An Experimental Study on Model-based Monaural Speech Denoising

(単一マイク入力を用いたモデルベースな雑音除去に関する実験的検討)

李 浩羽

LI Haoyu

ID Number: 37-165059

Supervisor: Prof. Nobuaki Minematsu

Department of Electrical Engineering and Information Systems,
Graduate School of Engineering,
The University of Tokyo

Master Thesis

July 2018

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and acknowledgements.

LI Haoyu
July 2018

Acknowledgements

I would like to dedicate this thesis to my loving family, who have kept giving me great spiritual and material support. I would like to thank my supervisor Prof. Nobuaki Minematsu, Lecturer Daisuke Saito, and all lab members who have given help to me in both lives and researches. Besides, I would like to thank all my friends I met in Japan. It is the cherished memory in my life that the time I spent together with them. I am very grateful to all of them.

Abstract

Monaural speech denoising is a technology that aims to improve the intelligibility and quality of degraded speech through a process of suppressing noise. Recent years, model-based approaches attract plenty of research interests due to their high effectiveness, especially in non-stationary noise. This thesis mainly focuses on the model-based monaural speech denoising, including NMF (Non-negative Matrix Factorization) -based methods, DNN (Deep Neural Network) -based methods and the combinations of the above two. Experiments are conducted to investigate the performances of the different methods in several objective measures. In addition, for DNN-based denoising, low-rank decomposition is employed to compress the neural networks but without too much performance degradation. The reduced calculation complexity makes this method more practical, especially for the real-time applications.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Speech is the main media for human to communicate with each other. However, the real world is always full of noise which seriously interfere the communication via speech. This thesis focuses on the monaural speech denoising, which aims to recover the clean speech from the observed single-channel noisy speech through a process of suppressing noise. It has many promising applications such as hearing aids, mobile speech communication and preprocessor to a robust speech recognition system. This challenging task has attracted plenty of research interests over the past several decades, and numerous methods have been proposed.

Spectral Subtraction method [5] subtracts an estimate of the short time noise amplitude spectrum to produce an estimated spectrum of the clean speech. However, this method inevitably brings annoying musical noise. Other conventional methods, [8, 9] were proposed to suppress musical noise. Among them, log-MMSE [9] shows the best performance because log domain is considered to be suitable for human perception. Those methods have achieved quite good performance under stationary noise condition. However, noise we hear in the real world is usually non-stationary, which means its statistical property dynamically changes with time going on. These conventional approaches, unfortunately, fail to deal with such kind of noises.

In recent years, model-based approaches become more and more popular because they are confirmed [29, 37] to be more effective than conventional ones, especially in non-stationary noises. Two models are discussed in this thesis. The first model is called Non-negative Matrix Factorization (NMF), the other one is Deep Neural Networks (DNN).

NMF-based approach decomposes the noisy speech into speech and noise components, and the clean speech is then reconstructed by processing speech and noise components. Through the experiments, we show that speech bases trained by sparse NMF algorithm combined with noise bases trained by a clustering method can achieve significant improvement over the standard NMF method. Furthermore, we propose a new NMF+DNN framework by integrate NMF with DNN. Here DNN is employed as a post processor to achieve a better performance by correcting

activation vectors directly in the activation domain. This NMF+DNN framework consistently outperforms the pure NMF method and is also much better than the pure DNN method in the extremely low SNRs. In addition, our proposed strategy is demonstrated to be more effective compared to a previously proposed method of combining NMF and DNN. While the noise dictionaries of NMF model are usually trained by a specific noise type because the number of them should not be too big. Otherwise it will lead to a bad performance and inefficient computation. Consequently the NMF-based denoising system is usually noise-dependent, which indicates its generalization ability to the unknown noise types is not satisfying. However, it is still a promising solution to suppress the noise in a specific noise environment. For example, it can be implemented in the factory to suppress the factory noise, or in the car to suppress the traffic noise, etc.

Besides NMF, DNN is used to model the relationship between the clean speech signals and its noisy signals in [37]. Unlike NMF model, it is straightforward to improve the DNN-based denoising performances to the unknown noises, through feeding the speech corpus with multiple noises. Therefore a practical DNN-based speech denoising system has to be trained by a large corpus with multiple noises. Such method eases the noise generalization problem but leads to a relatively big DNN model, that is not good for those real-time applications like mobile speech communication. In this thesis, we experimentally investigate different DNN architectures with different training targets to seek the most suitable configurations for supervised speech denoising task. Moreover, we apply low-rank decomposition technique to compress the neural networks. The compact structure with much smaller model size still shows comparable experimental results to those in the original DNN.

This thesis is organized as follows. Chapter 2 introduces some conventional speech denoising approaches and points out their limitations. Chapter 3 elaborates the standard NMF-based method and several effective training methods for improvements. The proposed NMF+DNN framework will be also explained. Chapter 4 explains the DNN-based approach and its compact implementation. Chapter 5 gives the detailed experimental settings and results. Chapter 6 summarizes the whole thesis and gives some future works.

Chapter 2

Overview of the Conventional Speech Denoising Approaches

This chapter gives an overview of the basic knowledge of speech signal processing. Then the conventional speech denoising approaches, including Spectral Subtraction, and Log-MMSE Estimator, will be discussed.

2.1 Basic knowledge of speech signal processing

Clean speech $s(t)$ is mixed with noise $n(t)$. The speech $x(t)$ that we hear is a result of addition of these signals as Equation (2.1) shows. The goal of speech denoising is to construct the estimated clean speech $\hat{s}(t)$ from the observed noisy $x(t)$. Followings are the typical steps to process the noisy speech.

$$x(t) = s(t) + n(t) \tag{2.1}$$

Segment speech to short frames

Although the characteristic of speech signal is dynamically changing, it does not change much in a short time duration about 20-40 ms. This is why the speech signal should be segmented by the *window function* to the short frames at the first step. Also, Since the rectangular window leads to the discontinuity at the cutting edge of the adjacent frames, it has disadvantages of high sidelobe attenuation and bad inhibition of spectral leakage. In practice, *Hamming window*

is considered as a good choice:

$$w(n) = \begin{cases} 0.54 - 0.46\cos(\frac{2\pi}{N-1}) & 0 \leq n \leq N-1 \\ 0 & otherwise \end{cases} \tag{2.2}$$

Frequency domain processing

Usually the noise reduction is done in the frequency domain. By applying DFT (Discrete Fourier Transformation) of both sides of windowed frame. We have

$$X(m,k) = S(m,k) + N(m,k) \tag{2.3}$$

where m denotes the m-th frame and k denotes the index of frequency bin. Since human auditory system is insensitive to the phase information[27], we assume that clean speech and noise have the same phase angle as noisy speech for every frame. Therefore the core step of speech denoising, also the main contents of this thesis, is to obtain the estimated clean spectrum amplitude $|\hat{S}(m,k)|$ from the noisy observation.

Reconstruct speech waveform

After estimating the clean spectrum amplitude $|\hat{S}(m,k)|$, we have to reconstruct speech in the time domain. A common way, as Equation (2.4) shows, is to combine $|\hat{S}(m,k)|$ with the original noisy phase and apply inverse DFT.

$$\hat{s}(n) = IDFT(|\hat{S}(m,k)|e^{\Phi_x}) \tag{2.4}$$

where Φ_x is the phase of the noisy signal $X(m,k)$.

Note that the constructed $\hat{s}(n)$ is just a short speech frame since we use window function to segment the original speech. A method to synthesis the speech waveform and remove the effect of window function is called *Overlap-and-Add* algorithm.

2.2 Spectral subtraction

Spectral Subtraction was first proposed by [5] as the most simple and classic approach. Consider the Equation (2.3) and the assumption that phase can be omitted, we have

$$X = S + N \tag{2.5}$$

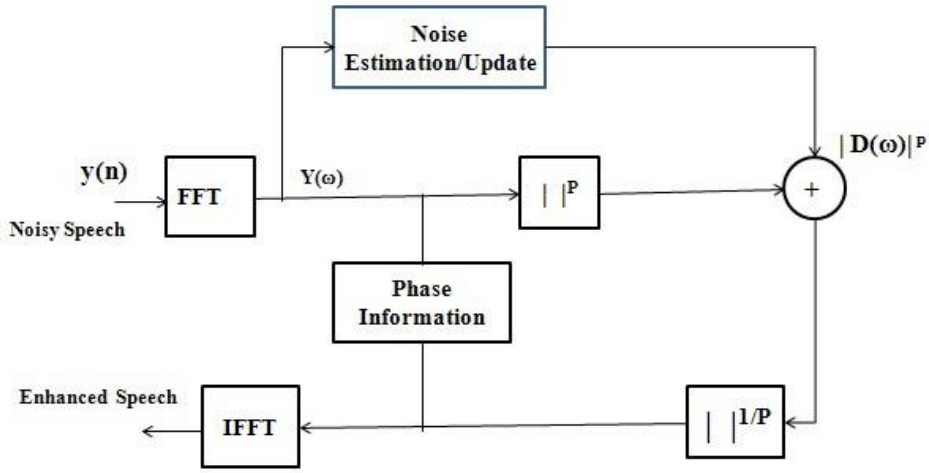


Fig. 2.1 The flow chart of generalized spectral subtraction algorithm[24].

where X, S, and N are the corresponding spectrum magnitudes of those in Equation (2.3). The key idea of Spectral Subtraction algorithm is to subtract an estimate of noise spectrum magnitude $\hat{N}$ to produce an estimated spectrum of the clean speech $\hat{S}$ as $\hat{S} = X - \hat{N}$.

A more generalized version of this algorithm is shown as Equation (2.6).

$$\hat{S}^p = X^p - \hat{N}^p \tag{2.6}$$

Here p denotes the exponent value. Figure 2.1 gives the flow chart of this generalized spectral subtraction algorithm.

2.3 Log-MMSE estimator

Log-MMSE estimator is an another statistical approach to denoise the speech. It can be derived by minimizing the expectation error of the log-magnitude spectra between the real speech spectrum S and the estimated one $\hat{S}$ as Equation (2.7) shows.

$$E\left\{(\log S - \log \hat{S})^2\right\} \tag{2.7}$$

The solution for $\hat{S}$ is relatively complicated. Before giving this solution, we have to define some terms. The terms ξ_k and γ_k are named as the *a priori* and *a posteriori* SNRs respectively,

where k denotes the index of frequency bin. The definitions are given as follows.

$$\xi_k = \frac{\lambda_s(k)}{\lambda_n(k)} \tag{2.8}$$

$$\gamma_k = \frac{X_k^2}{\lambda_n(k)} \tag{2.9}$$

where $\lambda_s(k) = E\{S_k^2\} \approx S_k^2$ and $\lambda_n(k) = E\{N_k^2\} \approx N_k^2$. The term v_k is then defined as:

$$v_k = \frac{\xi_k}{1+\xi_k}\gamma_k \tag{2.10}$$

Referring to [9], the optimal estimator is given by Equation (2.8):

$$\hat{S}_k = \frac{\xi_k}{\xi_k+1} exp\left\{\frac{1}{2}\int_{v_k}^{\infty}\frac{e^{-t}}{t}dx\right\}X_k \tag{2.11}$$

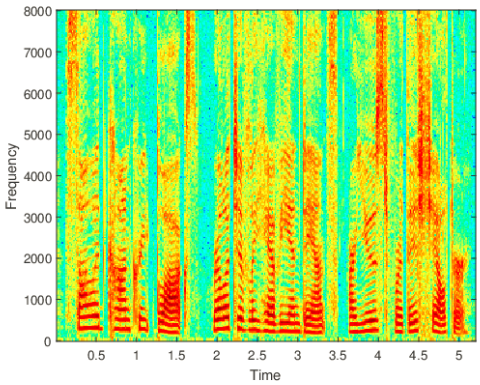
The spectrograms of the original noisy speech and processed speech are given in Figure 2.2. The original speech is corrupted by white Gaussian noise at SNR=3dB. For spectral subtraction, we can observe that there is still much residual noise in Figure 2.2 (c), which is called *musical noise*. Also, the speech component is seriously attenuated in the low-frequency. While Log-MMSE estimator gives a smoother result in Figure 2.2 (d). Even though the average level of background noise is little bit higher, the overall performance of Log-MMSE is considered to be better than that of Spectral Subtraction, at the cost of higher complexity.

2.4 Noise estimation

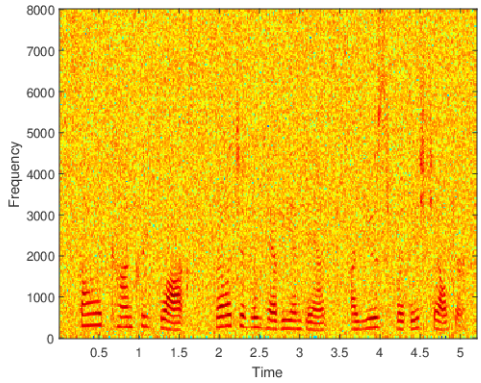
Either in Spectral Subtraction or Log-MMSE estimator, it is assumed that we have known the noise information, such as the power spectrum of noise $\hat{N}^2$ in Equation (2.6) (when $p = 2$) and a priori SNR ξ in Equation (2.11). This section gives some basic methods to estimate such noise information.

2.4.1 Update of noise power

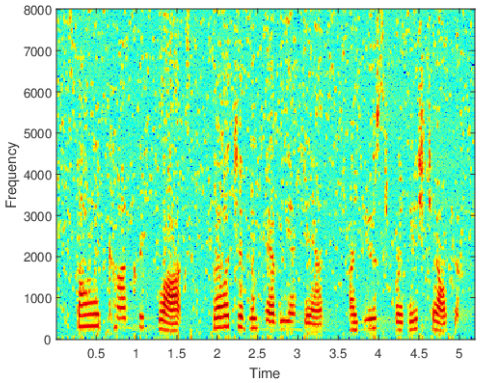
A simplest way is to estimate and update the noise power during the non-speech frame. A module, which is called *Voice Activity Detection* (VAD) [14], is employed to judge whether the signal segment is a speech frame or non-speech (or noise) frame. Then the power spectrum of the non-speech frame will be extracted to update the noise power using exponential smoothing,



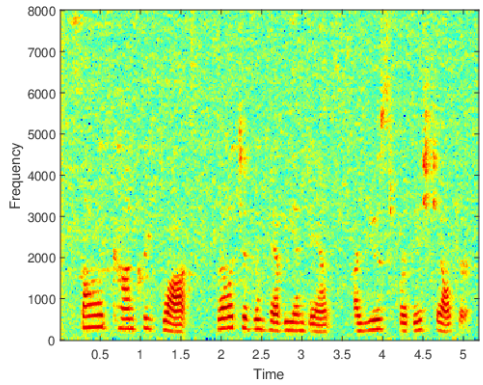
(a) Clean speech



(b) Noisy speech at SNR=3dB



(c) Processed by Spectral Subtraction



(d) Processed by Log-MMSE

Fig. 2.2 Spectrograms of speech with white Gaussian noise at SNR=3dB. (a) clean speech, (b) noisy (c) Spectral Subtraction, (d) Log-MMSE.

as Equation 2.12 shows.

$$\hat{N}^2(m) = \begin{cases} \alpha\hat{N}^2(m-1) + (1-\alpha)X^2(m), & noise\ frame \\ \hat{N}^2(m-1), & speech\ frame \end{cases} \tag{2.12}$$

Where m denotes the m-th frame and α is usually set as 0.98. After obtaining the noise power $\hat{N}^2$, it is straightforward to calculate $\hat{N}^p$ of Equation (2.6) by taking the exponential of $\frac{p}{2}$. In addition, a posteriori SNR γ, for Log-MMSE estimator, can be estimated simply by $\gamma = \frac{X^2}{\hat{N}^2}$.

2.4.2 Update of a priori SNR

A priori SNR ξ, which is defined as Equation (2.8), should be estimated when utilizing Log-MMSE estimator. Here we introduce a classic method, which is called *decision-directed approach* [8].

Consider the relationship between ξ_k and γ_k, we can rewrite Equation (2.8) as:

$$\begin{aligned}\xi_k(m) &= \frac{E\left\{X_k^2(m) - N_k^2(m)\right\}}{\lambda_n(k,m)} \\ &= \frac{E\left\{X_k^2(m)\right\}}{\lambda_n(k,m)} - \frac{E\left\{N_k^2(m)\right\}}{\lambda_n(k,m)} \\ &= E\{\gamma_k(m)\} - 1\end{aligned} \tag{2.13}$$

By combining Equation (2.8) and Equation (2.13), we have:

$$\xi_k(m) = E\left\{\frac{1}{2}\frac{S_k^2(m)}{\lambda_n(k,m)} + \frac{1}{2}[\gamma_k(m) - 1]\right\} \tag{2.14}$$

By using the previous equation, ξ_k is derived in a recursive way:

$$\begin{aligned}\hat{\xi}_k(m) &= \alpha\frac{\hat{S}_k^2(m-1)}{\lambda_n(k,m-1)} + (1-\alpha)max[\gamma_k(m) - 1, 0] \\ &= \alpha\hat{\xi}_k(m-1) + (1-\alpha)max[\gamma_k(m) - 1, 0]\end{aligned} \tag{2.15}$$

Note that the value $\frac{1}{2}$ in Equation (2.14) now is replaced by a weighting factor α (commonly set as 0.98). The operator max(.) ensures $\hat{\xi}_k$ to be non-negative since it should be a non-negative value from the definition. And finally, an initial condition is given by

$$\hat{\xi}_k(0) = \alpha + (1-\alpha)max[\gamma_k(0) - 1, 0] \tag{2.16}$$

A posteriori SNR γ can be estimated according to the above section, thus a priori SNR ξ can be estimated as well based on decision-directed approach. Therefore the Log-MMSE algorithm can be run normally.

2.5 Limitations of the conventional approaches

So far we have discussed the basic speech denoising approaches. Although numerous methods, such as [19, 25, 31, 23], have been proposed to improve their performances, limitations still remain. Among them, the biggest challenge is that the performances of those approaches seriously degrade under the non-stationary noise condition.

For almost all the conventional approaches, the noise information is needed to make algorithms work. In Section 2.4, some simple estimation methods are introduced. However, no matter the methods are simple or complex like [28], it is still so hard to estimate the

noise information accurately. That is because the noise we hear in our daily life is highly non-stationary in common. For such noises, the statistical property changes fast and constantly.

Hence it inspires me to turn to the model-based approaches. The principles and experimental results will be given in the following chapters.

Chapter 3

NMF-based Speech Denoising and Improved Technology

In this chapter, the principles of NMF-based speech denoising will be introduced. Besides, several strategies for performance improvements are discussed. First, we show that speech bases trained by sparse NMF algorithm combined with noise bases trained by a clustering method can achieve significant improvement over the standard NMF method. Second, deep neural network (DNN) is employed as a post processor to correct activation vectors, which are generated by the previous NMF algorithm, to achieve a better performance.

3.1 NMF-based speech denoising

3.1.1 Sparse NMF model

Non-negative matrix factorization (NMF) is an algorithm to decompose a matrix into the two matrices, with the property that all three matrices have no negative value. It is a useful tool to analyze the audio signal, since the spectrogram is inherently non-negative. Figure 3.1 give an illustration of NMF algorithm, where the original matrix V is approximately represented by the two matrices W and H.

When NMF algorithm is applied to Equation (2.5), it represents the matrix X as the production of W and H.

$$X = S + N \approx WH \tag{3.1}$$

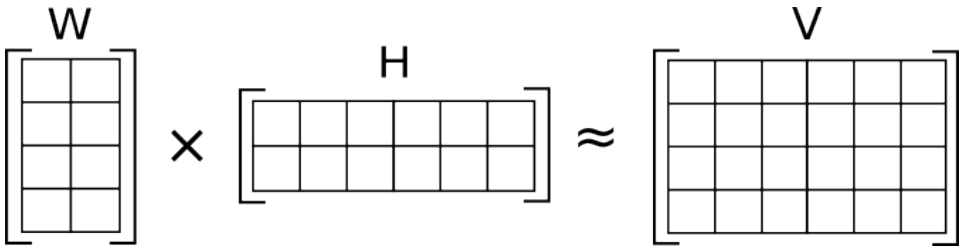


Fig. 3.1 An illustration of NMF algorithm.

Matrix W is called *dictionary* and H is called *activation*. The column vector $\mathbf{w_i}$ in matrix W is called *base* vector. Suppose the number of the bases is r, the j*th* column vector $\mathbf{x_j}$ of matrix X can be represented as the sum of the bases weighted by activation H.

$$\mathbf{x_j} = \begin{bmatrix} \mathbf{w_1} & \mathbf{w_2} & \cdots & \mathbf{w_r} \end{bmatrix} \begin{bmatrix} h_{1j} \\ h_{2j} \\ \vdots \\ h_{rj} \end{bmatrix} = \sum_{i=1}^{r} h_{ij}\mathbf{w_i} \tag{3.2}$$

where h_{ij} denotes the element in the i-th row, j-th column of activation matrix H. Considering Equation (3.2), we can easily find that base vectors $\mathbf{w_i}$ actually acts as *spectral templates*, which are used to build the real spectrum $\mathbf{x_j}$.

The cost function in Equation (3.3) is adopted to measure the similarity between X and the reconstructed matrix $\hat{X}$ $(= WH)$.

$$Cost = \frac{1}{2}||X - \overline{W}H||_F^2 + \lambda \sum_{ij} H_{ij} \tag{3.3}$$

where $\overline{W}$ is the column-wise normalized dictionary matrix. Cost function consists of Euclidean distance and an additional sparse regularization term. This additional term forces activation vector to be sparse, which means only few bases are active in meanwhile. Research in [30] indicates that a proper sparse regularization brings about benefit to improve the performance.

NMF algorithm iteratively updates the matrix W and H in order to minimize the cost function. The update rules [7] are shown as below.

$$H \leftarrow H \bullet \frac{\overline{W}^\top X}{\overline{W}^\top \overline{W} H + \lambda} \tag{3.4}$$

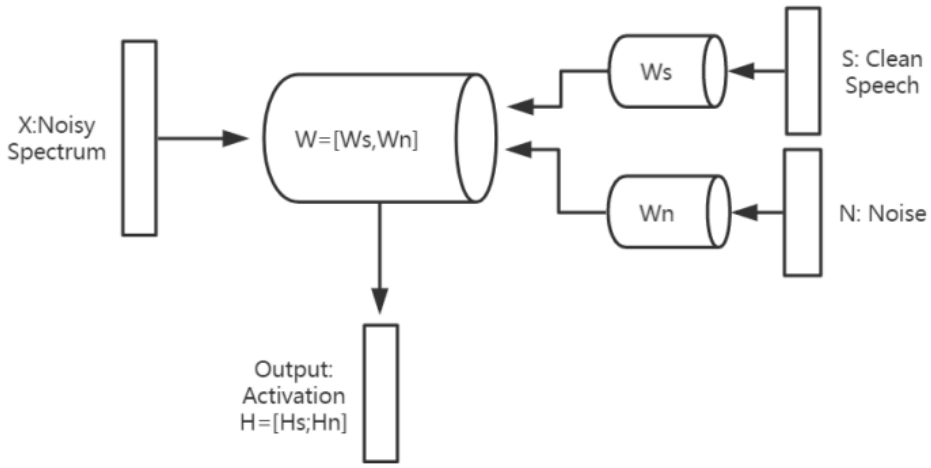


Fig. 3.2 A block diagram of NMF-based denoising method.

$$W \leftarrow \overline{W} \bullet \frac{XH^{\top} + \overline{W} \bullet \left(\mathbf{1} \left(\overline{W} HH^{\top} \bullet \overline{W} \right) \right)}{\overline{W} HH^{\top} + \overline{W} \bullet \left(\mathbf{1} \left(XH^{\top} \bullet \overline{W} \right) \right)} \tag{3.5}$$

where bold multiplications and divisions are element-wise operators. $\mathbf{1}$ denotes a matrix in which all elements are equal to one.

3.1.2 Speech denoising with NMF

In the task of speech denoising, we use speech and noise corpora to train the speech and noise dictionaries (W_s and W_n) respectively. Then we concatenate them into a big dictionary.

$$W = [W_s \; W_n] \tag{3.6}$$

In the denoising stage, we fix this big dictionary W and only calculate the activation matrix H through NMF algorithm. Because W consists of speech and noise dictionaries, we can divide activation H into speech and noise components similarly. Thus Equation (3.1) can be rewritten as Equation (3.7)

$$X = S + N \approx WH = \begin{bmatrix} W_s \; W_n \end{bmatrix} \begin{bmatrix} H_s \\ H_n \end{bmatrix} \tag{3.7}$$

The upper part of H is marked as H_s and lower part is H_n. They represent speech and noise components respectively. This NMF based denoising method is illustrated in Figure 3.2.

After we get activation matrix, an obvious approach to estimate the clean spectrogram $\hat{S}$ is based on Equation (3.8).

$$\hat{S} = W_s H_s \tag{3.8}$$

A more popular approach, which improves the speech quality, is to estimate a spectral mask [12], and this mask is further multiplied by noisy spectrogram X to obtain $\hat{S}$. As Equation (3.9) shows, spectral mask explains the contribution of the original spectrogram X when reconstructing clean spectrogram $\hat{S}$. If noise dominates in one time-frequency bin of X, the value of M will be close to zero. Thus this bin will be almost discarded. Similarly the bin of X will be saved to a large extent if speech dominates.

$$M = \frac{W_s H_s}{W_s H_s + W_n H_n} \tag{3.9}$$

$$\hat{S} = M \bullet X \tag{3.10}$$

The estimated clean spectrogram $\hat{S}$, combined with the phase extracted from the noisy signal, is then transformed to the speech signal $\hat{s}(t)$ in the time domain by inverse Fourier transformation.

3.1.3 Training methods for noise dictionary

NMF algorithm is effective to capture the hidden structures of signal and represents such structures as bases vector. Compared with the speech signal, however, there is generally no apparent structure in the noise signal. Consequently it is expected not to be efficient to train a noise dictionary by NMF. The residual error of Equation (3.3) is still large and not convergent even after plenty of iterations. Several methods including K-SVD [2], clustering [34] and sampling [11] are all available to train a noise dictionary instead of NMF. Among them, a clustering method is adopted here because it is quite simple but practical. Specifically, K-means [22] algorithm is applied to group the noise spectrums into several spectrum clusters. Those clusters are thus arranged to build the noise dictionary W_n. Performance improvement of denoising by using the clustering-based dictionary will be discussed in detail in Chapter 5.

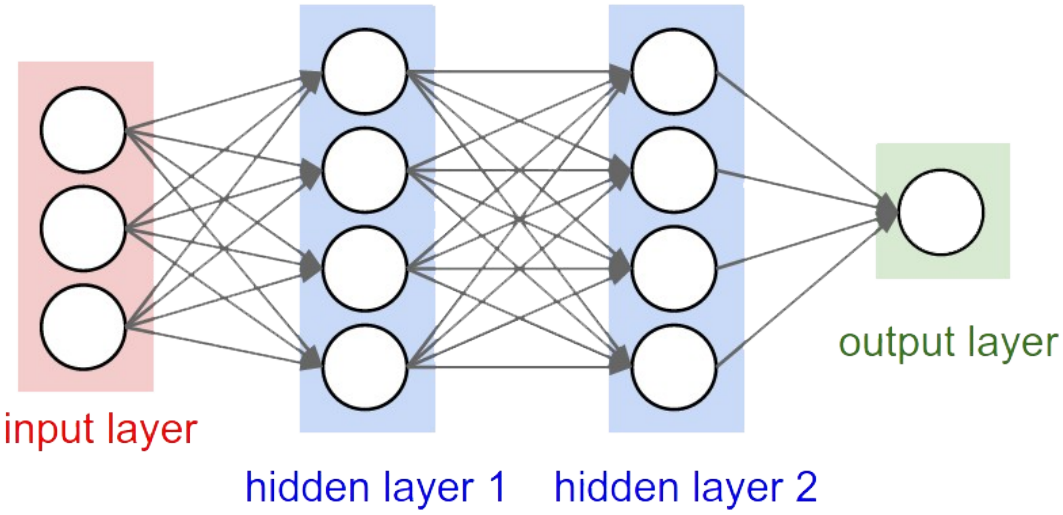


Fig. 3.3 A diagram of a two-hidden-layer DNN.

3.2 Improved technology: Integrated method with NMF and DNN

3.2.1 DNN model

We propose to incorporate Deep Neural Network (DNN) to further improve the performance of NMF-based approach. A basic DNN introduction is given in the following.

A DNN is literally a neural network with deep hidden layers. Figure 3.3 is an illustration of DNN model. As we can see in this figure, the left-most layer is called *input layer*, which accepts a vector as the input of network. Layers in the middle are called *hidden layers*, which take the output of its last layer as input, and deliver the output itself to the next layer. The right-most layer is called *output layer*, which represents the real output of the whole network.

Forward-feeding DNN shown Figure 3.3 is probably the most common architecture. After training stage, the parameters in DNN will be fixed. Then DNN accepts a vector as input, passes it through all its hidden layers and gives a result in the last output layer. The formula for hidden layers $i+1$ is:

$$x_{i+1,j} = h(\sum_q w_{iq}^{(i)} x_{iq} + b_j^{(i)}),$$

where x_{ij} is the value of j-th unit in layer i, $w^{(i)}$ is the weight parameter for layer i, $b^{(i)}$ is the bias for layer i, and h is the activation function. Thanks to the multiple stacks of linear transformations and non-linear activations, DNN is proved to have a powerful ability to model a complex relationship between its input and output vector[16].

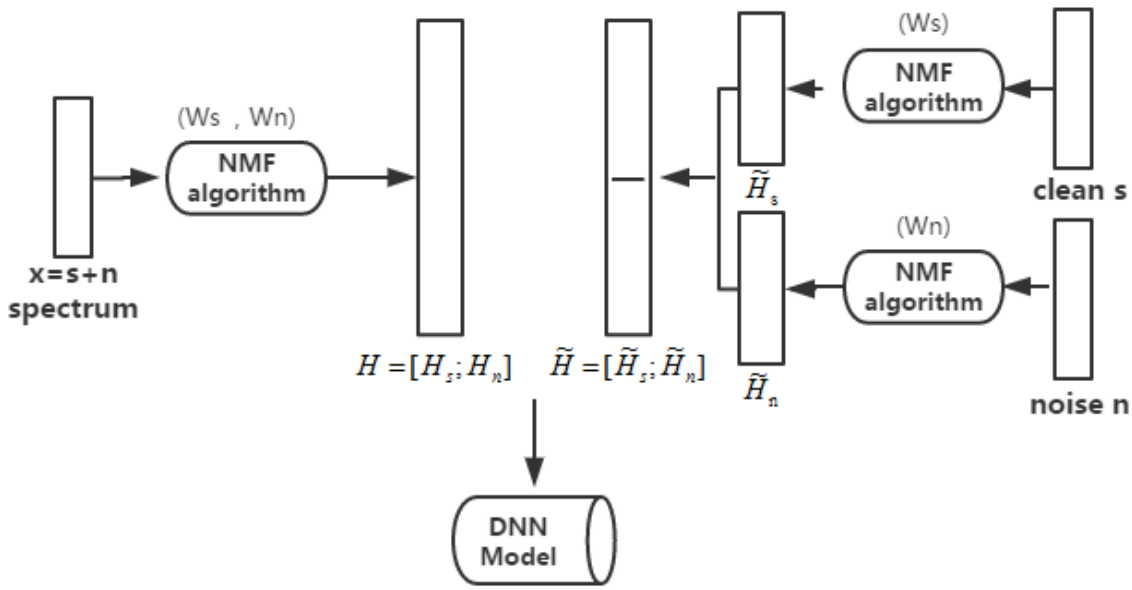
3.2.2 Integrating NMF with DNN

The NMF method described in Section 3.1 has achieved a good result. Nevertheless, the *overlap problem* exists and restricts the denoising performance. Speech and noise dictionaries correlate with each other to a large extent especially when the noise signal is acoustically similar to the speech signal. After being processed by the NMF algorithm, speech component H_s overlaps with noise component H_n to some degree, and this overlap will cause the serious degradation of denoising performance.

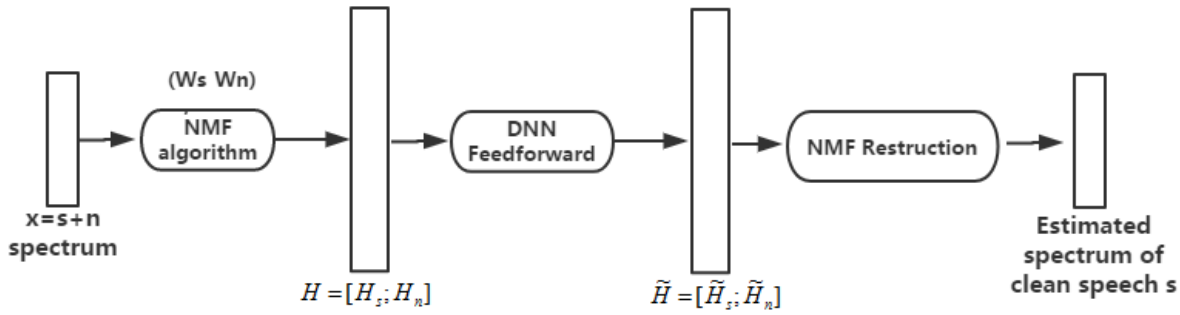
Previous work [21] pointed out this problem and solved it by incorporating deep neural network (DNN). In our work, NMF does not play any role in the denoising stage and DNN is used as mapping function from the noisy spectrum to its decorrelated activation vector directly. In this NMF+DNN framework, however, the activation vector H, rather than the noisy spectrum $\mathbf{x}$ (which is used in [21]), is fed as input to DNN. A DNN is then trained to convert H with some overlap into $\tilde{H}$ with less overlap. A similar DNN based mapping strategy was used in [35] to improve the performance of a Automatic Speech Recognition (ASR) system. In contrast to [35], the square error in activation domain, rather than that in log spectrum domain, is set as the training target in this thesis. We expect that this approach is more adequate than [35] because such simple but efficient training target is easier to be modeled by DNN in practice. In addition, Perceptual Evaluation of Speech Quality (PESQ) measure [3] is adopted because our work focuses on the perceptual evaluation of denoised speech signals instead of the word error rate when those signals are input to the ASR system. Moreover, a more efficient NMF front-end module will be experimentally studied and play an important role in our NMF+DNN framework. A detailed diagram is shown in Figure 3.4.

The left side of Figure 3.4 (a) is the pure NMF framework, which is described in Section 3.1, and it estimates activation vector H. $\tilde{H}_s$ and $\tilde{H}_n$ are generated with the corresponding dictionary (W_s and W_n) respectively so that they are theoretically uncorrelated. Labels in the training stage are artificially prepared by concatenating these two activation vectors, $\tilde{H}_s$ and $\tilde{H}_n$. A DNN architecture is then trained to create a mapping from overlapped vector H to the prepared label. Figure 3.4 (b) shows the denoising procedure. The noisy spectrum x is firstly transformed into overlapped activation vector H by NMF front-end process. DNN is then employed to convert H to $\tilde{H}$, which has the less overlap. Then same procedures described in Section 3.1 are carried out to reconstruct the speech signal $\hat{s}(t)$.

This improved technology, which combines NMF and DNN, consistently outperforms the pure NMF-based method and is also much better than the pure DNN-based method in the extremely low SNRs. In addition, our proposed strategy is demonstrated to be more effective compared to the previously proposed method of combining NMF and DNN. The detailed experiments will be described in Chapter 5.



(a) Training stage



(b) Denoising stage

Fig. 3.4 A block diagram of NMF+DNN framework for denoising.

Chapter 4

DNN-based Speech Denoising and Its Compact Implementation

In this chapter, DNN is applied to speech denoising task. We first introduce the different approaches and training configurations of DNN-based methods:

- Different DNN architectures, including Feed-forwarding NN and Long Short-Term Memory (LSTM) recurrent NN, are built respectively to learn a mapping from noisy features to the targets of interest.
- Two training targets, the clean log power spectra (LPS) and the ideal ratio mask (IRM), are evaluated respectively.

In addition, we attempt to compress the neural networks by means of low-rank decomposition. The compact structure with much smaller model size still shows comparable performances to those in the original NN.

4.1 DNN-based Speech Denoising

For speech denoising task, a DNN is used to model the relationship between the clean speech signals and its noisy signals. Specifically, the noisy features are extracted and fed as the input vector. The predicted target features then can be obtained by means of DNN. Usually, both the noisy and target features are represented in the frequency domain. Therefore the output targets, combined with the original phase information, should be processed by inverse Short-Time Fourier Transformation (STFT) to reconstruct the predicted clean waveform.

From Figure 4.1, We can find that a DNN based speech enhancement system consists of three parts, analysis, prediction, and synthesis. In analysis stage, some features are extracted

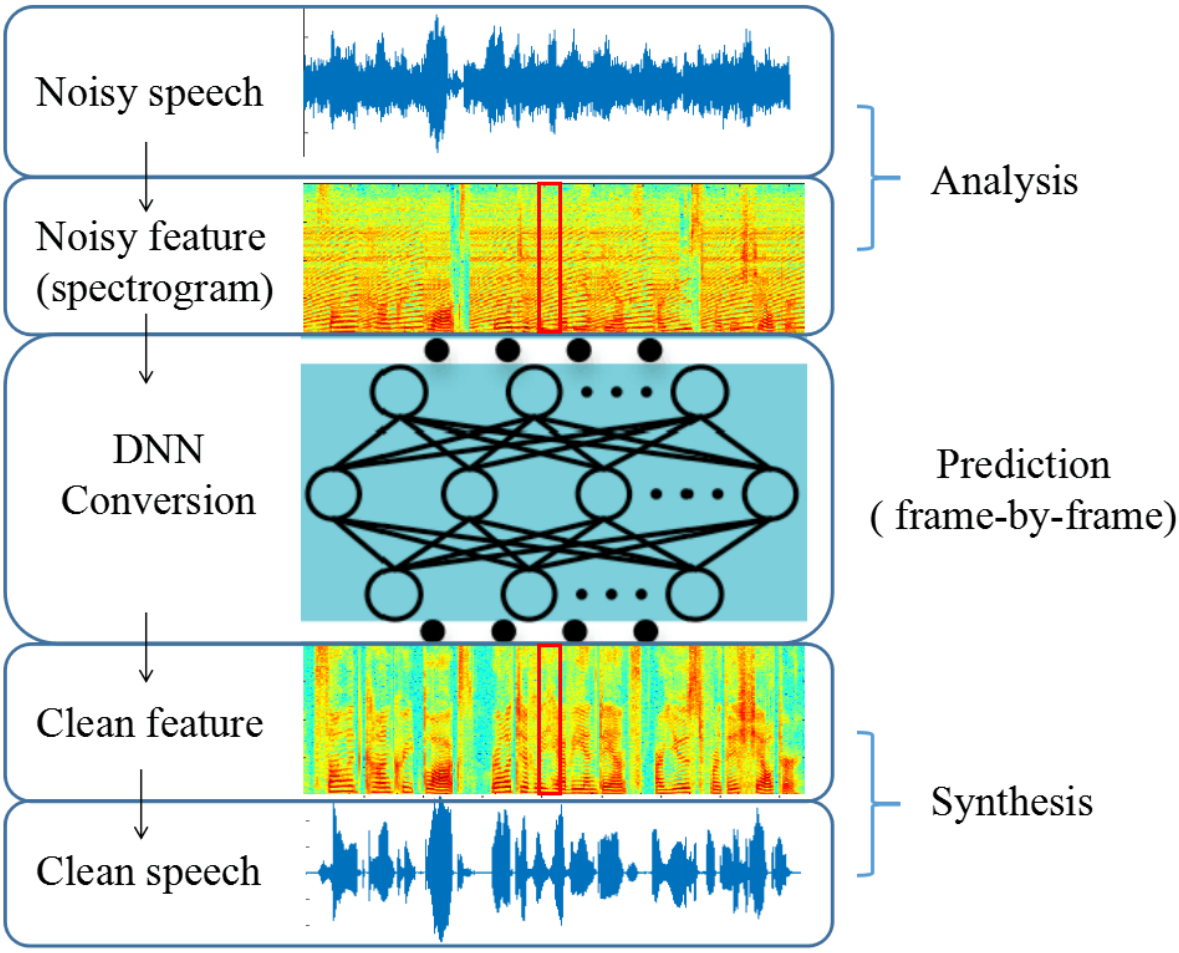


Fig. 4.1 A system diagram of DNN-based speech denoising.

and fed as the input for DNN. In prediction stage, a particular neural network architecture should be built to model the relationship between noisy and clean features. In synthesis stage, we have to firstly decide the desired target, which is the output of DNN. Then the estimated clean speech waveform can be reconstructed based on this output target.

4.1.1 DNN Architecture: Feed-forwarding vs. LSTM

At prediction stage, several DNNs have been proposed for modeling. The most popular types are probably Feed-forwarding NN and LSTM-RNN.

Feed-forward NN has been already introduced in the previous Section 3.2.1. It was adopted in [37] to directly map the noisy features to the clean labels in a frame-to-frame way. The final input vector can be shown as follows.

$$\hat{X}_n = [X_{n-\tau}, \ldots, X_{n-1}, X_n, X_{n+1} \ldots, X_{n+\tau}] \tag{4.1}$$

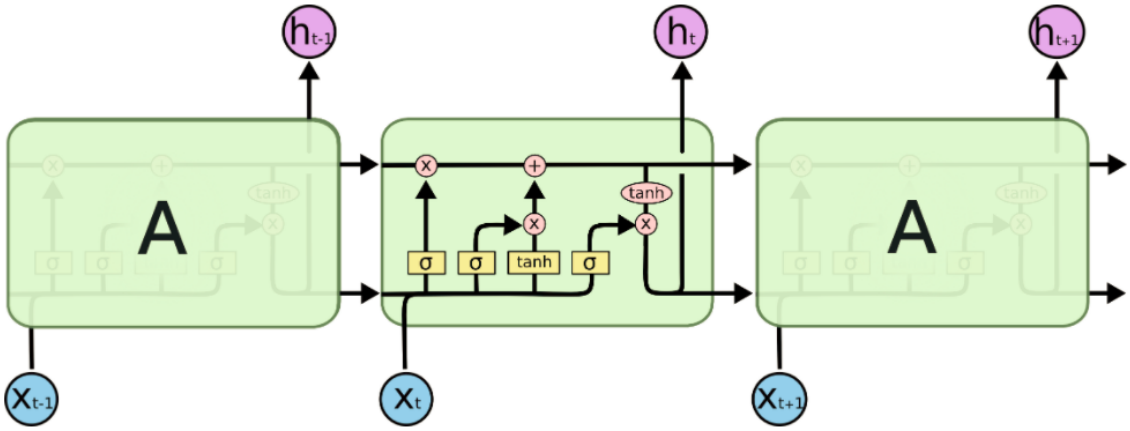


Fig. 4.2 A diagram of LSTM-RNN structure, from [1]

where $\hat{X}_n$ denotes the expanded input feature at n-th frame. That means the input vector consists of the current frame X_n, τ past frames and τ future frames. By expanding the features with the past and future frames, Feed-forward NN explicitly incorporates the context information to improve the denoising performance.

On the other hand, the inherent ability to capture the temporal dynamics of inputs features the Recurrent NN. Particularly, the popular LSTM-RNN which overcomes the vanishing gradient problem of conventional RNNs [17] , is a better choice in practice. LSTM-RNN allows the temporal information to persist through looping the LSTM units. Figure 4.2 gives a diagram of LSTM-RNN. When it is applied to enhance the speech in [36], promising results are achieved.

4.1.2 Training targets: Log Power Spectra vs. Ideal Ratio Mask

At synthesis stage, defining a proper target is crucial for DNN learning. In this Section, we compare two popular targets, Log Power Spectra (LPS) and Ideal Ratio Mask (IRM).

LPS is mathematically defined as the following form:

$$LPS(n,k) = \log(|S(n,k)|^2) \tag{4.2}$$

where $|S(n,k)|$ is the amplitude of STFT of the clean speech signal $s(t)$, n denotes the n-th frame and k denotes the index of frequency bin. Log operation compresses the value range and fits the human auditory perception. As can be seen in Eq. (4.2), LPS target is an unbounded value. By setting LPS as labels, DNN can directly learn the mapping from the noisy feature to the clean spectra.

IRM is shown as Eq. (4.3).

$$IRM(n,k) = \frac{|S(n,k)|^2}{|S(n,k)|^2 + |N(n,k)|^2} \tag{4.3}$$

where $|S(n,k)|$ is the amplitude of STFT of the clean speech signal $s(t)$, $|N(n,k)|$ is the amplitude of STFT of the noise signal. IRM serves as a bound mask, which is very similar to Wiener Filter [24]. The estimated clean spectrum $|\hat{S}(n,k)|$ then can be obtained by the Eq. (4.4).

$$|\hat{S}(n,k)| = IRM(n,k) * |X(n,k)| \tag{4.4}$$

where $|X(n,k)|$ is the amplitude of STFT of the observed noisy signal $x(t)$.

4.2 Compact neural networks

A big DNN model is not of practice for some real-time applications, such as mobile communication. These real-time requirements encourage us to compress the model size without too much performance degradation. Low-rank decomposition has been successfully used in DNN-based speech recognition systems [38]. Thus we propose to apply the same technique to our speech denoising task.

Consider a basic matrix calculation in both DNN training and inference stages. The time complexity of the following Equation (4.5) is $O(mn)$.

$$h_{m\times 1} = W_{m\times n} x_{n\times 1} \tag{4.5}$$

where $h_{m\times 1}$ denotes a m-dimensional output vector. $W_{m\times n}$ denotes a $m \times n$ weight matrix. $x_{n\times 1}$ denotes a n-dimensional input vector. In order to simplify this matrix-vector multiplication, low-rank decomposition is employed. Specifically, Singular Value Decomposition (SVD) is used to decompose the matrix $W_{m\times n}$ to the form shown in Equation (4.6).

$$W_{m\times n} = U_{m\times m} \Sigma_{m\times n} V^T_{n\times n} \tag{4.6}$$

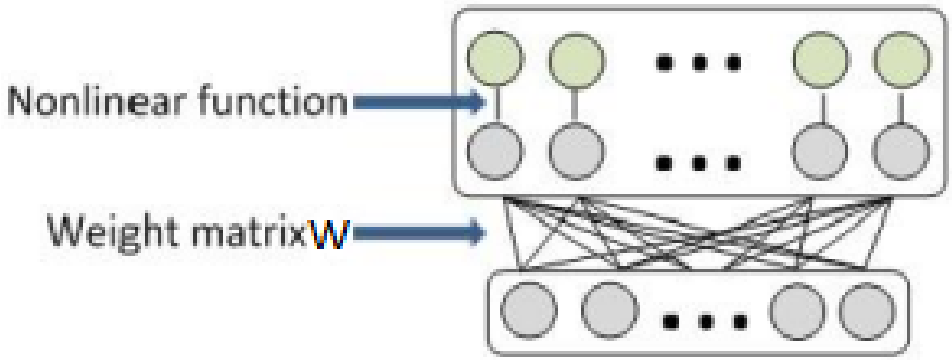
where all diagonal elements in Σ are the singular values of matrix $W_{m\times n}$. Non-diagonal elements in Σ are all zero-value. Singular values are arranged in the decreasing order. By preserving the top k ($k < min\{m,n\}$) singular values of matrix Σ, we can rewrite Equation (4.6) as follows:

$$W_{m\times n} \approx U_{m\times k} \Sigma_{k\times k} V^T_{k\times n} = U_{m\times k} A_{k\times n} \tag{4.7}$$

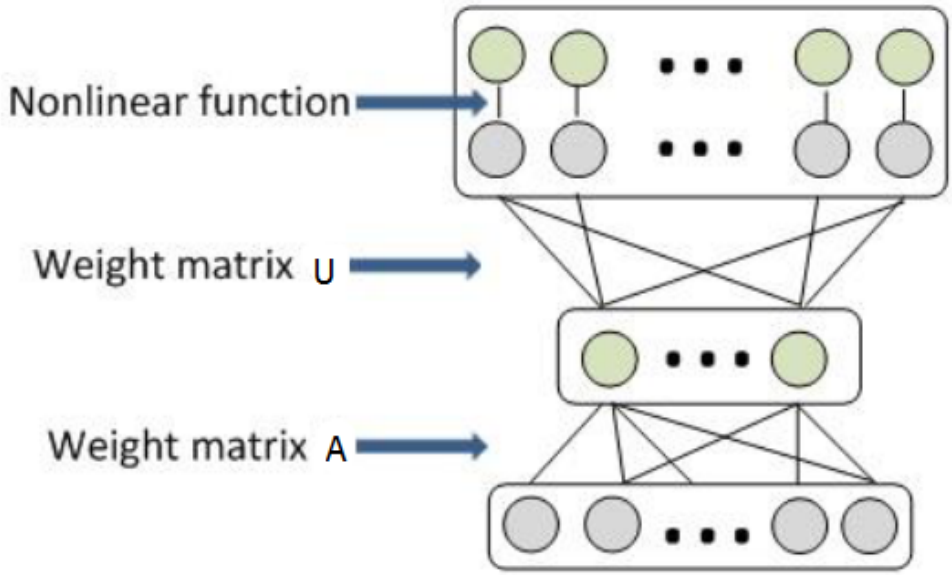
where $A_{k\times n} = \Sigma_{k\times k} V^T_{k\times n}$. Thus the matrix-vector multiplication in Equation (4.5) can be approximately calculated by Equation (4.8), which has the time complexity of $O(mk+nk)$.

$$h_{m\times 1} \approx U_{m\times k} A_{k\times n} x_{n\times 1} \tag{4.8}$$

Figure 4.3 gives the topology of such compact neural network. Figure 4.3 (a) describes how output, $h_{m\times 1}$, is calculated in the form of Equation (4.5) and Figure 4.3 (b) shows the topology of the compact neural network, which is defined as Equation (4.8). DNN model can be reduced if k is much smaller than $min\{m,n\}$, which leads to a simplification in both model parameter number (space complexity) and calculation time (time complexity).



(a) One layer in original DNN model



(b) Two corresponding layers in new DNN model

Fig. 4.3 The topology of compact DNN, from [38].

Chapter 5

Experiments

5.1 Experimental settings

In this Section, we gives the detailed experimental settings for evaluating NMF-based and DNN-based speech denoising systems.

5.1.1 NMF-based approach

Because the NMF model is suitable for modeling a specific noise type, a relatively small dataset is used to build a noise-dependent denoising framework. All 760 utterances from TIMIT (dialect2) database [10] are used as the clean speech corpus. *Factory1 noise*, which is highly non-stationary, are extracted from NOISEX-92 database [33] as the noise corpus. Only 60% part of the noise is used for training, and the other 40% part is used in testing. Both the speech and noise waveforms are re-sampled at 16kHz in our experiments.

At the training stage of NMF framework, the 512-dimensional Hamming window with 75% overlap is adopted for doing STFT. The speech dictionary with 128 bases and the noise dictionary with 64 bases are generated by using speech and noise corpus respectively. Noises are randomly added to the 100 utterances in TIMIT test set at five levels of SNR, i.e., -5dB, 0dB, 5dB, 10dB and 15dB, for test.

5.1.2 DNN-based approach

In our work, a large dataset which includes multiple noise types is used for training a DNN-based speech denoising system in order to improve its noise generalization ability. All 4620 utterances from TIMIT database are used as the clean speech corpus. We collect 112 noise

types, which are extracted from Nonspeech [18] and NOISEX-92 database [33], as noise corpus. Both the speech and noise waveforms are re-sampled at 16kHz in the following experiments.

To generate the noisy speech and its clean label, we randomly add the noise from the noise corpus to the utterances at six levels of SNR, i.e., -5dB, 0dB, 5dB, 10dB, 15dB and 20dB. Finally we generate a 100-hour training corpus. The input feature is the normalized LPS as shown in Eq. (4.2). The 512-dimensional Hamming window with 50% overlap is adopted for doing STFT. Therefore the dimension of the input vector is 257 (= 512/2+1). Three noises extracted from NOISEX-92, including *buccaneer*, *babble* and *facorty1* noises, are used for testing.

5.2 Experimental results

5.2.1 Results of NMF-based method

As is discussed in Chapter 3, we utilize two strategies to improve the performance of standard NMF-based speech denoising system. The detailed results are given in the followings.

Training methods for noise dictionary

In Chapter 3, we have pointed out that NMF algorithm is not suitable to train a noise dictionary. Therefore we conduct an experiment to compare NMF with Kmeans method, to verify which one is better for modeling noise. Besides, the performances with spectral mask given in Equation (3.9) and without are compared. Perceptual Evaluation of Speech Quality (PESQ) score [3] is used to measure the denoising performance. The score ranges from -0.5 to 4.5 and higher scores indicate higher quality. The parameter λ in Equation (3.3) is set to 0 when training speech and noise dictionaries. Table 5.1 presents the detailed experimental results.

The value in Table 5.1 denotes the pure PESQ improvement. We can observe that spectral mask obviously brings a significant improvement in either NMF or Kmeans modeling methods. At each SNR level, a noise dictionary trained by the Kmeans algorithm also outperforms that by the NMF algorithm. According to this result, we adopt Kmeans-based noise dictionary and the spectral mask technique in all the experiments later.

Sparse regularization for speech dictionary

In [30], a proper sparse regularization is proved to make the speech dictionary more representative. An experiment is conducted in our work to select the optimal sparse parameter λ for denoising task. Experimental results are shown in Table 5.2.

Table 5.1 Average PESQ improvements with different training methods for noise dictionaries. Factory1 noise was used.

	NMF method		Kmeans method	
	nomask	mask	nomask	mask
SNR -5	0.170	0.133	0.161	**0.332**
SNR 0	0.106	0.141	0.099	**0.427**
SNR 5	0.037	0.138	0.041	**0.446**
SNR 10	-0.036	0.137	-0.031	**0.412**
SNR 15	-0.123	0.131	-0.124	**0.336**
Ave	0.031	0.136	0.029	**0.391**

Table 5.2 Average PESQ improvements with different sparse parameter λ, under the factory1 noise. (Noise dictionary is trained by Kmeans algorithm, and spectral mask is used in reconstruction.)

	λ=0.0001	λ=0.001	λ=0.01	λ=0.1	λ=1
SNR -5	0.332	0.344	0.367	0.367	**0.390**
SNR 0	0.424	0.431	**0.443**	0.441	0.436
SNR 5	0.443	0.460	0.457	**0.465**	0.437
SNR 10	0.414	0.429	**0.439**	0.438	0.368
SNR 15	0.336	0.346	0.360	**0.371**	0.216
Ave	0.390	0.402	0.413	**0.416**	0.370

Sparse parameter λ in Equation (3.3) allows us to control the trade-off between the Euclidean distance and the sparsity of speech dictionary. According to Table 5.2, optimal performance can be achieved when $\lambda = 0.1$. This parameter setting will be also adopted in the following experiments.

5.2.2 Results of integrated method with NMF and DNN

Several experiments have been designed to demonstrate the effectiveness of the proposed NMF+DNN framework. We prepare the input features and output labels as Figure 3.4 (a) shows. Both feature and label vectors are pre-normalized by dividing the energy themselves,

and the *sigmoid activation function* is used in hidden and output layers. The form of sigmoid function is given as follow:

$$f(x) = \frac{1}{1+e^{-x}} \tag{5.1}$$

One important property is that the output range of this activation function limits from 0 to 1. Then a DNN architecture is built with 4 hidden layers, each of which has 512 nodes. We pretrain the parameters of DNN by using Restricted Boltzmann Machine [4] and apply Dropout technique [15] to avoid the overfitting problem.

In addition, to compare our proposed method with several related methods, we examined another two training targets.

- One is exactly the same as [35], which is denoted as *NMF+DNN(LOG)* because the square error in log spectrum domain works as the cost function.
- The other one, which incorporates the phase information, is denoted as *NMF+DNN(TDE)*, where the term *TDE* means the time domain error. In this TDE method, the estimated spectrum, combined with original noisy phase, is first transformed into speech signal in the time domain, then the square error in the time domain works as the cost function.
- Our approach is denoted as *NMF+DNN(ACT)*, in which the square error in activation domain is set as the training.

All experiment configurations of these three models, except training targets (or cost functions), are set exactly same at DNN training stage for a fair comparison.

Besides, a pure DNN-based speech denoising system is prepared and compared with the proposed NMF+DNN(ACT) model. The dataset used for training this DNN-based system is same with that used for the NMF-based system, which is described in Section 5.1.1. The DNN architecture is exactly same with that used in NMF+DNN framework. Above all is done for the sake of fairness.

Figure 5.1 gives the denoising results of the pure NMF-based, DNN-based, and three different NMF+DNN systems.

After integrating NMF with DNN, all NMF+DNN frameworks outperform the pure NMF framework on average. Among the three NMF+DNN frameworks, we can observe that the performance of TDE is slightly better than that of LOG at low SNRs but worse at high. It is clear that the NMF+DNN(ACT) method achieves the best PESQ performance on all SNR conditions. In addition, the NMF+DNN(ACT) method is much more effective than the pure DNN-base system at the extremely low SNRs (i.e. SNR=-5 and 0 dB). While the performance

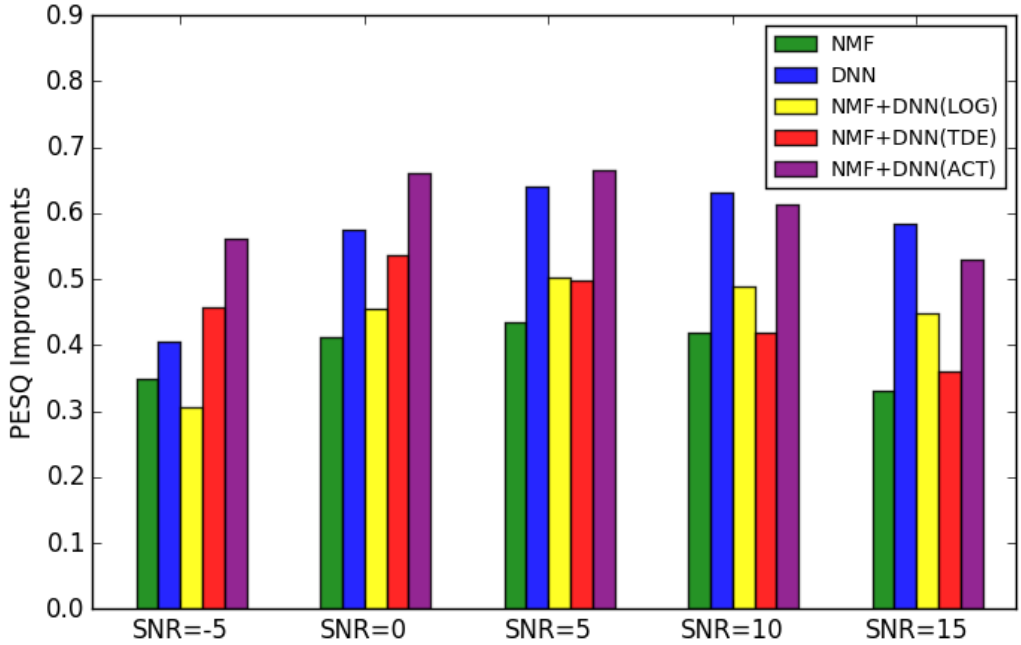


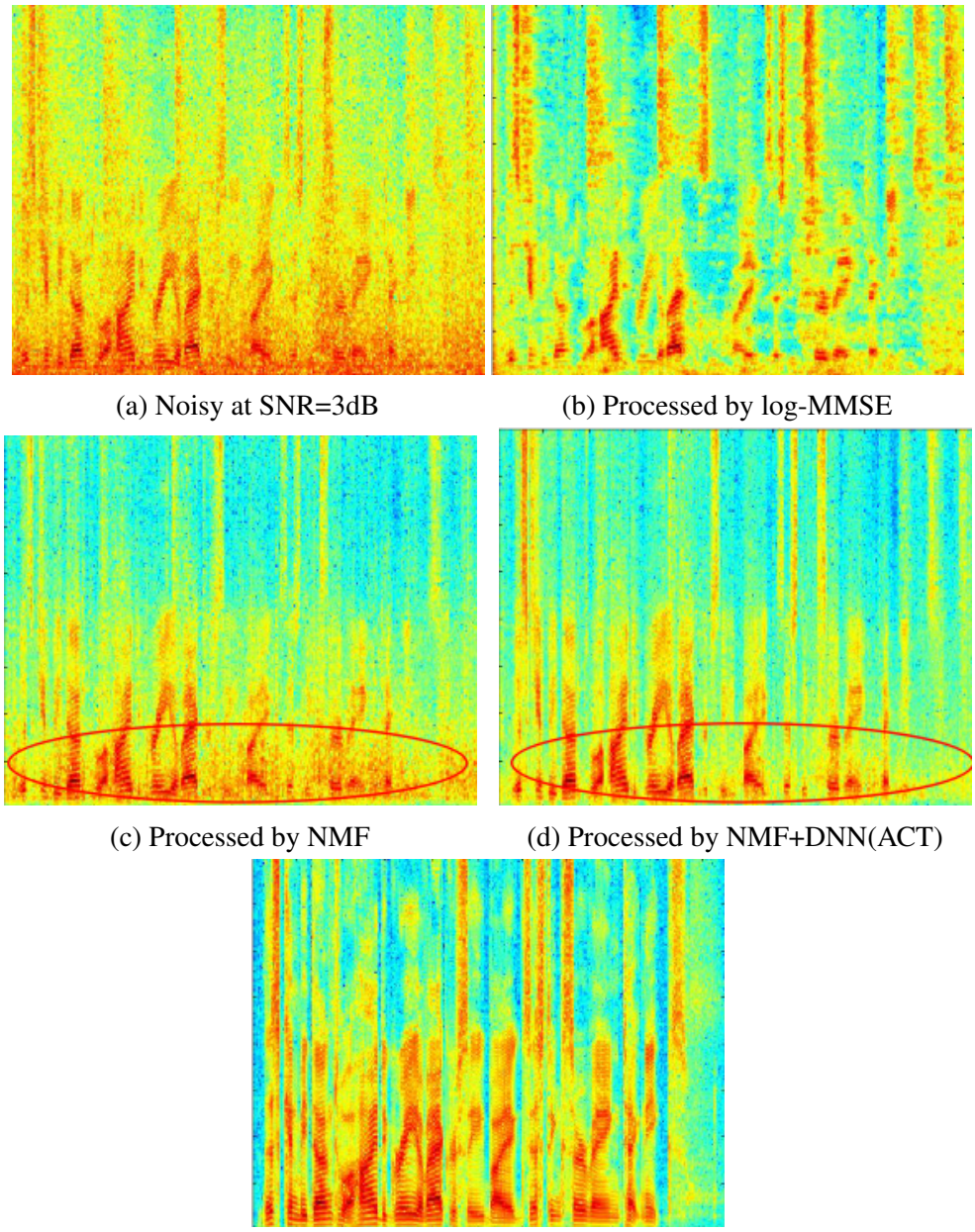
Fig. 5.1 PESQ improvements of the original noisy speech and the estimated speech processed by NMF, DNN and NMF+DNN framework, under the factory1 noise condition.

of it is just slightly better than DNN-based one at SNR=5dB and slightly worse at high SNRs (i.e. SNR=10 and 15 dB).

The spectrograms of the original noisy speech and processed speech are shown in Figure 5.2. A noisy sample speech with factory1 noise at SNR=3dB is processed by three denoising methods, including Log-MMSE, NMF and NMF+DNN(ACT). We can observe that the Log-MMSE method, as a conventional statistical approach, brings much distortion and a bad performance shown in Figure 5.2 (b). Compared with Figure 5.2 (c), low-frequency noise can be effectively suppressed by our proposed method shown in Figure 5.2 (d). It is clear that NMF+DNN(ACT) framework achieves the best performance among above methods.

5.2.3 Investigations of noise generalization ability

So far what we have discussed are all noise-dependent systems. That means all methods, including NMF, NMF+DNN and DNN models, are all built by using only one noise type. It is interesting and valuable to check whether such a noise-dependent system still works well in an unknown noise environment. In this section, we examine the noise generalization abilities of model-based speech denoising approaches.



(a) Noisy at SNR=3dB

(b) Processed by log-MMSE

(c) Processed by NMF

(d) Processed by NMF+DNN(ACT)

(e) Clean

Fig. 5.2 Spectrograms of speech with factory1 noise at SNR=3dB. (a) Noisy speech, (b) Log-MMSE, (c) NMF, (d) NMF+DNN(ACT), (e) Clean speech.

From Figure 5.1, we know that the proposed NMF+DNN(ACT) framework consistently obviously outperforms another two integrated methods. Therefore we only compare three models, NMF, DNN and NMF+DNN(ACT). All of them are trained by using the same dataset described in Section 5.1.1, which includes only factory1 noise. Another two noise types, buccaneer and babble nosies, are used for testing.

Result is given in Figure 5.3. The NMF+DNN system only shows a slightly better result than DNN-based one under the factory1 noise, which is the same noise type used for training. However, when we apply them to denoise the other two noises, the difference of results becomes quite big. We can clearly observe that the performance of DNN-based method degrades seriously under either buccaneer or babble noise, while the performance is still acceptable for NMF+DNN method. In addition, the pure NMF model is not as effective as the pure DNN model for the matched noise type. Yet NMF is proved to be more robust than DNN for the mismatched noises, especially for the babble noise. Thanks to the combination of NMF and DNN, both the robustness of NMF and the strong modeling ability of DNN are inherited. Experimental result indicates that our proposed NMF+DNN method shows not only the best denoising performance, but also the most robust generalization ability to the unknown noises.

Nevertheless, we have to point out that all above comparisons are made on the noise-dependent condition. It is still challenging to train a noise-independent NMF model. That is because the number of noise bases should be big enough to effectively model a large corpus, which contains multiple noise types. While a big number of noise bases will lead to a bad performance and inefficient computation. On the other hand, it is much more easier to improve the noise generalization ability for DNN model. The only thing we need to do is just to feed numerous noise types to train a noise-independent DNN. That is why we still choose to further study the DNN-based method, in the situation that the NMF+DNN method has been proposed. Results will be given in the following sections.

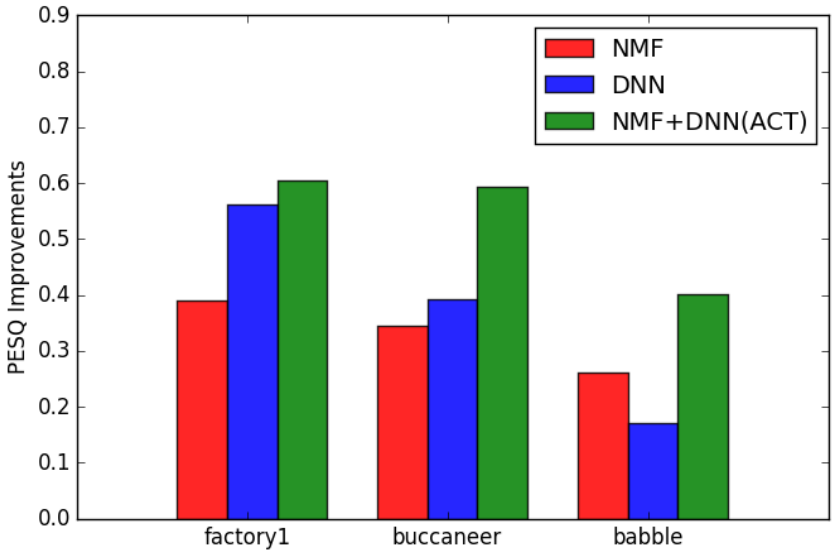


Fig. 5.3 Average PESQ improvements for NMF+DNN(ACT) and DNN systems.

5.2.4 Results of DNN-based method

Instead of serving as a post-processor for NMF module in so-called NMF+DNN framework, DNN in Figure 4.1 is directly used to model the relationship between the clean speech features and its noisy ones.

As described in Section 4.1, different DNN architectures and training targets should be selected for training a supervised speech denoising system. Four systems will be comprehensively evaluated in this thesis. They are:

- Feed-forward neural network with LPS target (FF-LPS).
- Feed-forward neural network with IRM target (FF-IRM).
- LSTM-RNN with LPS target (LSTM-LPS).
- LSTM-RNN with IRM target (LSTM-IRM).

The parameters of DNN structures are given as below:

- For Feed-forward NN, an architecture with 3 hidden layers each of which has 2048 nodes is built. In order to capture the context information, τ in Eq. (4.1) is set to 3.
- For LSTM-RNN, a two-layer LSTMs, each of which has 1024 nodes is adopted, and one dense layer is followed.

The total numbers of parameters of above two structures are almost equal to ensure the comparison is fair. Linear function $f(x) = x$ is used as the activation function for LPS target because LPS is unbounded value, and sigmoid function shown in Equation (5.1) is used for bounded IRM target, the value of which ranges from 0 to 1.

To evaluate the performances in a more comprehensive way, we adopt not only PESQ [3], but also STOI [32], and frequency-weighted Segmental SNR (fwSSNR) [24] as the objective quality measures. PESQ is highly correlated with subjective evaluation scores and ranges from -0.5 to 4.5. STOI is considered as a good measure for the human speech intelligibility and the score ranges from 0 to 1. The higher scores indicate higher performance for all above three measures.

Tables 5.3, 5.4, and 5.5 present the detailed experimental results under PESQ, STOI and fwSSNR measures, with 3 noise types: buccaneer, babble and factory1.

From above results, we can easily observe that the LSTM-IRM system achieves the best performance in all objective measures. Note that in Eq. (4.1), past and future frames are

Table 5.3 Average PESQ scores under different SNRs across 3 noise types.

SNR	NOISY	FF-LPS	FF-IRM	LSTM-LPS	LSTM-IRM
20	2.954	3.405	3.543	**3.639**	3.597
15	2.610	3.225	3.276	**3.398**	3.397
10	2.253	2.991	2.969	3.073	**3.151**
5	1.887	2.677	2.647	2.662	**2.862**
0	1.534	2.254	2.263	2.106	**2.515**
-5	1.227	1.783	1.824	1.558	**2.088**
Ave	2.078	2.723	2.754	2.740	**2.936**

Table 5.4 Average STOI scores under different SNRs across 3 noise types.

SNR	NOISY	FF-LPS	FF-IRM	LSTM-LPS	LSTM-IRM
20	0.968	0.950	0.979	0.977	**0.980**
15	0.927	0.932	0.959	0.958	**0.962**
10	0.856	0.900	0.921	0.922	**0.931**
5	0.756	0.845	0.861	0.850	**0.883**
0	0.640	0.760	0.772	0.727	**0.810**
-5	0.526	0.638	0.650	0.578	**0.701**
Ave	0.779	0.837	0.857	0.835	**0.878**

both incorporated for training Feed-forward NN, while only past information is available for LSTM one. Thanks to the powerful ability to capture the temporal dynamics, however, LSTM architecture is still more effective than Feed-forward one in both LPS and IRM targets.

5.2.5 Results of compact DNN model

In order to compress the neural networks, we adopt SVD described in Section 4.2 and conduct several experiments. The LSTM-IRM system which is used in Section 5.2.4 serves as the baseline since it is considered as the optimal configuration.

Table 5.5 Average fwSSNR scores under different SNRs across 3 noise types.

SNR	NOISY	FF-LPS	FF-IRM	LSTM-LPS	LSTM-IRM
20	15.30	13.21	18.30	17.46	**18.57**
15	11.44	12.27	15.28	15.32	**15.61**
10	8.18	11.10	12.57	13.00	**13.08**
5	5.61	9.66	10.02	10.12	**10.82**
0	3.86	8.02	7.75	7.18	**8.77**
-5	2.81	6.23	5.71	5.10	**6.88**
Ave	7.87	10.08	11.61	11.36	**12.29**

Compress NN by coupling LSTM gates

Before applying SVD to our system, we firstly compress the LSTM by coupling the input gate and the forget gate. Consider the original LSTM formula of the input and forget gates shown below.

$$g^I = \sigma(W^I \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix}) \tag{5.2}$$

$$g^F = \sigma(W^F \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix}) \tag{5.3}$$

where σ denotes the *sigmoid* function. W^I and W^F are the corresponding weight matrices. g^I and g^F are the input and forget gates. h_{t-1} is the recurrent hidden variable and x_t is the current input variable. Research [13] has shown coupling these two gates does not significantly impair performance. As [13] suggests, the forget gate can be simply represented in the form of Eq. (5.4)

$$g^F = 1 - g^I \tag{5.4}$$

We trained such simplified LSTM (denoted as Coupled) and compared it with the original LSTM (denoted as Full). Table 5.6 gives the performance comparison between the Coupled and Full LSTMs.

Table 5.6 Average Performance Comparison between Full and Coupled LSTMs, under six SNR levels across 3 noise types.

Model	Average Performance			Model Size
	PESQ	STOI	fwSSNR	
Full	2.936	0.878	12.29	49 MB
Coupled	2.920	0.872	12.20	37 MB

From Table 5.6, nearly 25% parameters can be reduced by this simple coupling strategy, at the cost of only small performance degradation of coupled LSTM. In the following experiments, we apply SVD to further compress this coupled LSTM model.

Compress NN by applying SVD

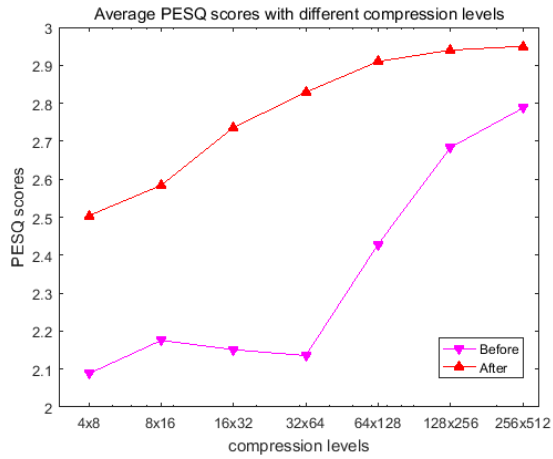
As described in Section 5.1.2, the dimension of input vector is 257 and the number of LSTM hidden nodes is 1024. Thus the big weight matrix W^1 (of coupled LSTM model) in the first layer is in the size of **3072** (1024 $\times$ 3) $\times$ **1281** (1024+257), and the matrix W^2 in the second layer is **3072** (1024 $\times$ 3) $\times$ **2048** (1024+1024). To do low-rank approximation, we only preserve the top k singular values. For example, if the configuration is set as 64×128, the matrix W_1 and W_2 then can be approximately reconstructed as below:

$$\hat{W}^1_{3072\times 1281} = U^1_{3072\times 64} A^1_{64\times 1281} \tag{5.5}$$

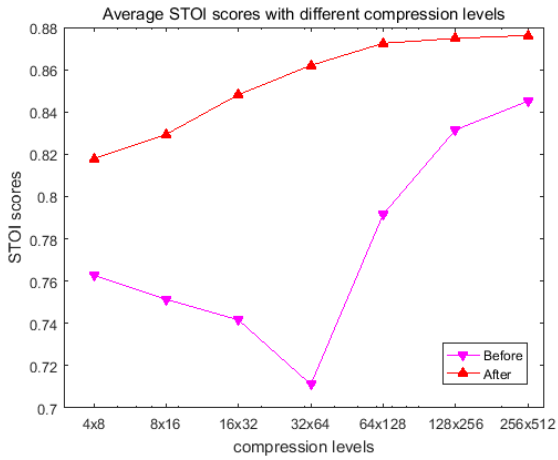
$$\hat{W}^2_{3072\times 2048} = U^2_{3072\times 128} A^2_{128\times 2048} \tag{5.6}$$

where A is defined in Eq. (4.7).

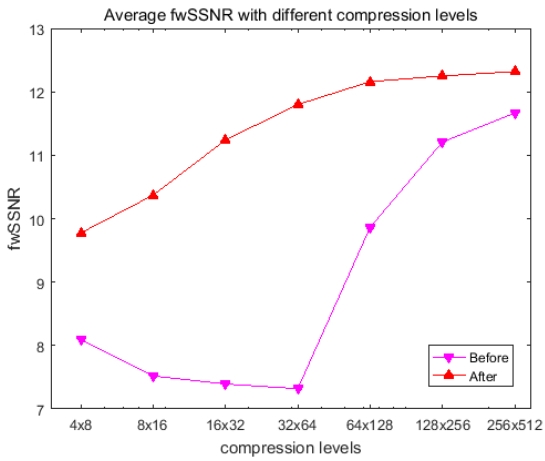
It is reasonable that the performance degradation will be serious if we compress the neural networks too heavily. A solution is to fine-tuning the networks, as was proposed in [38]. Figure 5.4 presents the average performances of models at different compression levels, where *Before*, *After* denote the networks without and with fine-tuning respectively. This figure shows that the denoising performances in all three measures can be significantly improved after fine-tuning procedure. For those models with fine-tuning (red line), the heavier the compression is, the lower the scores will be. We found that the configuration 64×128 is a *particular point*. When the compression level is slighter than that of this point, the performance scores remain roughly stable. While when the compression level is heavier, the scores drop dramatically.



(a) PESQ results



(b) STOI results

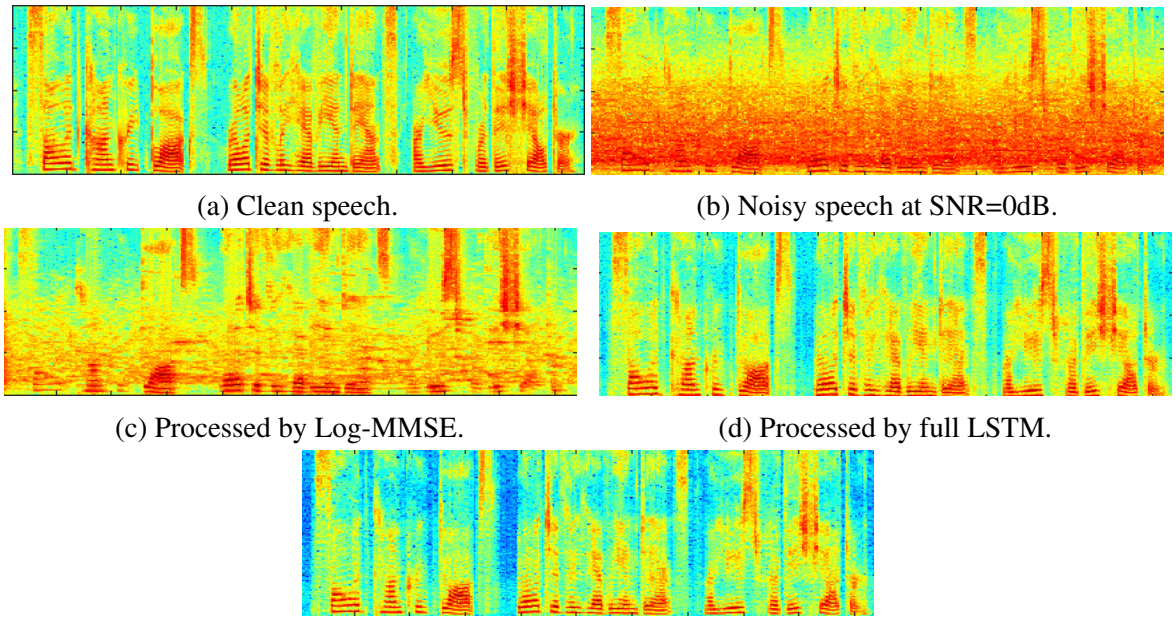


(c) FwSSNR results

Fig. 5.4 Average performances of models at different compression levels, under six SNR levels across 3 noise types

Table 5.7 gives the more detailed results. According to this table, we can observe that the compact model works well even the model is aggressively compressed. Compare the 64×128 network with the original full LSTM, the model size is relatively reduced by 91.2 %. But all the measure scores show that such light model is still quite comparable.

Last, we compare the DNN-based method with the conventional Log-MMSE method. From Figure 5.5, we can observe that the Log-MMSE method gives relatively bad performance shown in Figure 5.5 (c). By a comparison between Figure 5.5 (e) and Figure 5.5 (d), both compact and full DNN model achieve the satisfying denoising results.



(a) Clean speech.

(b) Noisy speech at SNR=0dB.

(c) Processed by Log-MMSE.

(d) Processed by full LSTM.

(e) Processed by compact(64×128) LSTM.

Fig. 5.5 Spectrograms of speech with babble noise at SNR=0dB. (a) Clean speech, (b) Noisy, (c) Log-MMSE, (d) Full Model, (e) Compact Model.

Table 5.7 Average Performance with several compact models, under six SNR levels across 3 noise types.

Model		Results			Model Size
		PESQ	STOI	fwSSNR	
No Process		2.078	0.779	7.87	None (0 MB)
Full		2.936	0.878	12.29	49 MB
Coupled		2.920	0.872	12.20	37 MB
16×32	Before	2.151	0.742	7.393	1.8 MB
	After	2.736	0.848	11.24	
64×128	Before	2.428	0.791	9.863	4.3 MB
	After	2.910	0.872	12.16	
256×512	Before	2.787	0.845	11.66	14 MB
	After	2.950	0.876	12.32	

Chapter 6

Conclusions and Future Works

6.1 Conclusions

In this thesis, we mainly present the model-based monaural speech denoising approaches.

First, we discuss the NMF-based approach in the noise-dependent situation. We show that the performance of NMF-based approach can be improved by adopting sparse training for speech dictionary and clustering-based training for noise dictionary. For a further improvement, we integrate NMF with DNN. We experimentally compare the NMF-based, DNN-based, and the integrated NMF+DNN methods. Results indicate that our proposed NMF+DNN(ACT) method achieves the best denoising performance and the most robust noise generalization ability for unknown noises, in the measure of PESQ score.

Second, we investigate the DNN-based approach in the noise-independent situation. Different DNN architectures and training targets are compared and finally the LSTM-IRM configuration is demonstrated to be the most effective one. Moreover, we compress the original complex DNN model by low-rank approximation. Experimental results show that the compact model, which has much smaller complexity, achieves comparably good performance in the objective measure.

6.2 Future works

In future studies, we are going to:

- Explore the noise-adaptive framework for NMF-based approach. We would try to solve this problem by using a semi-supervised NMF algorithm [26].

- Apply post-filtering techniques [20, 6] to avoid the speech over-smoothing phenomenon, which especially occurred in the high frequency band, for both NMF-based or DNN-based approaches.
- Further compress the model by quantifying the float value with lower bits, instead of the default 32 bits.

References

[1] Understanding lstm networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[2] Michal Aharon, Michael Elad, and Alfred M Bruckstein. K-svd and its non-negative variant for dictionary design. In *Wavelets XI*, volume 5914, page 591411. International Society for Optics and Photonics, 2005.

[3] J. G. Beerends, A. P. Hekstra, A. W. Rix, and M. P. Hollier. Perceptual evaluation of speech quality (pesq) the new itu standard for end-to-end speech quality assessment part ii: psychoacoustic model. *Journal of the Audio Engineering Society*, 50(10):765–778, 2002.

[4] Y. Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[5] S. F. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans.acoust.speech & Signal Process*, 27(2):113–120, 1979.

[6] Juin-Hwey Chen and Allen Gersho. Adaptive postfiltering for quality enhancement of coded speech. *IEEE Transactions on Speech and Audio Processing*, 3(1):59–71, 1995.

[7] J Eggert and E Korner. Sparse coding and nmf. In *IEEE International Joint Conference on Neural Networks, 2004. Proceedings*, pages 2529–2533 vol.4, 2004.

[8] Yariv Ephraim and David Malah. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *IEEE Transactions on acoustics, speech, and signal processing*, 32(6):1109–1121, 1984.

[9] Yariv Ephraim and David Malah. Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. *IEEE transactions on acoustics, speech, and signal processing*, 33(2):443–445, 1985.

[10] John S Garofolo et al. Getting started with the darpa timit cd-rom: An acoustic phonetic continuous speech database. *National Institute of Standards and Technology (NIST), Gaithersburgh, MD*, 107:16, 1988.

[11] Jort F Gemmeke, Tuomas Virtanen, and Antti Hurmalainen. Exemplar-based sparse representations for noise robust automatic speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):2067–2080, 2011.

[12] Emad M Grais and Hakan Erdogan. Single channel speech music separation using nonnegative matrix factorization and spectral masks. In *Digital Signal Processing (DSP), 2011 17th International Conference on*, pages 1–6. IEEE, 2011.

[13] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2017.

[14] JA Haigh and JS Mason. Robust voice activity detection using cepstral features. In *TENCON'93. Proceedings. Computer, Communication, Control and Power Engineering. 1993 IEEE Region 10 Conference on*, volume 3, pages 321–324. IEEE, 1993.

[15] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[16] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[18] G. Hu. 100 nonspeech environmental sounds. http://www.cse.ohio-state.edu/pnl/corpus/HuCorpus.html, 2004.

[19] Sunil Kamath and Philipos Loizou. A multi-band spectral subtraction method for enhancing speech corrupted by colored noise. In *ICASSP*, volume 4, pages 44164–44164. Citeseer, 2002.

[20] Takuhiro Kaneko, Hirokazu Kameoka, Nobukatsu Hojo, Yusuke Ijima, Kaoru Hiramatsu, and Kunio Kashino. Generative adversarial network-based postfilter for statistical parametric speech synthesis. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 4910–4914. IEEE, 2017.

[21] Tae Gyoon Kang, Kisoo Kwon, Jong Won Shin, and Nam Soo Kim. Nmf-based speech enhancement incorporating deep neural network. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[22] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE transactions on pattern analysis and machine intelligence*, 24(7): 881–892, 2002.

[23] Philip Lockwood and Jérôme Boudy. Experiments with a nonlinear spectral subtractor (nss), hidden markov models and the projection, for robust speech recognition in cars. *Speech communication*, 11(2-3):215–228, 1992.

[24] Philipos C Loizou. *Speech enhancement: theory and practice*. CRC press, 2007.

[25] Yang Lu and Philipos C Loizou. Speech enhancement by combining statistical estimators of speech and noise. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 4754–4757. IEEE, 2010.

[26] Y. Luan, D. Saito, Y. Kashiwagi, N. Minematsu, and K. Hirose. Semi-supervised noise dictionary adaptation for exemplar-based noise robust speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 1745–1748. IEEE, 2014.

[27] Kuldip K Paliwal and Leigh D Alsteris. On the usefulness of stft phase spectrum in human listening tests. *Speech Communication*, 45(2):153–170, 2005.

[28] Sundarrajan Rangachari and Philipos C Loizou. A noise-estimation algorithm for highly non-stationary environments. *Speech communication*, 48(2):220–231, 2006.

[29] M. N. Schmidt, J. Larsen, and F. T. Hsiao. Wind noise reduction using non-negative sparse coding. In *Machine Learning for Signal Processing, 2007 IEEE Workshop on*, pages 431–436, 2007.

[30] Mikkel N Schmidt and Rasmus K Olsson. Single-channel speech separation using sparse non-negative matrix factorization. In *Ninth International Conference on Spoken Language Processing*, 2006.

[31] Volker Stahl, Alexander Fischer, and Rolf Bippus. Quantile based noise estimation for spectral subtraction and wiener filtering. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, volume 3, pages 1875–1878. IEEE, 2000.

[32] Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):2125–2136, 2011.

[33] Andrew Varga and Herman JM Steeneken. Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech communication*, 12(3):247–251, 1993.

[34] Tuomas Virtanen and Ali Taylan Cemgil. Mixtures of gamma priors for non-negative matrix factorization based speech separation. In *International Conference on Independent Component Analysis and Signal Separation*, pages 646–653. Springer, 2009.

[35] T. T. Vu, B. Bigot, and E. S. Chng. Combining non-negative matrix factorization and deep neural networks for speech enhancement and automatic speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 499–503. IEEE, 2016.

[36] Felix Weninger, Hakan Erdogan, Shinji Watanabe, Emmanuel Vincent, Jonathan Le Roux, John R Hershey, and Björn Schuller. Speech enhancement with lstm recurrent neural networks and its application to noise-robust asr. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 91–99. Springer, 2015.

[37] Y. Xu, J. Du, L. R. Dai, and C. H. Lee. A regression approach to speech enhancement based on deep neural networks. *IEEE/ACM Transactions on Audio Speech & Language Processing*, 23(1):7–19, 2015.

[38] Jian Xue, Jinyu Li, and Yifan Gong. Restructuring of deep neural network acoustic models with singular value decomposition. In *Interspeech*, pages 2365–2369, 2013.

Appendix A

Publications

National conferences and meetings

- Haoyu Li, Daisuke Saito and Nobuaki Minematsu, “Single Channel Speech Denoising by Integrating NMF with Deep Neural Network”. In 情報処理学会音声言語情報処理研究会資料, 2017-SLP-117(10), pp. 1-5 (2017-7).
- Haoyu Li, Daisuke Saito and Nobuaki Minematsu, “An Experimental Study on Deep Neural Network based Speech Enhancement and Its Compact Implementation”. In 日本音響学会講演論文集, pp. 93-96, 2018-3.