A Reduced Complexity MFCC-based Deep Neural Network Approach for Speech Enhancement

Ryan Razani, Hanwook Chung, Yazid Attabi and Benoit Champagne
Department of Electrical and Computer Engineering,
McGill University, 3480 University St., Montreal, Canada
Email: {ryan.razani, hanwook.chung, yazid.attabi}@mail.mcgill.ca
Email: benoit.champagne@mcgill.ca

***Abstract*—This paper focuses on a regression-based deep neural network (DNN) approach for single-channel speech enhancement. While DNN can lead to improved speech quality compared to classical approaches, it is afflicted by high computational complexity in the training stage. The main contribution of this work is to reduce the DNN complexity by introducing a spectral feature mapping from noisy mel frequency cepstral coefficients (MFCC) to enhanced short-time Fourier transform (STFT) spectrum. This approach requires much fewer input features and consequently lead to reduced DNN complexity. Exploiting the frequency domain speech features obtained from this mapping also avoids the information loss in reconstructing the speech signal back to time domain from its MFCC. Compared to the STFT-based DNN approach, the complexity of our approach for the training phase is reduced by a factor of 4.75. Moreover, experimental results of perceptual evaluation of speech quality (PESQ) and source-to-distortion ratio (SDR) show that the proposed approach outperforms the benchmark algorithms and this for various noise types, and different SNR levels.**

***Index Terms*—Speech enhancement, deep learning, neural networks, low-complexity, MFCC**

I. Introduction

The purpose of speech enhancement is to improve the perceived quality or intelligibility of speech signals that have been degraded due to different types of acoustic background noise and interference. Speech enhancement is used in various applications such as hearing aids, cellular phones, multiparty conferencing, robust speech/speaker recognition, security monitoring and intelligence.

Several single channel speech enhancement techniques have been proposed during the past decades, including spectral subtraction [1], [2], Wiener filtering [3], [4], minimum mean square error short-time spectral amplitude estimation (MMSE-STSA) [5], [6], Kalman filtering [7], [8], subspace methods [9], [10]. These techniques rely on a simplified signal model where the background noise is assumed to be additive with statistical characteristics that change slowly over time [11]. While such modeling leads to tractable signal processing operations, the enhancement performance of these traditional methods suffers from limited noise reduction, musical noise, and non-linear distortion.

Recently, there has been much interest towards the application of machine learning techniques to the speech enhancement problem, including non-negative matrix factorization (NMF) [12], [13] and deep neural network (DNN) [14], [15]. Early work on using shallow neural networks (SNN) as non-linear filters in speech enhancement has been presented in [16]. Yet, the performance of the SNN model with limited network size and small training set is not satisfactory. With the advancement of machine learning algorithms and improvement in digital hardware performance, the DNN structure has been drawing considerable attention lately within the research community, as it can achieve significantly better performance compared to SNN, at the cost of increased computational complexity. DNN with multiple hidden layers are now preferred for many applications as they can more efficiently learn statistical information [14].

In recent works on speech enhancement, DNN-based models were presented that employ multi-condition training procedures to initialize the network parameters, such as restricted Boltzmann machine (RBM) [17] and deep denoising autoencoder (DAE) [18]. However, the use of these pre-training approaches is computationally expensive and does not seem to notably affect the final enhancement performance of the DNN with ReLU activation function, given sufficiently large and varied training data sets [19]. In [15], Liu *et al.* presented a simpler speech enhancement approach using DNN with no pre-training, which can achieve better performance when compared to NMF techniques with comparable complexity. In [20], a DNN-based speech separation technique was proposed using time-frequency masking, as obtained from a second DNN. The singular value decomposition (SVD) reduction techniques with DNN training for noisy reverberant speech recognition was investigated in [21]. The NMF-based target speech enhancement using DNN was proposed in [22]. In [23], a signal pre-processing front-end based on DNN was presented to enhance the speech signal for robust speech recognition; however, the learning-based noise model was not considered. In addition, the deep recurrent neural network (DRNN) [24] was introduced to exploit temporal information in the source separation problem. Although DRNN is capable of modeling sequential data for speech processing tasks, its performance is weak when trained on

978-1-5386-4662-5/17/$31.00 ©2017 IEEE

limited noise types [25]. Subsequently, the long short-term memory (LSTM) [26], [27] model was used to tackle the gradient vanishing and exploding problem with DRNN and to learn long-term dependencies. While the use of LSTM with DRNN leads to improved performance, it requires increased complexity in implementation.

This paper aims to overcome these limitations by introducing a low-complexity DNN for nonlinear regression-based feed-forward DNN model presented in [15]. While the latter framework can lead to improved speech quality compared to classical approaches, it is afflicted by high computational complexity in the training stage. The main contribution of this work is to reduce the DNN complexity by introducing a spectral feature mapping from noisy mel frequency cepstral coefficients (MFCC) to enhanced STFT spectrum. Compared to the STFT-based DNN and NMF approaches, our model reduces the processing complexity for the training phase by a factor of 4.75. Results of the PESQ and SDR show that the proposed method outperforms the benchmark algorithms for various noise types, and different SNR levels.

II. Neural Networks for Speech Denoising

In this section, we review the basic features of a STFT-based DNN structure for speech enhancement and associated training procedure. In single-channel speech enhancement, the noisy speech spectrum, obtained via STFT, can be expressed as,

$$Y(\nu,k) = X(\nu,k) + D(\nu,k) \tag{1}$$

where $Y(\nu,k)$, $X(\nu,k)$ and $D(\nu,k)$ refer to the STFT coefficients of the noisy speech, clean speech and noise at the (ν,k)-th time-frequency bin, respectively. Here, $\nu \in \{1,2,\ldots,N\}$ and $k \in \{0,1,\ldots,K-1\}$, where N is the total number of frames and K is the STFT dimension.

A. DNN Structure

The architecture adopted for our model is based on a feed-forward DNN consisting of multiple non-linear hidden layers. This architecture, shown in Fig. 1, allows to represent a highly non-linear regression function, which maps noisy speech features at the input into clean speech features at the output. Each hidden layer, labeled with index $l \in \{1,2,\ldots,L-1\}$, where L is the total number of layers, consists of I_l neurons. The output values of the l-th layer are represented by vector $\mathbf{h}^{(l)} \in \mathbb{R}^{I_l}$ and are expressed as,

$$\mathbf{h}^{(l)} = f\left(\mathbf{W}^{(l)}\,\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right) \tag{2}$$

where $\mathbf{W}^{(l)} \in \mathbb{R}^{I_l \times I_{l-1}}$ is a linear transformation matrix with (i,j)-th entry $w_{ij}^{(l)}$, $\mathbf{b}^{(l)} \in \mathbb{R}^{I_l}$ is a bias vector with i-th entry $b_i^{(l)}$, and $f(.)$ represents a non-linear activation function which operates element-wise. Depending on the application, the activation function can be selected accordingly, such as a sigmoid or piecewise linear function

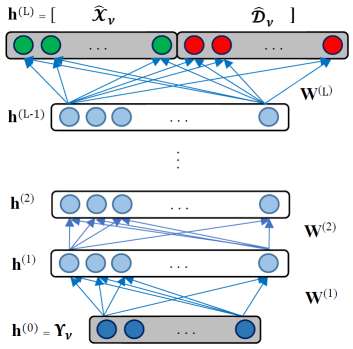

Fig. 1. Feed forward DNN

[28]. However, the rectified linear unit function [19], turns out to be more effective in our prediction problem. In the DNN architecture of Fig. 1, the input (bottom) layer consists of the noisy spectrum magnitudes at the ν-th frame. Specifically $\mathbf{h}^{(0)} = \boldsymbol{\mathcal{Y}}_\nu \equiv [\mathcal{Y}_{\nu,0}, \mathcal{Y}_{\nu,1}, \ldots, \mathcal{Y}_{\nu,I_0-1}]^T$ where $\mathcal{Y}_{\nu,k} = |Y(\nu,k)|$, $I_0 = K'$ and $K' = K/2+1$. The output (top) layer in Fig. 1, represented by vector $\mathbf{h}^{(L)} \in \mathbb{R}^{I_L}$, is obtained through a linear regression as,

$$\mathbf{h}^{(L)} = \mathbf{W}^{(L)}\mathbf{h}^{(L-1)} + \mathbf{b}^{(L)} \tag{3}$$

where $\mathbf{W}^{(L)} \in \mathbb{R}^{I_L \times I_{L-1}}$ and $\mathbf{b}^{(L)} \in \mathbb{R}^{I_L}$. For the output layer, we adopt a special configuration where $I_L = 2K'$ and $\mathbf{h}^{(L)} = [\hat{\boldsymbol{\mathcal{X}}}_\nu, \hat{\boldsymbol{\mathcal{D}}}_\nu]$ consists of two K'-dimensional prediction vectors. In this notation, $\hat{\boldsymbol{\mathcal{X}}}_\nu = [\hat{\mathcal{X}}_{\nu,0}, \hat{\mathcal{X}}_{\nu,1}, \ldots, \hat{\mathcal{X}}_{\nu,K'-1}]^T$ and $\hat{\boldsymbol{\mathcal{D}}}_\nu = [\hat{\mathcal{D}}_{\nu,0}, \hat{\mathcal{D}}_{\nu,1}, \ldots, \hat{\mathcal{D}}_{\nu,K'-1}]^T$. The components $\hat{\mathcal{X}}_{\nu,k}$ and $\hat{\mathcal{D}}_{\nu,k}$ provide preliminary estimates of the clean speech and noise spectrum magnitude, that is $\mathcal{X}(\nu,k) = |X(\nu,k)|$ and $\mathcal{D}(\nu,k) = |D(\nu,k)|$, respectively.

The predicted spectrum of the clean speech at the ν-th frame is finally obtained from the DNN output by applying the Wiener filter [29] as given by,

$$\hat{X}(\nu,k) = \frac{P_{\mathcal{X}}(\nu,k)}{P_{\mathcal{X}}(\nu,k) + P_{\mathcal{D}}(\nu,k)}\, Y(\nu,k) \tag{4}$$

In this expression, the quantities $P_{\mathcal{X}}(\nu,k)$ and $P_{\mathcal{D}}(\nu,k)$ represent the smoothed clean speech and noise power spectral densities (PSDs) for the k-th frequency bin and ν-th frame. They are computed recursively as,

$$P_{\mathcal{X}}(\nu,k) = \tau_x P_{\mathcal{X}}(\nu-1,k) + (1-\tau_x)\hat{\mathcal{X}}(\nu,k)^2 \tag{5}$$

$$P_{\mathcal{D}}(\nu,k) = \tau_d P_{\mathcal{D}}(\nu-1,k) + (1-\tau_d)\hat{\mathcal{D}}(\nu,k)^2 \tag{6}$$

where τ_x and τ_d denote the temporal smoothing factors for the clean speech and noise, respectively.

B. Training Procedure

In the training stage, we estimate the weight matrices $\mathbf{W}^{(l)}$ and bias vectors $\mathbf{b}^{(l)}$ for each layer, i.e. for all $l \in \{1,2,\ldots,L\}$ by employing training data, represented by the triplet $\{\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{D}}\}$. The latter consists of the input noisy speech matrix $\boldsymbol{\mathcal{Y}} = [\boldsymbol{\mathcal{Y}}_1, \ldots, \boldsymbol{\mathcal{Y}}_N]$, clean speech target $\boldsymbol{\mathcal{X}} = [\boldsymbol{\mathcal{X}}_1, \ldots, \boldsymbol{\mathcal{X}}_N]$, and clean noise target $\boldsymbol{\mathcal{D}} = [\boldsymbol{\mathcal{D}}_1, \ldots, \boldsymbol{\mathcal{D}}_N]$.

The DNN parameters are estimated by minimizing a suitable cost function. Among the different cost functions available for this task, such as the mean-squared error (MSE), cross-entropy, Kullback-Leibler and Itakura-Staito divergence, the minimum MSE (MMSE) turns out to perform better in our work. The MSE function of the DNN output is calculated as,

$$E = \frac{1}{N}\sum_{n=1}^{N} \|[\hat{\boldsymbol{\mathcal{X}}}_n, \hat{\boldsymbol{\mathcal{D}}}_n] - [\boldsymbol{\mathcal{X}}_n, \boldsymbol{\mathcal{D}}_n]\|_2^2 + \lambda \sum_{l=1}^{L} \|\mathbf{W}^{(l)}\|_2^2 \quad (7)$$

where $[\hat{\boldsymbol{\mathcal{X}}}_n, \hat{\boldsymbol{\mathcal{D}}}_n]$ and $[\boldsymbol{\mathcal{X}}_n, \boldsymbol{\mathcal{D}}_n]$ denote the estimated and target spectral feature vectors of the clean speech and noise pair, respectively. In order to avoid overfitting, Ridge regularization is considered through the term $\lambda \sum_{l=1}^{L} \|\mathbf{W}^{(l)}\|_2^2$, where $\lambda > 0$ is the regularization parameter.

We can use the error backpropagation technique to estimate the parameters that minimize the cost function in (7), such as the common stochastic gradient descent algorithm, conjugate gradient and Levenberg-Marquardt algorithms [30]. In addition, there is an interest towards using an additional greedy layer-wise pre-training stage via the RBM [17] or autoencoder techniques [18].

However, these approaches are computationally expensive and do not seem to critically affect the final enhancement performance of the DNN with ReLU activation function, given sufficiently large and varied data sets. In this paper therefore, we choose an improved version of the resilient back-propagation (Rprop) [31] algorithm, called iRprop$^-$ and presented in [32]. It is a first-order iterative learning algorithm, which has been shown to provide a rapid and reliable convergence compared to the conjugate gradient algorithm with much less computations.

III. Proposed framework

In this section, we first introduce the proposed DNN framework for speech enhancement in general terms. Then speech feature extraction based upon MFCC is briefly reviewed. The spectral feature mapping introduced in this work is then presented. Finally, we discuss the computational complexity of the resulting low complexity DNN scheme in comparison to the STFT-based DNN approach.

A. Proposed Structure and Motivation

DNN has received much attention in the field of speech enhancement and automatic speech recognition over the past few years. The acoustic feature extraction plays a key role as a pre-processing stage to these tasks. The MFCCs are one of the most commonly used features in this context as they provide a spectral representation of speech that incorporates some aspects of audition. Implementation of the spectral feature mapping technique using MFCC features has the advantage of reducing the length of the input feature vector. Hence, a smaller DNN model (i.e., with reduced number of nodes) can be employed. Consequently, this leads to a faster convergence time in training and lower computational complexity as compared to the conventional STFT approach. In addition, the process of calculating the MFCC vectors from the observed speech signal includes some non invertible stages. It might be possible to make certain approximations about the information that has been discarded during this process to allow estimating the magnitude spectrum of the input speech as a result of the MFCC inversion process. Yet, it is still a challenge to ensure that the MFCC inversion process will achieve perfect reconstruction without additional computational complexity.

Therefore, in this work, a spectral feature mapping from noisy MFCC to the enhanced STFT spectrum is introduced based on DNN modeling, in order to predict the clean speech signal from a noise corrupted input signal. Mapping the MFCC features directly into the frequency domain allows one to bypass the information loss caused by the inversion of the MFCC process. These are the main motivations for the proposed approach, whose main processing steps are summarized below.

A block diagram of the proposed DNN approach for speech enhancement system is illustrated in Fig. 2. The system operation consists of two stages, that is, training and enhancement. In both stages, we consider the MFCC of the noisy speech signal as the input feature to the DNN, where a more detailed description of the MFCC computation is provided in the following subsection. In the training stage, a regression-based DNN model is trained using the training features from the triplet of the noisy and clean speech, and noise data. In the enhancement stage, the clean speech magnitude spectrum is predicted from processing the noisy speech frames by the well-trained DNN model. Finally, the clean speech spectrum is estimated via the Wiener filtering as introduced in Section II.A. The time-domain enhanced speech signal is obtained via the inverse STFT followed by the overlap-add method.

B. MFCC Feature Mapping

In the MFCC feature extraction module, the speech signal is passed through a first order FIR filter in the pre-emphasis stage to boost the highband formants. Specifically, the filter output signal is computed as,

$$y'[n] = y[n] - \alpha y[n-1] \quad (8)$$

where α is a pre-emphasis coefficient, $0.95 \leq \alpha \leq 1$. Next, the short time Fourier transform of the boosted speech signal $y'[n]$ is computed as mentioned in Section II. Specifically, the signal is segmented into consecutive overlapping frames of length K and each frame is multiplied with an analysis window. Then for each windowed frame, a DFT is computed. The squared magnitude of the resulting STFT coefficients, i.e., $|\mathcal{Y}_{\nu}^{'}(k)|^2$, are then passed through a mel-scale filterbank, consisting of M overlapping triangular pass-band filters [33], indexed by $m \in \{0, 1, .., M-1\}$. Specifically, for the m-th pass-band filter, the filter output denoted as $\mathcal{Y}_{\nu}^{''}(m)$, is calculated as a weighted sum of the

squared magnitude values within the corresponding pass-band [33], as expressed by,

$$\mathcal{Y}''_{\nu}(m) = \sum_{k=0}^{K'-1} |\mathcal{Y}'_{\nu}(k)|^2 \, \mathcal{W}_{km} \tag{9}$$

where $\mathcal{W}_{km} \geqslant 0$ is the (k,m)-th entry of the filterbank matrix $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{K' \times M}$. Next, a logarithmic operation is applied to the filter outputs. Finally, the outputs are further processed by taking the Type III discrete cosine transform (DCT), as expressed by [34],

$$\mathcal{C}_{\nu}(p) = \sqrt{\frac{2}{M}} \sum_{m=0}^{M-1} (\log_{10} \mathcal{Y}''_{\nu}(m)) \cos(\frac{p\pi}{M}(m-0.5)) \tag{10}$$

where $\mathcal{C}_{\nu}(p)$ refers to the p-th MFCC, $p \in \{0, 1, \ldots, P-1\}$, and P is the number of mel-scale cepstral coefficients. For convenience, we define the vectors: $\boldsymbol{\mathcal{Y}}'_{\nu} = [\mathcal{Y}'_{\nu}(0),\ \mathcal{Y}'_{\nu}(1),\ \ldots,\ \mathcal{Y}'_{\nu}(K'-1)]$, $\boldsymbol{\mathcal{Y}}''_{\nu} = [\mathcal{Y}''_{\nu}(0),\ \mathcal{Y}''_{\nu}(1),\ \ldots,\ \mathcal{Y}''_{\nu}(M-1)]$ and $\boldsymbol{\mathcal{C}}_{\nu} = [\mathcal{C}_{\nu}(0),\ \mathcal{C}_{\nu}(1),\ \ldots,\ \mathcal{C}_{\nu}(P-1)]$, also shown in Fig. 2.

C. Incorporation of MFCC within DNN

While the training procedure is the same as that explained in section II, the DNN structure is less complex. Each hidden layer, labeled with index $l \in \{1, 2, \ldots, L-1\}$ consists of I' neurons, where $I' = I/\beta$, I is the number of neurons per layer in an STFT-based DNN system with similar performance, and $\beta > 1$ is a complexity reduction factor. In the training phase, the network is presented with the input noisy speech matrix $\boldsymbol{\mathcal{C}} = [\boldsymbol{\mathcal{C}}_1, \ldots, \boldsymbol{\mathcal{C}}_N]$, the clean speech target matrix $\boldsymbol{\mathcal{X}} = [\boldsymbol{\mathcal{X}}_1, \ldots, \boldsymbol{\mathcal{X}}_N]$, and the noise target matrix $\boldsymbol{\mathcal{D}} = [\boldsymbol{\mathcal{D}}_1, \ldots, \boldsymbol{\mathcal{D}}_N]$.

In the enhancement stage, at the ν-th frame, the network is presented with the noisy input MFCC vector $\boldsymbol{\mathcal{C}}_{\nu}$. The output layer, $\mathbf{h}^{(L)} = [\hat{\boldsymbol{\mathcal{X}}}_{\nu}, \hat{\boldsymbol{\mathcal{D}}}_{\nu}]$ consists of two K'-dimensional prediction components. In this notation, $\hat{\boldsymbol{\mathcal{X}}}_{\nu} = [\hat{\mathcal{X}}_{\nu,0},\ \hat{\mathcal{X}}_{\nu,1},\ \ldots,\ \hat{\mathcal{X}}_{\nu,K'-1}]$, and $\hat{\boldsymbol{\mathcal{D}}}_{\nu} = [\hat{\mathcal{D}}_{\nu,0},\ \hat{\mathcal{D}}v_{\nu,1},\ \ldots,\ \hat{\mathcal{D}}_{\nu,K'-1}]$, where the components $\hat{\mathcal{X}}_{\nu,k}$ and $\hat{\mathcal{D}}_{\nu,k}$ provide preliminary estimates of the clean speech and noise spectrum magnitudes, that is $\mathcal{X}_{\nu,k} \equiv |X(\nu,k)|$ and $\mathcal{D}_{\nu,k} \equiv |D(\nu,k)|$, respectively. After DNN processing, the predicted magnitude spectrum of the clean speech for the ν-th frame is derived by applying a Wiener filter, as explained in section II.

D. Complexity

The computational complexity of an algorithm is often measured in terms of the number of computer instructions or operation cycles (e.g., floating point multiplications) needed to execute the algorithm [35]. It depends upon the implementation of the individual sub-algorithms composed the speech enhancement system. In the acoustic feature extraction the pre-emphasis and windowing require $2K$ multiplications per frame. The STFT can be implemented using FFT with $K \log_2 K$ complexity, as opposed to a direct realization of the DFT with complexity K^2. The required STFT magnitude coefficients for each frame are then computed at the cost of $2K'$. These values are used as input to the mel-scale triangular filterbank with complexity upper bounded by MK'. The DCT in the mel-scale cepstral analysis stage is implemented using the fast cosine transform (FCT) algorithm with complexity $M \log_2 M$. The signal reconstruction module involves: updating the speech and noise PSD based on (5)-(6) at the cost of $6K'$ per frame; implementing the Wiener filter in (4) at the cost $4K'$ per frame; and signal reconstruction via inverse STFT, which requires $K \log_2 K$ multiplications per frame.

We now consider a DNN with L hidden layers, each containing I neurons for simplicity, and I_L output neurons, and assume that the maximum number of training iterations is T. In the training stage including the forward and the backward propagation, the MFCC-based DNN requires $3(L-2)I'^2T + 3I'I_LT + 2PI'T + 2I_LT$ multiplications per frame, while the STFT-based DNN requires $3(L-2)I^2T + 3II_LT + 2K'IT + 2I_LT$ multiplications per frame, where $I' = I/\beta$ [36]. Subsequently, the reduced computational workload of the low complexity DNN, that is $I' < I$ and $P < K'$, will allow for a faster running time.

IV. Experimental Results

A. Methodology

The clean speech signals used in our experiment were selected from the TSP-speech database [37] and consisted of 1500 utterances from 25 different male and female speakers (60 utterances per speaker). As for the noise signals, five different types were selected from the NoiseX92 database [38], namely: babble, pink, buccaneer2, factory1, and hfchannel. The noisy speech utterances were generated by adding noise sequences to the clean speech, appropriately scaled to achieve input SNRs of 0, 5 and 10 dB. The sampling frequency of all the signals was set to its original value of 16 KHz. The noisy speech utterances were divided into two sets. The first set, referred to as the training and validation set, includes 18750 utterances, corresponding to 11 hours of speech, while the second set referred to as the test set, includes 3750 utterances, corresponding to 2 hours.

To compare the proposed method, we implemented a standard NMF approach and a STFT-based DNN. The basic settings for the STFT analysis and synthesis were kept identical for all three methods. Specifically, a Hanning window was employed in computing the STFT. The length of the window was set to $K = 1024$ with a 75% frame overlap for both the analysis and the synthesis. The values of $(\tau_x, \tau_d) = (0.4\ ,\ 0.9)$, $\lambda = 0.01$ are used as the temporal smoothing factors and the regularization parameter, respectively. The same dataset was also applied to train and evaluate all three methods. For the implementation of the MFCC-based DNN, we consider $M = 64$ filterbank channels in the frequency range of $[300, 3700]$ Hz and the pre-emphasis factor of $\alpha = 0.97$. In our experiment, the MFCC-based DNN is input with the corresponding vector of $P = 22$ MFCC, while the output consists of two vectors

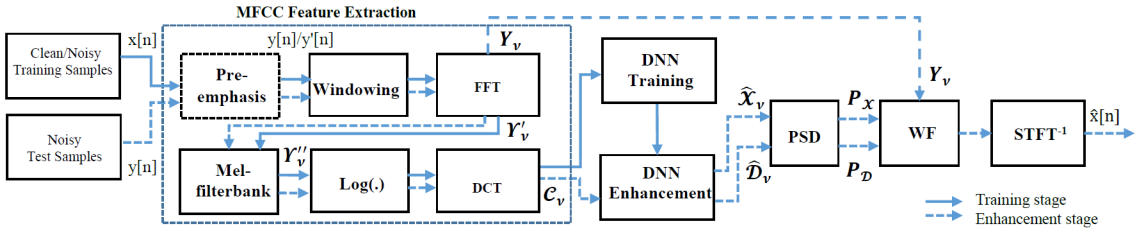

Fig. 2. Block diagram of the DNN-based speech enhancement system

containing the $K' = 513$ magnitude coefficients of the clean speech and noise signals, respectively. We considered a noise dependent application for implementing the NMF algorithm based on the Kullback-Leibler divergence (KL) with 80 basis vectors.

B. Results and Discussion

In order to evaluate the quality of the enhanced signal, the PESQ [39] and the SDR [40], are used. For all the measures, a higher value implies a better result.

TABLE I
RUNNING TIME INCLUDING THE TRAINING

Time (min)	NMF	DNN-STFT	DNN-MFCC
Training	5	38	8

The numerical experiments were run on a computer featuring the Intel(R) Xeon(R) central processing unit (CPU), 2 processors operating at the speed of 2.3 GHz, and 64GB of RAM. In our experiments, we have compared the speech enhancement performance based on both DNN models trained with different size of hidden layers, i.e., 1024, 2048, and 4096. The optimum number of hidden layers resulting in the best performance of the STFT and the MFCC-based DNN are 4096 and 1024, respectively.

Table I demonstrates the running time comparison of different algorithms. In the DNN-STFT approach, we trained a DNN model with 2 hidden layers of size $I = 4096$ units each and the STFT features as input. In the DNN-MFCC approach, the MFCC input features were applied to a DNN model with 2 hidden layers of size $I' = 1024$ units each. It can be seen that the proposed DNN structure using the MFCC features as the input as apposed to the STFT features, leads to a significant reduction in training time complexity, where the runtime is reduced by a factor of approximately 5 (4.75) in our experiments.

Fig. 3 gives the average PESQ performance comparison of different number of MFCC coefficients for pink noise at 5 dB input SNR. As shown, the optimal value of MFCC coefficient of 22 is chosen for the proposed method.

The PESQ and SDR scores for the benchmark and the proposed algorithms per each noise type and SNR level are provided in Tables II. The results presented here show that the proposed MFCC-based DNN method outperforms the NMF and STFT-based DNN in most cases, for the objective performance metrics under consideration.

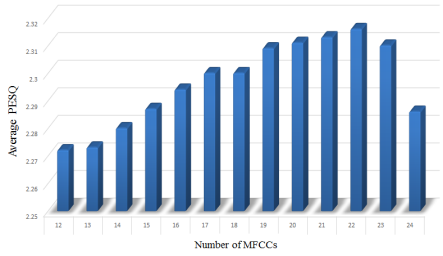

Fig. 3. Average PESQ results for different numbers of MFCCs

TABLE II
AVERAGE PESQ VALUES FOR PINK (N1), BABBLE (N2),BUCCANEER2 (N3), FACTORY1 (N4), AND HFCHANNEL (N5) NOISE

SNR	Algo.	Eval.	N1	N2	N3	N4	N5
0dB	Noisy	PESQ	1.31	1.45	1.13	1.36	1.19
		SDR	0.02	0.03	0.01	0.01	0.01
	NMF	PESQ	1.70	1.52	1.67	1.60	1.62
		SDR	5.16	2.42	3.89	4.05	5.45
	DNN-STFT	PESQ	2.06	1.82	1.90	1.88	2.00
		SDR	5.23	2.53	4.03	4.23	5.53
	DNN-MFCC	PESQ	**2.12**	**1.92**	**2.03**	**1.93**	**2.09**
		SDR	**5.36**	**2.60**	**4.11**	**4.24**	**5.61**
5dB	Noisy	PESQ	1.68	1.83	1.52	1.73	1.45
		SDR	5.01	5.02	5.01	5.00	5.00
	NMF	PESQ	2.13	1.92	2.07	2.05	1.80
		SDR	10.08	7.28	8.64	8.84	10.01
	DNN-STFT	PESQ	2.21	2.20	2.08	2.20	2.09
		SDR	10.31	7.43	8.32	**8.90**	11.30
	DNN-MFCC	PESQ	**2.23**	**2.22**	**2.26**	**2.24**	**2.17**
		SDR	**10.38**	**7.44**	**8.39**	8.89	**11.41**
10dB	Noisy	PESQ	2.08	2.01	1.90	2.11	1.79
		SDR	10.00	10.02	10.01	10.00	10.01
	NMF	PESQ	2.45	2.24	2.30	2.30	2.22
		SDR	14.30	11.45	13.07	13.26	14.03
	DNN-STFT	PESQ	2.32	2.31	2.24	2.33	2.14
		SDR	14.57	11.68	13.80	13.29	14.12
	DNN-MFCC	PESQ	**2.46**	**2.39**	**2.42**	**2.36**	**2.24**
		SDR	**14.77**	**11.71**	**13.83**	**13.32**	**14.33**

V. CONCLUSION

Deep neural network have been a subject of interest in many fields, such as speech enhancement. While other deep learning techniques are promising in the speech enhancement task, they are complex systems to implement. The main objective of this work was to reduce the DNN complexity by introducing a spectral feature mapping from the noisy mel frequency cepstral coefficients (MFCC) to the enhanced short time Fourier transform (STFT) spectrum. Therefore, a low-complexity DNN model is presented, in order to efficiently perform noise suppression in a single

channel speech enhancement. In this paper, we have implemented a regression-based DNN approach for single-channel speech enhancement. Although we can extend the proposed approach to RNN, we consider the regression based feed-forward DNN for our problem. We implemented the proposed DNN model with different numbers of MFCC coefficients and network structure and were able to achieve a significant reduction in runtime by a factor of 4.75. The system performance was evaluated using SDR and PESQ scores which is shown that the proposed approach outperforms other benchmark algorithms in most cases while remaining relatively simple.

REFERENCES

[1] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 27, pp. 113–120, Apr. 1979.

[2] N. Virag, "Single channel speech enhancement based on masking properties of the human auditory system," *IEEE Trans. Speech Audio Process.*, vol. 7, pp. 126–137, Mar. 1999.

[3] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," vol. 67, pp. 1586–1604, Dec. 1979.

[4] P. Scalart and J. V. Filho, "Speech enhancement based on a priori signal to noise estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 2, pp. 629–632, May 1996.

[5] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, pp. 1109–121, Dec. 1984.

[6] E. Plourde and B. Champagne, "Auditory-based spectral amplitude estimators for speech enhancement," *IEEE Trans. Audio, Speech, Language Process.*, vol. 16, pp. 1614–1623, Nov. 2008.

[7] K. Paliwal and A. Basu, "A speech enhancement method based on kalman filtering," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 12, pp. 177–180, Apr. 1987.

[8] R. Ishaq, B. G. Zapirain, *et al.*, "Subband modulator kalman filtering for single channel speech enhancement," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pp. 7442–7446, May 2013.

[9] Y. Ephraim and H. L. V. Trees, "A signal subspace approach for speech enhancement," *IEEE Trans. Speech Audio Process.*, vol. 3, pp. 251–266, Jul. 1995.

[10] K. Hermus, P. Wambacq, *et al.*, "A review of signal subspace speech enhancement and its application to noise robust speech recognition," *EURASIP J. Adv. Signal Process.*, vol. 2007, no. 1, pp. 1–15, 2006.

[11] A. Chaudhari and S. B. Dhonde, "A review on speech enhancement techniques," in *Proc. Int. Conf. Pervasive Computing*, pp. 1–3, Jan. 2015.

[12] N. Mohammadiha, P. Smaragdis, *et al.*, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 10, pp. 2140–2151, 2013.

[13] H. Chung, E. Plourde, *et al.*, "Discriminative training of nmf model based on class probabilities for speech enhancement," *IEEE Signal Process. Lett.*, vol. 23, pp. 502–506, Apr. 2016.

[14] G. Hinton, L. Deng, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Mag.*, vol. 29, pp. 82–97, Nov. 2012.

[15] D. Liu, P. Smaragdis, *et al.*, "Experiments on deep learning for speech denoising.," in *Interspeech*, (Singapore), pp. 2685–2689, 2014.

[16] E. A. Wan and A. T. Nelson, "Networks for speech enhancement," *neural networks speech process. Artech House, Boston, USA*, vol. 139, 1999.

[17] Y. Xu, J. Du, *et al.*, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, pp. 7–19, Jan/ 2015.

[18] X. Lu, Y. Tsao, *et al.*, "Speech enhancement based on deep denoising autoencoder," in *Proc. INTERSPEECH Conf.*, 2013.

[19] W. Chan and I. Lane, "Deep recurrent neural networks for acoustic modelling," *arXiv preprint arXiv:1504.01482*, Apr. 2015.

[20] A. Narayanan and D. Wang, "Investigation of speech separation as a front-end for noise robust speech recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 22, pp. 826–835, Apr. 2014.

[21] Y. Tachioka, S. Watanabe, *et al.*, "Sequence discriminative training for low-rank deep neural networks," in *Proc. IEEE Global Conf. Signal Info. Process.*, pp. 572–576, Dec. 2014.

[22] T. G. Kang, K. Kwon, *et al.*, "Nmf-based target source separation using deep neural network," *IEEE Signal Process. Letters*, vol. 22, pp. 229–233, Feb. 2015.

[23] J. Du, Q. Wang, *et al.*, "Robust speech recognition with speech enhanced deep neural networks.," in *Proc. INTERSPEECH Conf.*, pp. 616–620, 2014.

[24] P. S. Huang, M. Kim, *et al.*, "Deep learning for monaural speech separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pp. 1562–1566, May 2014.

[25] A. Maas, Q. V. Le, *et al.*, "Recurrent neural networks for noise reduction in robust asr," in *INTERSPEECH*, 2012.

[26] F. Weninger, H. Erdogan, *et al.*, "Speech enhancement with lstm recurrent neural networks and its application to noise-robust asr," in *Proc. Int. Conf. Latent Variable Analysis Signal Separation*, pp. 91–99, Springer, Aug. 2015.

[27] S. Shin, K. Hwang, *et al.*, "Fixed-point performance analysis of recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, (Shanghai, China), pp. 976–980, Mar. 2016.

[28] B. Karlik and A. V. Olgac, "Performance analysis of various activation functions in generalized mlp architectures of neural networks," *Int. J. Artificial Intel. Expert Sys.*, vol. 1, no. 4, pp. 111–122, 2011.

[29] E. M. Grais, M. U. Sen, *et al.*, "Deep neural networks for single channel source separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pp. 3734–3738, May 2014.

[30] S. Haykin, *Neural networks and learning machines*, vol. 3. Pearson Upper Saddle River, NJ, USA, 2009.

[31] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the rprop algorithm," in *Proc. IEEE Int. Conf. Neural Networks*, vol. 1, pp. 586–591, 1993.

[32] C. Igel and M. Hüsken, "Improving the rprop learning algorithm," in *Proc. 2nd Int. Symp. Neural Computation*, vol. 2000, pp. 115–121, Citeseer, 2000.

[33] S. Young, G. Evermann, *et al.*, "The htk book," *Cambridge university engineering department*, vol. 3, p. 175, 2002.

[34] D. O'Shaughnessy, *Speech Communications: Human and Machine*. Wiley-IEEE Press, 1999.

[35] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[36] P. Vincent, de Brébisson, *et al.*, "Efficient exact gradient update for training deep networks with very large sparse targets," in *Advances in Neural Information Process. Sys.*, pp. 1108–1116, 2015.

[37] R. University, "Signal processing information base: noise data," Nov. 1993.

[38] P. Kabal, "Tsp speech database," *McGill University, Database Version*, vol. 1, pp. 09–02, 2002.

[39] A. Rix, J. Beerends, *et al.*, "Perceptual evaluation of speech quality (pesq), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs," *ITU-T Recommendation*, p. 862, Feb. 2001.

[40] E. Vincent, R. Gribonval, *et al.*, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, pp. 1462–1469, Jul. 2006.