter learnt from the input mel spectrogram is invariably a low-pass in CAE and CRBM, while the first filter from GAN model has a bandpass characteristic in clean condition (Figure 3 (a) and 3 (b)). As seen here, deriving the filters using MP style algorithm provides irredundant filters. In the case of multi condition filter learning, we assume that a filter will learn common underlying representation of all types of input noises. The features for ASR are derived by filtering the log mel spectrogram using filters learnt from unsupervised models. The features are mean-variance normalized at utterance level before DNN training.

Table 1: *Comparison of WER (%) in Aurora-4 database for each filter of corresponding model.*

Model	R1	R2	R3
CRBM	27.7	**23.0**	23.1
CAE	27.4	**20.7**	21.9
GAN	**20.3**	20.7	22.9

3. Experiments and results

3.1. Noisy Speech Recognition

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). 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.

The speech recognition Kaldi toolkit [25] is used for building the ASR. 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. We compare the ASR performance of the discussed unsupervised modulation filtering approaches with traditional mel filter bank energy (MF) features, power normalized filter bank energy (PN) features [26], advanced ETSI front-end (ET) [27] and RASTA features (RA) [4].

We trained the ASR in clean condition for each of the learnt filter R1, R2, R3 individually for all 3 models, and observed that the filter with bandpass characteristic gives the best performance amongst the three. From the average word error rate (WER) reported in Table 1, the CRBM and CAE gives the best performance with R2, while GAN having bandpass characteristic in R1 provides the best performance over the other two.

The ASR performance in clean training condition with the best filters is reported in Table 2 for each of the 14 test conditions. From the table, it can be observed that PN and ET features provide better performance compared to the MF and RAS features. The data driven modulation filtering approach on mel spectrogram provides significant improvement in noisy and channel distortion scenarios. The GAN features also gave superior performance compared to CAE and CRBM features (average relative improvements of 18 % by GAN - R1 over MF).

In multi condition training scenario, a similar trend was observed for ASR with respect to bandpass characteristic of the learnt filter. The CRBM and GAN gave the best perfromance with R2, while CAE having bandpass characteristic in R1 provides the best performance over the other two. The comparison of the best filters in multi condition are reported in Table 3. The

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

Cond	MF	PN	ET	RA	CRBM R2	CAE R2	GAN R1
A. Clean with same Mic							
Clean	3.4	3.3	3.2	3.5	**2.7**	3.0	3.2
B: Noisy with same Mic							
Airport	21.9	18.3	15.0	19.3	18.0	13.7	13.3
Babble	19.6	16.0	15.5	19.9	17.4	14.0	13.7
Car	8.0	6.2	9.8	7.9	6.6	6.0	6.1
Rest.	24.9	22.9	20.5	23.0	22.4	18.0	17.7
Street	19.5	17.8	19.5	18.7	17.6	15.8	14.9
Train	19.8	16.3	17.4	19.4	18.1	16.8	16.7
Avg.	18.9	16.2	16.3	18.0	16.7	14.0	**13.8**
C: Clean with diff. Mic							
Clean	15.3	**11.7**	14.5	16.0	13.9	13.1	13.6
D: Noisy with diff. Mic							
Airport	40.1	36.4	31.4	39.2	37.1	34.2	32.6
Babble	37.3	34.2	32.1	38.5	35.0	33.4	32.3
Car	24.9	21.5	24.9	24.8	24.1	22.5	21.3
Rest.	39.6	39.0	35.4	39.1	37.7	35.6	34.2
Street	35.7	34.1	35.0	35.8	35.2	34.4	32.1
Train	35.6	31.8	33.2	36.4	35.6	33.7	32.3
Avg.	35.2	32.8	32.0	35.6	34.1	31.6	**30.8**
Avg. of all conditions							
Avg.	24.7	22.1	21.9	24.4	23.0	20.7	**20.3**

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

Cond	MF	PN	ET	RA	CRBM R2	CAE R1	GAN R2
A. Clean with same Mic							
Clean	4.2	4.1	4.5	4.6	**3.4**	3.8	3.5
B: Noisy with same Mic							
Airport	7.5	7.9	8.0	8.1	6.9	7.3	6.7
Babble	7.7	7.9	7.9	8.7	7.0	7.7	7.1
Car	4.7	4.9	5.6	5.0	4.1	4.4	4.1
Rest.	9.8	10.2	11.0	11.0	9.1	9.1	8.6
Street	8.6	8.8	10.0	9.0	8.1	8.7	8.3
Train	8.7	8.3	9.3	9.1	8.1	8.6	8.5
Avg.	7.8	8.0	8.6	8.5	**7.2**	7.6	**7.2**
C: Clean with diff. Mic							
Clean	8.4	7.8	8.0	9.7	**7.1**	8.0	7.5
D: Noisy with diff. Mic							
Airport	19.7	20.9	18.5	20.1	18.1	19.2	17.9
Babble	20.3	20.9	19.3	20.0	18.0	19.5	19.0
Car	11.8	13.1	14.1	12.5	10.2	11.5	10.7
Rest.	21.7	23.7	21.8	23.1	19.6	21.9	20.5
Street	19.1	20.0	19.4	18.9	17.8	19.5	17.9
Train	18.3	19.6	19.6	19.9	17.1	18.8	18.4
Avg.	18.5	19.7	18.8	19.1	**16.8**	18.4	17.4
Avg. of all conditions							
Avg.	12.1	12.7	12.6	12.8	**11.0**	12.0	11.3

filtered features improves the performance of ASR compared to the baseline features. Here, the CRBM provide the best performance which is found to be moderately better than GAN (average relative improvements of 9% with CRBM-R2 and 7% with GAN-R2 over MF).

3.2. Reverberant speech recognition

The ASR experiments on reverberant speech data are performed using WSJCAM0 corpus in a single channel scenario, released as a part of REVERB challenge [28]. This database consists of 7861 recordings from 92 training speakers, 1488 recordings from 20 development test (dt) speakers and 2178 recordings from two sets of 14 evaluation test (et) speakers, with each speaker providing about 90 utterances. These recordings were carried out with two sets of microphone- head mounted as well as desk microphone positioned about half meter from the speaker's head. The database consists of three subsets: training data set (Train) - for both clean and multi condition training using simulated reverb data, a simulated test dataset (Sim) and a naturally reverberant recording of the test dataset (Real). The rate filters with GAN are learnt from mel spectrogram of Train dataset - separately for both clean and multi condition. Table 4 shows the ASR performance for clean and multi-condition training conditions using MF, PN and the selected modulation filtering GAN-R1 (clean) and GAN-R2 (multi condition).

It can be observed that the selected features perform better than MF and PN under almost all test conditions with clean and reverb training data. For the clean training, there is an average relative improvement of 23 % over MF features on Sim test data and about 5 % with Real test data. For the multi condition reverb training, there is improvement under all test conditions with average relative improvement of 6 % over MF features on Sim test data and about 3 % for Real test data.

3.3. Semi-supervised training

This section is motivated for the case when only a fraction of the available training data is labelled. In our case, the semi-supervised learning is done using the full unsupervised data for filter learning, while the supervised ASR model is trained with

Table 4: *Word error rate (%) in REVERB Challenge database for clean and multi condition training.*

Cond.	MF	PF	GAN	MF	PF	GAN
	Clean training			Multi training		
Sim_dt	37.2	36.3	**28.9**	11.9	**11.3**	11.4
Sim_et	35.8	35.2	**27.4**	12.2	11.5	**11.4**
Real_dt	70	73.3	**67.1**	25.9	25.7	**24.8**
Real_et	73.1	77	**69.1**	30.9	30.7	**30.1**

Table 5: *Word error rate (%) in Aurora-4 database using lesser amount of labeled training data (70 %, 50 %, 30 %).*

Training data	100 %		70 %		50 %		30 %	
	MF	GAN	MF	GAN	MF	GAN	MF	GAN
Clean	24.6	**20.3**	26.3	**20.8**	29.3	**21.4**	33.8	**22.9**
Multi cond.	12.1	**11.3**	15.8	**13.4**	17.6	**14.5**	21.0	**16.4**

reduced labeled data (70, 50 and 30 %). The performance comparison of ASR with semi-supervised training is shown in Table 5 for MF and the selected (GAN-R1 for clean, GAN-R2 for multi) feature scheme (average WER of all 14 test data conditions). These results indicate that the selected filtered features are more resilient to reduced amounts of labeled training data as compared to the baseline system. These features perform significantly better than MF features (average relative improvement of 32 % in clean training and 22 % in multi condition training with the use of 30 % labeled data).

4. Summary

This work compares the three unsupervised models for modulation filter learning. The model architectures are designed to learn and interpret kernel as modulation filter. This work also develops a filter learning model using generator in an adversarial paradigm. From the ASR results, the bandpass temporal modulation region proved to be useful for noise robustness, and the models are able to capture these regions. The proposed approach also gives considerable improvements in ASR performance under reverberant environments. Further, the modulation filtered features using GAN prove to be resilient in semi-supervised training scenario.

5. References

[1] N. Mesgarani and S. Shamma, "Speech processing with a cortical representation of audio," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 5872–5875.

[2] T. M. Elliott and F. E. Theunissen, "The modulation transfer function for speech intelligibility," *PLoS comput biol*, vol. 5, no. 3, p. e1000302, 2009.

[3] N. Mesgarani, M. Slaney, and S. A. Shamma, "Discrimination of speech from nonspeech based on multiscale spectro-temporal modulations," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 3, pp. 920–930, 2006.

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

[5] H. Hermansky and S. Sharma, "Temporal patterns (TRAPS) in asr of noisy speech," in *International Conference on Acoustics, Speech, and Signal Processing, Proceedings.*, vol. 1. IEEE, 1999, pp. 289–292.

[6] B. Chen, Q. Zhu, and N. Morgan, "Long-term temporal features for conversational speech recognition," in *International Workshop on Machine Learning for Multimodal Interaction*. Springer, 2004, pp. 232–242.

[7] M. Kleinschmidt, "Localized spectro-temporal features for automatic speech recognition." in *Proc. of Eurospeech*, 2003, pp. 2573–2576.

[8] X. Domont, M. Heckmann, F. Joublin, and C. Goerick, "Hierarchical spectro-temporal features for robust speech recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2008, pp. 4417–4420.

[9] M. R. Schädler and B. Kollmeier, "Separable spectro-temporal gabor filter bank features: Reducing the complexity of robust features for automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 137, no. 4, pp. 2047–2059, 2015.

[10] J.-W. Hung and L.-S. Lee, "Optimization of temporal filters for constructing robust features in speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 3, pp. 808–832, 2006.

[11] 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.

[12] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[13] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[14] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *International Conference on Artificial Neural Networks*. Springer, 2011, pp. 52–59.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[16] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[17] M. Norouzi, "Convolutional restricted boltzmann machines for feature learning," Ph.D. dissertation, School of Computing Science-Simon Fraser University, 2009.

[18] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[19] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CVPR*, 2017.

[20] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *Proceedings of the 24th International Conference on Machine Learning*. ACM, 2007, pp. 791–798.

[21] M. Norouzi, M. Ranjbar, and G. Mori, "Stacks of convolutional restricted boltzmann machines for shift-invariant feature learning," in *Conference on Computer Vision and Pattern Recognition, 2009. (CVPR) 2009.* IEEE, 2009, pp. 2735–2742.

[22] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 609–616.

[23] S. K. Nemala, K. Patil, and M. Elhilali, "A multistream feature framework based on bandpass modulation filtering for robust speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 2, pp. 416–426, 2013.

[24] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[25] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011.

[26] 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)*. IEEE, 2012, pp. 4101–4104.

[27] 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.

[28] K. Kinoshita, M. Delcroix, S. Gannot, E. A. Habets, R. Haeb-Umbach, W. Kellermann, V. Leutnant, R. Maas, T. Nakatani, B. Raj *et al.*, "A summary of the REVERB challenge: state-of-the-art and remaining challenges in reverberant speech processing research," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, pp. 1–19, 2016.