Deep Learning Approach for Robust Voice Activity Detection and Speech Enhancement

잡음에 강인한 음성 구간 검출과 음성 향상을 위한 딥 러닝 기반 기법 연구

2017년 2월

Tae Gyoon Kang

Department of Electrical and Computer Engineering

Seoul National University

Abstract

Over the past decades, a number of approaches have been proposed to improve the performances of voice activity detection (VAD) and speech enhancement algorithms which are crucial for speech communication and speech signal processing systems. In particular, the increasing use of machine learning-based techniques has led to the more robust algorithms in low SNR conditions. Among them, the deep neural network (DNN) has been one of the most popular techniques.

While the DNN-based technique is successfully applied to these tasks, the characteristics of VAD and speech enhancement tasks are not fully incorporated to the DNN structures and objective functions. In this thesis, we propose the novel training schemes and post-filter for DNN-based VAD and speech enhancement. Unlike algorithms with basic DNN-based framework, the proposed algorithm combines the knowledge from signal processing and machine learning society to develop the improve DNN-based VAD and speech enhancement algorithm. In the following chapters, the environmental mismatch problem in the VAD area is compensated by applying multi-task learning to the DNN-based VAD. Also, the DNN-based framework is proposed in the speech enhancement scenario and the novel objective function and post-filter which are derived from the characteristics on human auditory perception improve the DNN-based speech enhancement algorithm.

In the VAD task, the DNN-based algorithm was recently proposed and outperformed the traditional and other machine learning-based VAD algorithms. However, the performance of the DNN-based algorithm sometimes deteriorates when the training and test environments are not matched with each other. In order to increase the performance of the DNN-based VAD in unseen environments, we adopt the multi-task learning (MTL) framework which consists of the primary VAD and subsidiary feature enhancement tasks. By employing the MTL framework, the DNN learns the denoising function in the shared hidden layers that is useful to maintain the VAD performance in mismatched noise conditions.

Second, the DNN-based framework is applied to the speech enhancement by considering it as a regression task. The encoding vector of the conventional nonnegative matrix factorization (NMF)-based algorithm is estimated by the proposed DNN and the performance of the DNN-based algorithm is compared to the conventional NMF-based algorithm.

Third, the perceptually motivated objective function is proposed for the DNN-based speech enhancement. In the proposed technique, a new objective function which consists of the Mel-scale weighted mean square error, temporal and spectral variations similarities between the enhanced and clean speech is employed in the DNN training stage. The proposed objective function helps to compute the gradients based on a perceptually motivated non-linear frequency scale and alleviates the over-smoothness of the estimated speech.

Furthermore, the post-filter which adjusts the variance over frequency bins further compensates the lack of contrasts between spectral peaks and valleys in the enhanced speech. The conventional GV equalization post-filters do not consider the spectral dynamics over frequency bins. To consider the contrast between spectral

peaks and valleys in each enhanced speech frames, the proposed algorithm matches the variance over coefficients in the log-power spectra domain.

Finally, in the speech enhancement task, an integrated technique using the proposed perceptually motivated objective function and the post-filter is described. In matched and mismatched noise conditions, the performance results of the conventional and proposed algorithm are discussed. Also, the subjective preference test result of these algorithms is also provided.

Keywords: Voice activity detection, speech enhancement, deep learning, deep neural network (DNN), noise suppression, multi-task learning, objective function, weighted mean square error, temporal and spectral variation similarities, variance compensation post-filter.

Student number: 2012-30189

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recently, voice activity detection (VAD) and speech enhancement algorithms are playing important roles in many speech signal processing and communication systems. Since voice presence interval estimation and speech enhancement are closely related, many studies proposed the algorithms which solve both the VAD and speech enhancement. While the considerable performance improvements have been achieved by various approaches, VAD and speech enhancement in realistic noise environments still remain the challenging problems.

Early studies on this area are mostly based on the minimum mean-square error (MMSE) criterion [1] which tracks the noise statistics over frames to estimate the speech presence probability and the clean speech spectrum. In this approach, the speech and noise powers are modeled by the Gaussian distribution and MMSE estimator are derived from *a priori* and *a posteriori* signal-to-noise ratios (SNRs). In these algorithms, the generalized likelihood ratio between the speech presence and absence hypotheses which can decide the voice intervals from the noisy speech utterances is also obtained [2].

While the algorithms based on this approach can estimate the voice presence intervals and clean speech waveform with affordable computational cost, they have difficulties in tracking non-stationary noises which causes the speech quality degradation in real-world applications. Though the noise tracking performance in these environments can be improved by adopting the minima controlled recursive averaging noise estimation [3], [4], these algorithms still show poor performance in impulsive or speech-like noise environments.

Instead of tracking the noise statistics estimated from a noisy speech utterance, recent studies focus on adopting machine learning-based algorithms which learn the spectral characteristics of speech and noise from a huge amount of exemplars. Among them, the non-negative matrix factorization (NMF) which approximates a non-negative input matrix as a product of a basis matrix and an encoding matrix with non-negative elements is one of most well-known algorithms [5], [6]. The NMF-based algorithms can separate the speech components from the noisy speech mixture in non-stationary environments more easily since the NMF algorithm does not need to track the noise statistics. However, the NMF algorithm assumes that the speech and noise subspaces are almost orthogonal to each other. In many real-world applications where the speech and noise subspaces often overlap, the quality of the enhanced speech from NMF-based algorithms are usually deteriorated.

In this thesis, motivated by the success of deep neural networks (DNNs) in speech recognition area, we adopt the deep learning approach to the VAD and speech enhancement tasks. Compared to the conventional statistical model and machine learning-based algorithms, the DNN-based algorithm can easily learn nonlinear relationship between the input and output features. Also, a large amount of parameters of the DNN enable to learn speech and noise subspaces that cannot be represented by a few parameters. The proposed DNN-based VAD and speech enhancement algorithms are further improved by adopting various techniques that are based on the knowledge from speech signal processing society.

In Chapter 4, we propose a DNN-based VAD algorithm which adopts the multi-task learning (MTL) framework. In the MTL framework, the generalization power of DNN is improved by sharing hidden layers of the DNNs for multiple related tasks [7]–[10]. In the proposed algorithm, the main task of VAD is jointly trained with a subsidiary task of feature enhancement. By sharing the hidden layer with the feature enhancement task, the shared hidden layer of the DNN would learn the denoising functions of the input feature while the they can still be used to estimate speech activity status. The performance improvement of the DNN-based VAD with MTL framework has been confirmed by the experiment on Aurora2 database.

In Chapter 5, we propose a DNN-based technique to estimate the encoding vectors of the NMF framework. Unlike the previous NMF-based speech enhancement algorithms, the proposed algorithm can deal with the source subspace overlap problem since the DNN can learn the complicated relationship between speech and noise that cannot be fully represented linearly. The mapping between the noisy speech vectors and the corresponding speech and noise encoding vectors is modeled with a DNN for which the training database is artificially generated by mixing the speech and noise database for various signal-to-noise ratios (SNRs). To show the performance of the proposed algorithm, we measured the speech quality of the enhanced speech which is generated after Wiener-filter like gain function is applied to the noisy speech. The performance of the proposed algorithm was evaluated in various matched and mismatched noise conditions and better results were observed compared to the conventional NMF-based algorithms.

In Chapter 6, we propose a novel DNN-based speech enhancement algorithm which is trained to minimize objective function that consists of the Mel-scale weighted mean square error, temporal and spectral variations between the enhanced and clean speech. The Mel-scale frequency scale is adopted to the DNN-based enhancement to emphasize the frequency bands which heavily affects the human auditory perception and intelligibility than other frequency bands. Also, the short-time temporal variation of the one-third octave bands and the spectral variation over the magnitude spectrum are also considered in the training stage to compensate the lack of variation of the DNN estimate trajectories. The effect of these sub-cost terms in the objective function were analyzed in the experiment on matched and mismatched noise environments.

In Chapter 7, the contrast between spectral peaks and valleys of the enhanced speech are further improved by a spectral variance (SV) equalization post-filter. Conventional frequency-dependent and -independent GV algorithms partly alleviate the over-smoothing problem of the DNN with GV factors obtained from the distribution of the output features. However, these GV factors do not consider the spectral dynamics over frequency bins. To consider the contrast between spectral peaks and valleys in enhanced speech frames, the proposed algorithm matches the variance over spectral coefficients of the enhanced speech in the log-power spectra domain to that of the clean speech. In the experiment, the perceptual quality of the enhanced speech using the algorithms proposed in Chapter 6is described in both the objective and subjective tests.

The rest of the thesis is organized as follows: The next chapter introduces the conventional speech enhancement and the Chapter 3 introduces the basic structure of the DNN. In Chapter 4, a DNN-based VAD algorithm with MTL framework is proposed. In Chapter 5, a hybrid DNN-NMF algorithm in which the DNN estimates the encoding vector of the NMF framework is proposed. In Chapter 6, a novel objective function which incorporates the non-linear frequency scale, temporal and spectral variation is introduced. Finally, a post-filter for the DNN-based speech enhancement algorithm is discussed in Chapter 7. The conclusions are drawn in Chapter 8.

Chapter 2

Conventional Approaches for Speech Enhancement

2.1 NMF-Based Speech Enhancement

In the NMF analysis, an $l_F \times l_T$ dimensional matrix V is described by the product of W and H as follows:

$$V \approx WH \tag{2.1}$$

where W is a nonnegative $l_F \times l_H$ dimensional matrix and H is a nonnegative $l_H \times l_T$ dimensional matrix. In this thesis, W used for the NMF analysis is denoted as a basis matrix and H is denoted as an encoding matrix.

In the NMF analysis, W and H are iteratively updated while minimizing the objective function $C(V|WH)$ which measures the distance between an input matrix V and a multiplication of the basis and encoding matrices WH. The Euclidean distance and KL divergence can be examples of the objective function. At each iteration of the multiplicative rule for Euclidean distance, W and H are updated as follows [11]:

$$H \leftarrow H \quad \otimes \quad \frac{W^\dagger V}{W^\dagger W H} \tag{2.2}$$

$$W \leftarrow W \quad \otimes \quad \frac{V H^\dagger}{W H H^\dagger} \tag{2.3}$$

where $\otimes$ and $\frac{(\ )}{(\ )}$ denote element-wise multiplication and division operations between the matrices or vectors, and † denotes the transpose of a matrix or a vector.

In the case of the KL divergence, the basis and encoding matrices are updated as

$$H(j,k) \leftarrow H(j,k) \frac{\sum_i W(i,j) V(i,k) / (WH)(i,k)}{\sum_m W(m,j)} \tag{2.4}$$

$$W(i,j) \leftarrow W(i,j) \frac{\sum_k H(j,k) V(i,k) / (WH)(i,k)}{\sum_v H(j,v)} \tag{2.5}$$

where $H(j,k)$ and $W(i,j)$ are the jk-th and ij-th elements of H and W, respectively. In the source separation task, the KL divergence is more popularly used than the Euclidean distance.

Through the NMF analysis, the basis matrix W describes the characteristics of the data matrix V. Then, for given data vectors, the encoding vectors are derived and used many tasks, i.e. unsupervised clustering or data compression.

In the NMF analysis, each column of W denotes a basis vector for V. Since each column of H represents the corresponding column of V with a weighted sum of basis vectors in V independently, the basis matrix from the NMF algorithm can effectively represent the non-stationary characteristics of a data matrix. By applying NMF algorithm into speech enhancement task, the speech and noise parameters can be obtained without the stationary noise assumption.

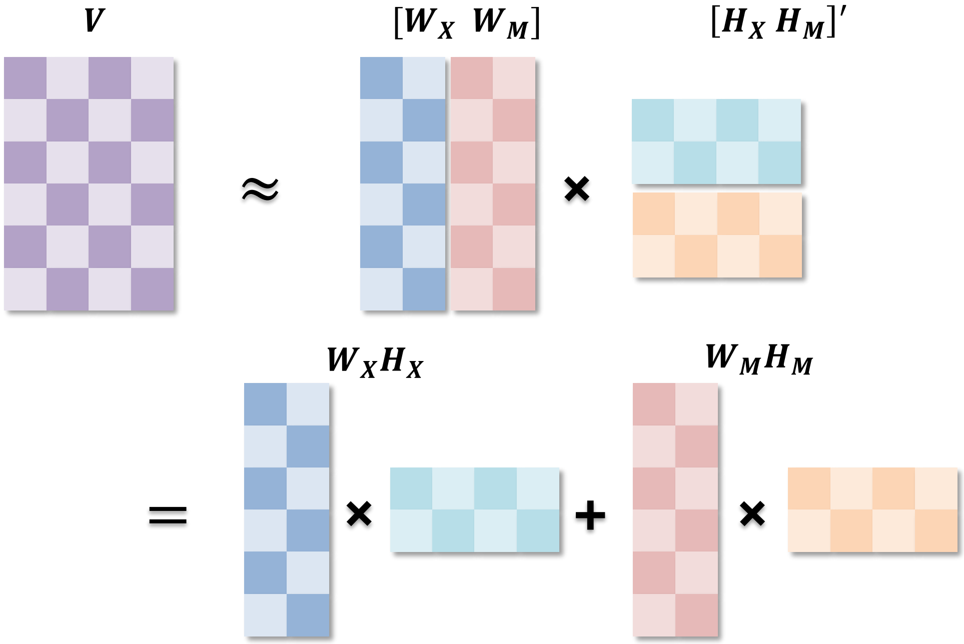

Figure 2.1: Scheme of the NMF-based speech enhancement.

Since the basis vectors of the NMF analysis can extract the unique characteristics of the acoustic sources from the given source spectrogram, this approach can be easily applied into the speech enhancement area. Figure 2.1. shows the scheme of the NMF-based speech enhancement algorithm. When we assume that the subspace of the speech and noise basis matrices W_X, W_M are orthogonal to each other, we can estimate the speech and noise encoding basis matrices H_X, H_M without ambiguity of the mixed sources from the mixture spectrogram V. The speech and noise sources can respectively generated by the multiplication of the basis matrices and encoding matrices. Since the NMF analysis do not rely on the stationary noise assumption, the algorithms with the NMF analysis can be widely applied to realistic noise environments.

Speech enhancement using NMF algorithm consists of training and test stages. In the training stage, the speech and noise spectral magnitude from training database are analyzed by the NMF algorithm. Let us denote an $l_F \times l_X$ dimensional speech spectral magnitude matrix as X and an $l_F \times l_M$ dimensional noise spectral magnitude matrix as M where l_F denotes the number of frequency bins and l_X, l_M denotes the number of speech and noise frames, respectively. The NMF algorithm finds the speech and noise models $\{W_X, H_X\}$ and $\{W_M, H_M\}$ from X and M respectively through an iterative procedure such as (2.4) and (2.5).

In the test stage, each noisy speech spectral magnitude vector $\mathbf{v}$ is separated into speech and noise components by NMF algorithm with a concatenated basis matrix of W_X and W_M [6], [12], [13]. The concatenated basis matrix W for speech and noise is formed by a simple concatenation as follows:

$$W = [W_X \quad W_M]\,. \tag{2.6}$$

By applying (2.4) iteratively with fixed W, a corresponding encoding data vector $\mathbf{h}(\mathbf{v})$ in which each element has information for each speech or noise basis from $\mathbf{v}$ is obtained. From W and $\mathbf{h}(\mathbf{v})$, $\mathbf{v}$ can be factorized as

$$\begin{aligned} \mathbf{v} &= W\mathbf{h}(\mathbf{v}) && (2.7) \\ &= [W_X \quad W_M] \begin{bmatrix} \mathbf{h}_X(\mathbf{v}) \\ \mathbf{h}_M(\mathbf{v}) \end{bmatrix} && (2.8) \end{aligned}$$

where $\mathbf{h}_X(\mathbf{v})$ and $\mathbf{h}_M(\mathbf{v})$ are encoding vectors for speech and noise basis matrices which are derived by applying (2.4) iteratively.

The estimated spectral magnitudes of speech and noise can be obtained by multiplying each basis matrix and the corresponding part of encoding vector, i.e., $W_X\mathbf{h}_X(\mathbf{v})$ and $W_M\mathbf{h}_M(\mathbf{v})$. Instead of using $W_X\mathbf{h}_X(\mathbf{v})$ as the estimated speech spectral magnitude $\mathbf{x}$ directly, the gain function similar to Wiener filter is usually applied to increase the speech quality [6], [12], [13]. Let us denote the estimated speech and noise spectral magnitude vectors from NMF algorithm as $\mathbf{p}_X(\mathbf{v})$ and $\mathbf{p}_M(\mathbf{v})$. From these parameters, the gain function for $\mathbf{v}$ is obtained and $\mathbf{x}$ is derived as follows [13]:

$$\begin{aligned} \mathbf{p}_X(\mathbf{v}) &= W_X \mathbf{h}_X(\mathbf{v}) && (2.9) \\ \mathbf{p}_M(\mathbf{v}) &= W_M \mathbf{h}_M(\mathbf{v}) && (2.10) \\ \mathbf{x} &= \frac{(\mathbf{p}_X(\mathbf{v}))^r}{(\mathbf{p}_X(\mathbf{v}))^r + (\mathbf{p}_M(\mathbf{v}))^r} \otimes \mathbf{v} && (2.11) \end{aligned}$$

where r is a positive constant which controls the order of the filter. Combining $\mathbf{x}$ with the phase information from noisy speech, the speech spectrum is estimated and can be easily transformed to the estimated speech waveform.

Though the speech enhancement algorithm using NMF technique is simple and easy to implement, the performance of the NMF-based algorithms usually degraded when the subspaces of speech and noise overlap. In these noise conditions, the basis vectors of the speech and noise could be similar and minimizing the objective function do not secure the perfect separation of source components from their mixture. This problem will be discussed in Chapter 5 in more detail.

Chapter 3

Deep Neural Networks

3.1 Introduction

In this chapter, the structure of DNN used in this thesis is introduced. The conventional feedforward network structure are explained and the backpropagation algorithm and objective functions which are used to train the DNN are introduced.

The DNN consists of an input layer, a few hidden layers and an output layer which are fully connected to their adjacent layers. For the sake of notation simplicity, the number of hidden layers is denoted as K and the input and output layers of the DNN are denoted as the 0-th and $(K+1)$-th layers of the DNN, respectively.

The input feature structure of the DNN is decided considering the task of the DNN. After the features are extracted from the input data, the DNN can estimate the corresponding output vectors using the feedforward algorithm.

For the k-th hidden layer, the number of nodes in the layer is denoted by l_k. The l_k-dimensional activation vector for the t-th frame $\mathbf{v}_t^k$ is defined as follows:

$$\mathbf{v}_t^k = q(\mathbf{a}_t^k) = q(W^k \mathbf{v}_t^{k-1} + \mathbf{b}^k) \tag{3.1}$$

where $\mathbf{a}_t^k$, W^k, and $\mathbf{b}^k$ denote the l_k-dimensional excitation vector, $l_k \times l_{k-1}$-dimensional weight matrix and l_k-dimensional bias vector, respectively, and $q(\cdot)$ represents an element-wise activation function. In this thesis, the hidden layers of the DNN use the element-wise logistic sigmoid function or rectified linear unit (ReLU) functions which are respectively defined as follows:

$$\sigma(\mathbf{a}_t^k(i)) = \frac{1}{1 + e^{-\mathbf{a}_t^k(i)}} \tag{3.2}$$

$$g(\mathbf{a}_t^k(i)) = max(\mathbf{a}_t^k(i), 0) \tag{3.3}$$

where $\mathbf{a}_t^k(i)$ denotes the i-th element of the vector $\mathbf{a}_t^k$.

For the output layer of the DNN, various activation functions are chosen depending on the target task. In this thesis, the element-wise logistic sigmoid and linear activation functions are used for the activation function of the output layer. The linear activation function is given by

$$\mathbf{v}_t^{K+1} = W^{K+1}\mathbf{v}_t^K + \mathbf{b}^{K+1}. \tag{3.4}$$

The element-wise logistic sigmoid function for the output layer is the same to that for the hidden layer.

In the rest of the thesis, we denote the network output feature as $\mathbf{x}_t$ and the corresponding target feature as $\mathbf{y}_t$ for simplicity.

3.2 Objective Function

In this subsection, the conventional objective function for the regression and classification tasks are introduced. In the regression task, the DNN-based algorithms usually adopt the linear activation function. Then, the mean square error between

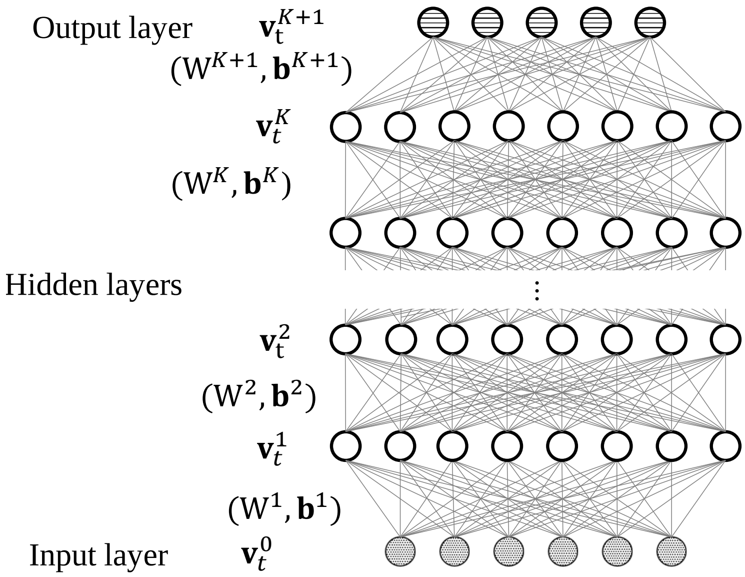

Figure 3.1: Scheme of the DNN with K hidden layers.

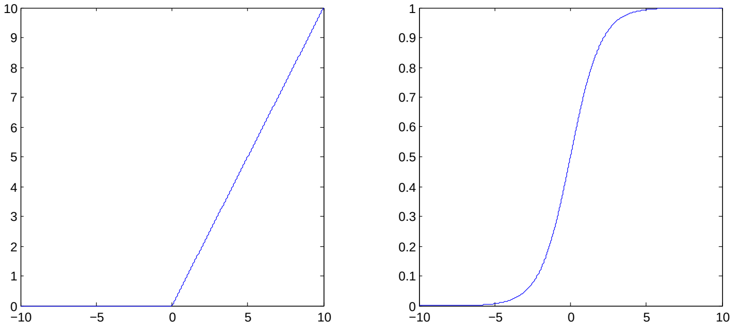

Figure 3.2: Plot of the ReLU (left) and logistic sigmoid (right) functions for the hidden layers of the DNN.

the network output and target features is minimized to learn the mapping between the input and output data. The mean square error is defined as follows:

$$C_{mse} = \frac{1}{I} \sum_{t=1}^{T} \sum_{i=1}^{I} (\mathbf{x}_t(i) - \mathbf{y}_t(i))^2. \tag{3.5}$$

where I denotes the dimension of $\mathbf{x}_t$.

In the classification task, the output layer of the DNN is usually estimated through the softmax function. By applying the softmax function, each output node of the DNN represents the posterior probabilities of each class given the input feature. Each output node of the DNN is given by which is defined as follows:

$$\begin{aligned} p(Class_i|\mathbf{v}_t^0) &= \mathbf{v}_t^{K+1}(i) && (3.6) \\ &= \frac{e^{\mathbf{a}_t^{k+1}(i)}}{\sum_{j=1}^{I} e^{\mathbf{a}_t^{k+1}(j)}}. && (3.7) \end{aligned}$$

When the softmax function is adopted to the DNN, the parameters of the DNN are optimized by the gradient from the cross-entropy function which is defined as follows:

$$C_{ce} = -\sum_{i=1}^{I} \mathbf{y}_t(i) \ln \mathbf{x}_t(i) + (1 - \mathbf{y}_t(i)) \ln(1 - \mathbf{x}_t(i)) \tag{3.8}$$

where $\ln(\cdot)$ denotes the natural log function.

3.3 Stochastic Gradient Descent

From the objective function which are defined in the previous subsection, the parameters of the DNN are optimized to minimize the objective function. The traditional approach for the DNN training is based on the stochastic gradient descent algorithm and the backpropagation algorithm with the chain rule. In this subsection, the vanilla stochastic gradient descent algorithm is briefly described.

From differentiable objective function C, the gradient of the parameter of the DNN θ is given by the derivative of C with respect to θ. Then, the parameters of the DNN could be updated as follows:

$$\theta_{t+1} = \theta_t - \gamma \frac{\partial C}{\partial \theta}(t) \tag{3.9}$$

where γ is the learning rate. In the mini-batch gradient descent algorithm, the parameters of the DNNs are modified by the gradient averaged for each mini-batch.

In the conventional stochastic gradient descent algorithm, the gradients of the parameters would be oscillate in direction of high curvature while the gentle and consistent gradient would likely to be neglected. In order to damp the oscillation in direction of the high curvature, the momentum method was introduced [14]. In the stochastic gradient descent algorithm with the momentum, θ is updated as follows:

$$\Delta\theta_{t+1} = \beta\Delta\theta_t - \gamma\frac{\partial C}{\partial\theta}(t), \tag{3.10}$$

$$\theta_{t+1} = \theta_t + \Delta\theta_t \tag{3.11}$$

where β controls the decay rate of $\Delta\theta_t$.

Chapter 4

DNN-Based Voiced Activity Detection with Multi-Task Learning Framework

4.1 Introduction

Voice activity detection (VAD) algorithms have been widely applied to speech communication systems and front-end processing modules for the last few decades. As discussed in earlier chapters, the traditional VAD algorithms have usually been designed based on the assumption of stationary background noise and track the noise power over frames. Meanwhile, by considering the VAD problem as a two-class classification task, recent papers on VAD adopt several machine learning techniques [15]–[18] to overcome the disadvantage of the conventional algorithms. The fundamental idea of these techniques is to learn the mapping between the noisy speech features and the corresponding voice activity status from a huge amount of exemplars.

Among a number of machine learning techniques, the deep neural network (DNN) which learns the mapping between the noisy speech features and the corresponding voice activity status with its deep hidden structure has been one of the most popular techniques. The DNN-based VAD algorithm outperformed the traditional and other machine learning-based VAD algorithms since the DNN is efficient in learning the complicated inter-dependencies between the input variables [19].

Similar to other machine learning-based algorithms, The DNN-based models show worse performance in the noise conditions which are not considered in the training database compared to those in the matched noise conditions. To ameliorate this performance degradation, the DNN should learn the general mapping between the input and output features to cover various noise environments.

In this chapter, we propose a novel approach which enhances the robustness of DNN with the use of the multi-task learning (MTL) framework [20]. Recently, MTL framework have been widely studied in the various applications including the speech signal processing system [7]–[10]. Motivated by these studies, we related the feature enhancement task with the VAD task and jointly train them in the training stage.

By training the DNN with the gradient form the feature enhancement task, the shared hidden layers are encouraged to learn the denoising mapping from the noisy speech features while they also extract information that are relevant to the VAD task. This regularization makes the MTL-DNN to be robust to the environmental mismatches. Experiments performed in the matched and mismatched noise conditions show that the performance of the DNN-based VAD can be improved by adopting the MTL framework.

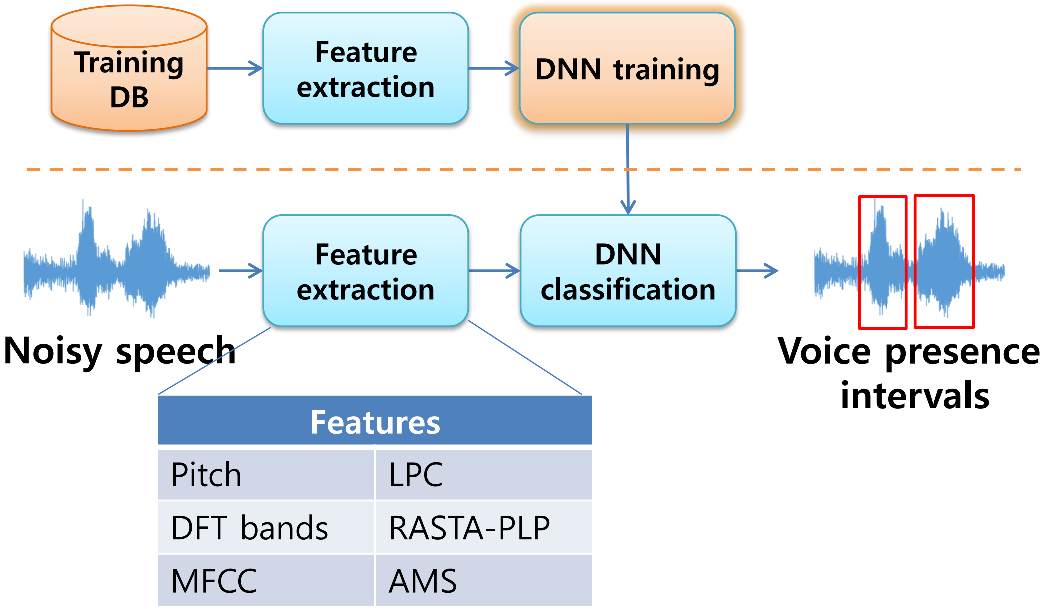

Figure 4.1: Scheme of the DNN-based VAD.

4.2 DNN-Based VAD Algorithm

Fig. 4.1 shows the scheme of the DNN-based VAD. The DNN-based VAD consists of three parts: the pre-training, fine-tuning and test stages. In the training stages, the speech utterances and the background noise waveform are artificially added in order to build a training database for DNN. Then, the DNN is trained to estimate the speech presence probability of the given frame from the concatenated feature vectors.

Fig. 4.2(a) shows the structure of the conventional DNN for the VAD task. In this chapter, the DNN is constructed by stacking several hidden layers which adopts the element-wise logistic sigmoid that is introduced in Chapter 3.

For the output layer, the activation vector consists of a single logistic sigmoid node $\hat{z}$ which is given by

$$\hat{z} = \sigma(W^{K+1}\mathbf{v}^K + \mathbf{b}^{K+1}). \tag{4.1}$$

In the binary classification problem, the logistic sigmoid node is identical to the two-dimensional softmax vector. The network output $\hat{z}$ obtained from (4.1) denotes the speech presence probability of the given frame.

In the pre-training stage, DNN parameters are initialized using stacked restricted Boltzmann machines trained through greedy layer-wise unsupervised learning [22]. After the pre-training stage, the fine-tuning stage which involves stochastic gradient descent and backpropagation is carried out with the cross-entropy objective function C_{VAD} which is defined as follows:

$$C_{VAD} = -zln(\hat{z}) - (1-z)ln(1-\hat{z}) \tag{4.2}$$

where z denotes the actual target output value which equals 1 for active voice and 0 for inactive voice, respectively.

In the test stage, the same feature extraction algorithm is applied to obtain the feature vector sequence from the given noisy utterances. The speech presence probabilities of the input noisy speech frames are estimated by the conventional feedforward algorithm. The speech presence interval decision is made for each frame by following rule:

$$H_d = \begin{cases} H_1, & \text{if } \hat{z} > \varepsilon, \\ H_0, & \text{otherwise} \end{cases} \tag{4.3}$$

where H_1 and H_0 denote active voice and noise-only hypothesis, respectively, and ε is a threshold which is usually set to 0.5.

The post-filters which smooths the output of the DNN over frames or heuristically modify the output of the DNN for perceptual quality improvement could also

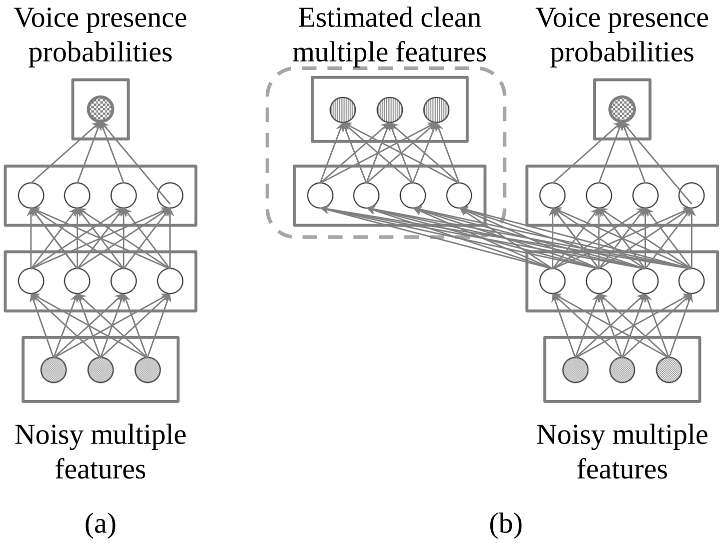
Voice presence probabilities
Noisy multiple features
(a)
Estimated clean multiple features
Voice presence probabilities
Noisy multiple features
(b)

Figure 4.2: Scheme of the conventional DNN (a) and the MTL-DNN (b) for the VAD task. The layers in the dotted line in (b) are discarded before the test stage.

be done. In this chapter, we concentrate on the performance of the DNN in the VAD task. Post-processing techniques such as smoothing the fragile segments [21] could increase the performance of VADs, but this post-processing techniques are beyond the scope of this letter.

4.3 DNN-Based VAD with MTL framework

The performance of the DNN-based VAD algorithm with the conventional training procedure is deteriorated in some mismatched noise conditions since the mapping learned by the DNN is not general enough to cover the environmental mismatches. When the DNN is trained with the conventional training procedure, the DNN can learn the mapping between the noisy features and the corresponding voice activity status in several ways, e.g., relying on trivial characteristics or simply memorizing

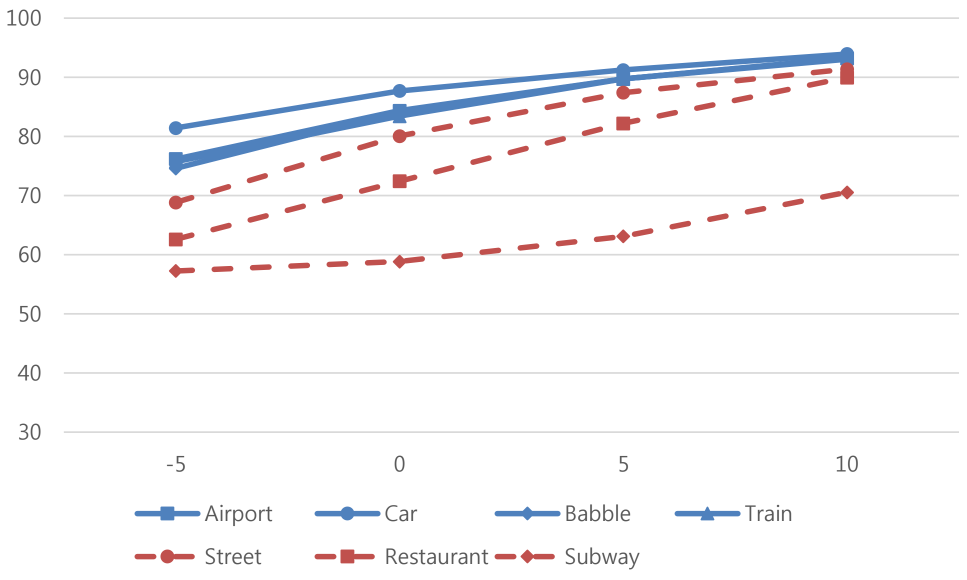

Figure 4.3: Frame accuracies of the DNN-based VAD in matched (solid) and mismatched (dash-dotted) noise environments in various SNR values.

the training data [9]. Thus DNNs with these mappings may have difficulties in estimating voice activity status when there exists severe mismatch in noise condition. Fig. 4.3 shows the performance of the conventional DNN in the matched and mismatched environments. In this figure, the performance of the DNN in mismatched noise environments were degraded compared to those in the matched noise environments.

In this chapter, we introduce the MTL framework which combines the conventional VAD task with a feature enhancement task during the training stage in order to ameliorate this performance degradation. The DNN with the proposed MTL framework (MTL-DNN) denoises the noisy speech features in the shared hidden layers and learns the mapping between the denoised hidden representation and the corresponding voice activity status in the separated layers for the VAD task. The mapping which is learned by the MTL-DNN is more robust against the environmental mismatches since it represents the general denoising function for the speech features.

Fig. 4.2(b) shows the network structure of the MTL-DNN where the conventional VAD and feature enhancement tasks share the lower hidden layers of the DNN. The right part of this network has the same structure with Fig. 4.1(a) which performs VAD while the left part performs feature enhancement. Both the left and right parts of the DNN share the lower hidden layers including the input layer but produce different types of outputs; left part gives the enhanced speech features while the right part outputs the voice activity status. The left part of the network is treated as a subsidiary DNN, which means that it is used only for training the DNN parameters and it is removed after training.

Similar to the conventional DNN-based VAD technique, the MTL-DNN is trained by passing through the pre-training and fine-tuning stages. In the pre-training stage, the parameters of the MTL-DNN are initialized by the same layer-wise unsupervised learning algorithm. In the fine-tuning stage, the objective function for the feature enhancement task C_{FE} is given by the Euclidean distance between the target clean feature $\mathbf{y}$ and its estimated value $\mathbf{x}$ as follows:

$$C_{FE} = \sum_{i} (\mathbf{x}(i) - \mathbf{y}(i))^2. \tag{4.4}$$

The objective function for MTL-DNN training, C_{MTL} is derived by combining C_{VAD} and C_{FE} as given by

$$C_{MTL} = \lambda C_{VAD} + (1 - \lambda) C_{FE} \tag{4.5}$$

where λ is a trade-off parameter between the VAD and feature enhancement tasks.

One important characteristic of the MTL framework is that it only increases the training complexity. After the fine-tuning stage, the layers for the conventional VAD task are preserved while those parts that are relevant to only the subsidiary task are discarded. In the test stage, the same feedforward algorithm and decision rule to the conventional DNN-based VAD algorithm are applied to estimate the voice activity status.

4.4 Experimental Results

4.4.1 Experiments in Matched Noise Conditions

In order to evaluate the performance of the proposed algorithm, we conducted a set of VAD experiments. In the experiments, the {Airport, Babble, Car, Restaurant, Street, Subway, Train} noisy speech data was taken from the Aurora2 database [23]. Each waveform was sampled at 8 kHz and the frame length was 25 ms with a frame-shift of 10 ms. The list of features for the DNN input used in the experiments is shown in Table 4.1. We compared the frame level accuracies of VAD obtained from the proposed algorithm with those from the conventional DNN-based VAD algorithm [19].

To train the DNNs, a set of noisy speech utterances with SNRs from -5 to 10 dB were used. 1001 utterances for each SNR and each noise type were randomly split into 300 utterances of training set, 300 utterances of validation set and 401 utterances of test set, respectively. The input features of the DNNs were normalized to have zero mean and unit variance. The DNNs were implemented using the Theano neural network toolkit [24].

The DNN with conventional training procedure was constructed by stacking 2 hidden layers of 1024 nodes. We ran 30 epochs for pre-training of each hidden layer

Table 4.1: Feature structures extracted from noisy and clean speech waveform.

Feature	Dimension	Feature	Dimension
Pitch	1	MFCC_{16}	20
DFT	16	LPC	12
DFT_8	16	RASTA-PLP	17
DFT_{16}	16	AMS	135
MFCC	20	**Total**	**273**
MFCC_8	20		

to train the DNN. For Gaussian-Bernoulli RBMs, we fixed the learning rate to 0.001 while for Bernoulli-Bernoulli RBMs we fixed the learning rate to 0.01. For the fine-tuning stage, the learning rate started at 0.1. At the end of each epoch, if the frame accuracy on the development set decreased, the parameters of the DNN were returned to their values at the beginning of the epoch and the learning rate was exponentially decayed with a decaying factor of 0.8. This procedure was continued until the learning rate fell below 0.001. For both stages, we fixed the mini-batch size to 100.

The MTL-DNN was constructed by stacking one shared hidden layer and one separated hidden layer for each task with 1024 nodes each. The clean features for the feature enhancement task were normalized to have zero mean and unit variance. The MTL-DNN was trained with the same training configuration to that of the conventional DNN except the objective function in the fine-tuning stage was changed to (4.5). During the fine-tuning stage, we fixed λ to 0.9.

Tables 4.2 and 4.3 show the frame accuracies of the DNNs with or without the MTL framework in matched noise conditions. From the results, we can see that

Table 4.2: Frame Accuracies (%) of the conventional DNN-based VAD in matched noise conditions.

	SNR (dB)				
	-5	0	5	10	Average
Street	73.45	81.57	87.22	90.45	83.17
Airport	76.19	84.17	89.89	93.38	85.91
Car	79.18	86.84	90.91	93.74	87.67
Babble	73.57	83.24	89.16	93.00	84.74
Train	76.22	83.98	89.92	93.09	85.80
Restaurant	69.93	80.87	87.78	92.15	82.68
Subway	69.77	79.51	87.39	91.62	82.07
Average	74.04	82.88	88.9	94.49	**84.58**

the proposed algorithm showed slightly better performance than the conventional DNN-based VAD. The performance difference between the two DNNs in matched noise condition was not significant since the DNN can learn the mapping between the noisy speech features and the corresponding voice activity status without any denoising function when the background noises match.

4.4.2 Experiments in Mismatched Noise Conditions

We also evaluated the performance of the DNNs when the noises were mismatched between the training and test phases. In this experiment, the DNNs were trained with {Airport, Babble, Car, Train} noisy speech data and tested with {Street, Restaurant, Subway} noisy speech data. For each SNR and each noise in {Airport, Babble, Car, Train} data, 600 utterances were assigned to the training set

Table 4.3: Frame Accuracies (%) of the MTL-DNN-based VAD algorithm in matched noise conditions.

	SNR (dB)				
	-5	0	5	10	Average
Street	73.93	81.71	86.95	90.57	83.29
Airport	77.35	84.68	90.10	93.46	86.40
Car	79.60	86.79	91.04	93.92	87.84
Babble	74.72	84.07	89.78	93.28	85.46
Train	75.05	82.89	89.55	93.17	85.17
Restaurant	70.61	81.20	88.14	92.36	83.08
Subway	69.09	78.84	86.91	91.53	81.59
Average	74.33	82.89	88.92	92.61	**84.69**

and 401 utterances were assigned to the validation set. For each SNR and each noise in {Street, Restaurant, Subway} data, 401 utterances were used as the test data.

Tables 4.4 and 4.5 show the frame accuracies of the DNNs with or without MTL framework in mismatched noise conditions. From the results, we can see that the proposed algorithm outperformed the conventional DNN-based VAD algorithm. These results show that the MTL framework improves the robustness of the DNN especially in mismatched noise conditions.

Figures 4.4, 4.5, 4.6, 4.7 show the ROC curve of the DNN and MTL-DNN of various noise environments in each SNR value. In these figures, we can see that the performance of the DNN-based VAD was also improved in terms of the false alarm and detection rate.

Table 4.4: Frame Accuracies (%) of the conventional DNN-based VAD algorithm in mismatched noise conditions.

	SNR (dB)					
	-10	-5	0	5	10	Average
Street	61.23	68.84	80.06	87.39	91.37	77.78
Restaurant	58.57	62.58	72.43	82.21	89.96	73.15
Subway	56.99	57.25	58.83	63.09	70.54	61.34
Average	58.93	62.89	70.44	77.56	83.96	**70.76**

Table 4.5: Frame Accuracies (%) of the MTL-DNN-based VAD algorithm in mismatched noise conditions.

	SNR (dB)					
	-10	-5	0	5	10	Average
Street	61.75	71.50	81.87	88.40	91.90	79.08
Restaurant	59.00	63.18	73.90	84.37	90.94	74.28
Subway	57.08	57.66	60.80	67.96	77.85	64.27
Average	59.27	64.11	72.19	80.24	86.90	**72.54**

4.5 Summary

In this chapter, we have proposed an MTL framework for robust DNN-based VAD algorithm in mismatched noise conditions. The gradient from the feature enhancement task encourages the shared hidden layers to learn the feature denoising function as well as the mapping for the VAD task. The experiments on the Aurora2 database have been shown that the proposed algorithm outperformed the conventional algorithm in mismatched noise conditions.

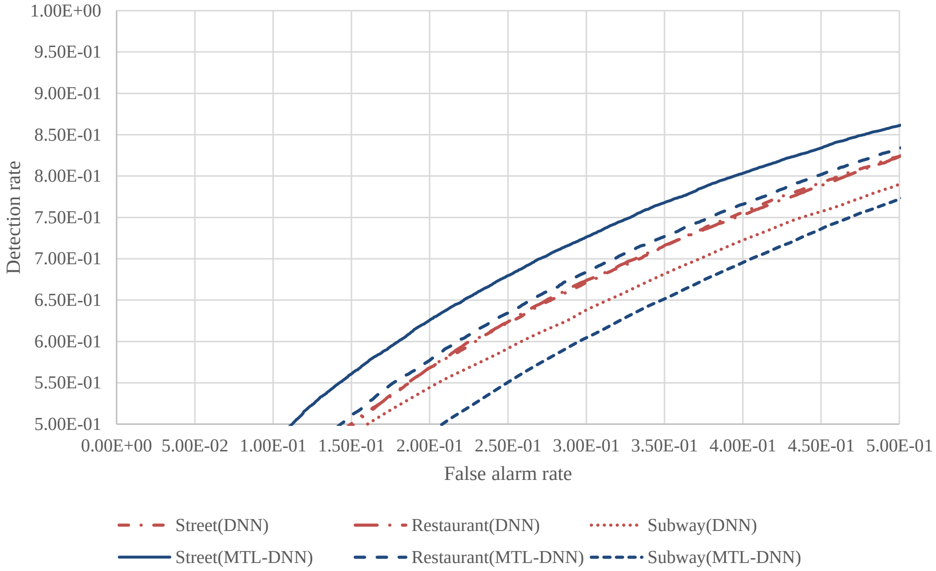

Figure 4.4: ROC curves of the DNN-based VADs various noise environments with -5 dB SNR.

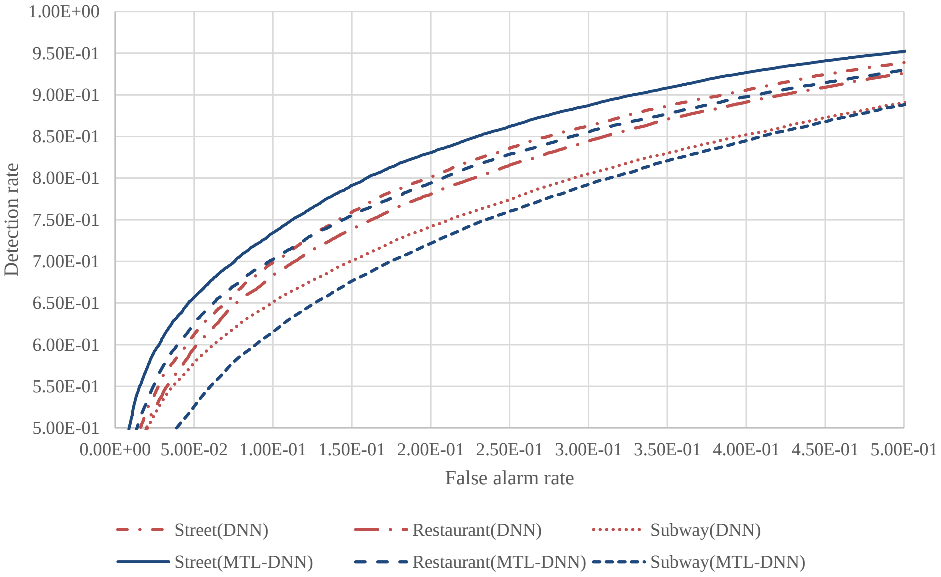

Figure 4.5: ROC curves of the DNN-based VADs various noise environments with 0 dB SNR.

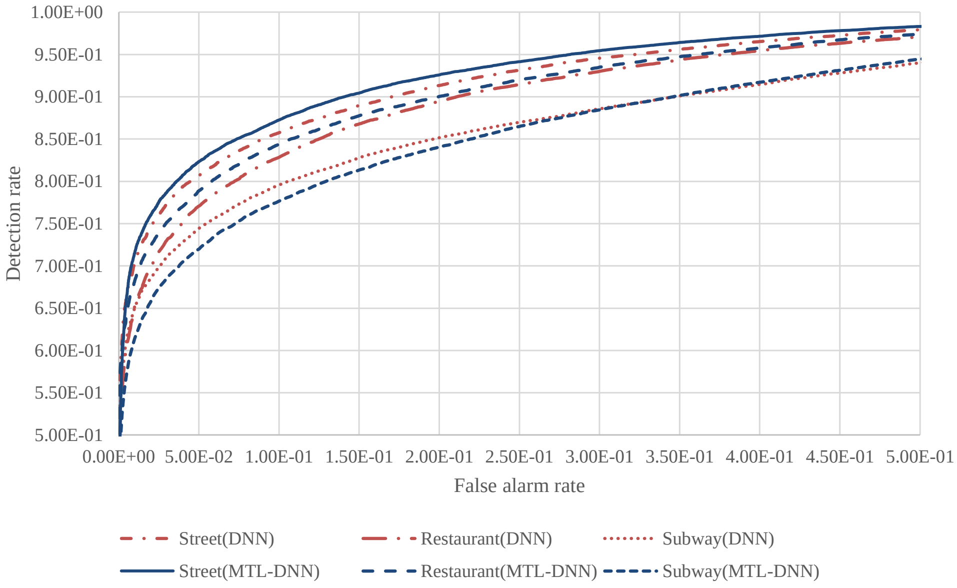

Figure 4.6: ROC curves of the DNN-based VADs various noise environments with 5 dB SNR.

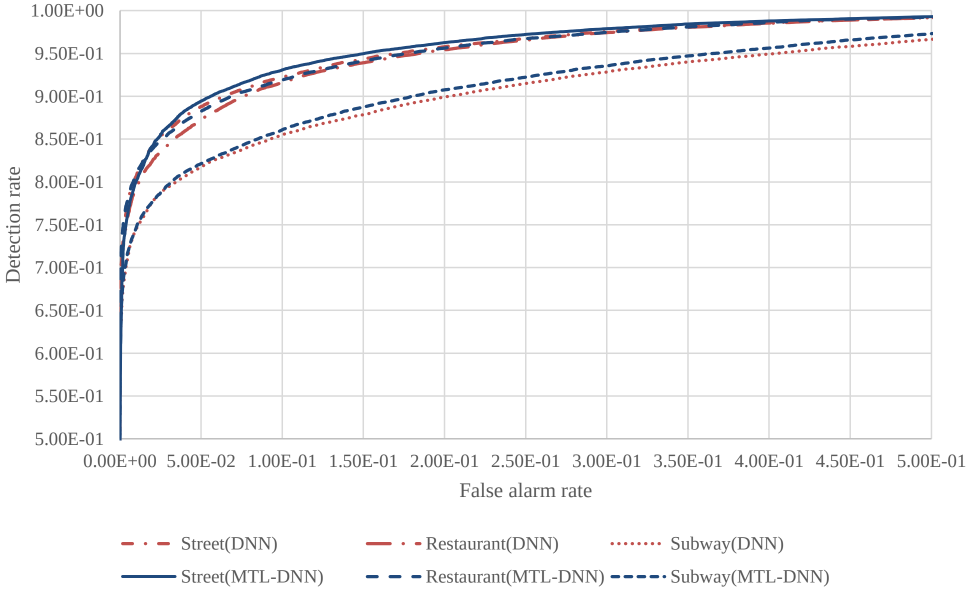

Figure 4.7: ROC curves of the DNN-based VADs various noise environments with 10 dB SNR.

Chapter 5

NMF-based Speech Enhancement Using Deep Neural Network

5.1 Introduction

Dictionary learning has been found to be useful in many classification and regression tasks of signal processing [6], [11]–[13], [25]–[28]. A crucial part of this approach is to analyze a given data matrix based on a fundamental basis structure while minimizing a specific cost function.

Non-negative matrix factorization (NMF) is known as one of the most popular techniques for dictionary learning [11]. In this approach, a non-negative data matrix is approximated by a product of a basis matrix and an encoding matrix with non-negative elements. The NMF technique has been applied to a variety of tasks including object recognition, acoustic signal detection, speech enhancement, speech recognition in adverse environment, and acoustic source separation to name just a few [6], [12], [13], [27], [28].

In [6], [12], [13], NMF is applied to separate a speech source from mixture data. In this task, the basis matrix for all sources is built by concatenating the basis matrices for individual sources so that the product of the corresponding parts of the basis and encoding matrices becomes separated sources. The fundamental assumption underlying the conventional NMF technique with a concatenated basis matrix is that the subspaces which the speech and noise sources span are almost orthogonal to each other.

However, the speech and noise subspaces often overlap, which makes the estimation of an encoding vector and the speech enhancement difficult. This implies that if a data vector generated by a speech source can be possibly represented by a linear combination of basis vectors corresponding to additive noises or vice versa, then the speech source separation is likely to fail. Even though an orthogonality constraint is employed for the basis matrix in [29], it would not resolve this difficulty as long as the basis matrix for each source is trained independently. Recently, several discriminative training approaches which aim to achieve low reconstruction error and high discrimination among different classes at the same time have been proposed [30], [31]. These approaches could enhance the classification and separation performances, but the reconstruction error for each source might be compromised for the vectors which lie in the overlapping regions of the source subspaces.

Unlike the previous approaches, we focus on improving encoding vector estimation for data vectors which are mixtures of speech and interfering sources. To improve the performance of a speech enhancement algorithm with source subspace overlap, we propose a novel supervised algorithm to estimate encoding vectors [32].

A deep neural network (DNN) is used to estimate the encoding vectors which can faithfully reconstruct the desired source data vectors. In the proposed approach, the mapping between the data vectors and the corresponding encoding vectors is modeled with a DNN for which the training data includes mixture data. DNNs are widely used in classification [33] and regression [34] problems and found to be efficient in learning complicated inter-dependencies between the input variables. To show the performance of the proposed algorithm, we performed experiments on the matched and mismatched noise conditions. Experimental results demonstrated that the proposed method outperformed other encoding vector estimation algorithms.

5.2 Encoding Vector Estimation Using DNN

Though simple and easy to implement, the speech enhancement approach with the concatenated basis matrix has serious problems. These problems come from the fact that each basis matrix W_n is independently trained and the NMF analysis is valid only when some orthogonality conditions among $\{W_n\}$ are satisfied.

To show the performance degradation of the conventional NMF technique for speech enhancement, we conducted an experiment with the various signal-to-noise (SNR) conditions. We added clean speech and factory noise samples and reconstructed the speech and noise magnitude spectra from their mixture using the conventional NMF approach which minimizes the KL-divergence. We measured the reconstruction error in terms of the log-spectral distance (LSD) [35] and the results are shown in Fig. 5.1. We can see from the results that the performance of the conventional NMF-based approach degrades severely as the strength of the interfering source increases.

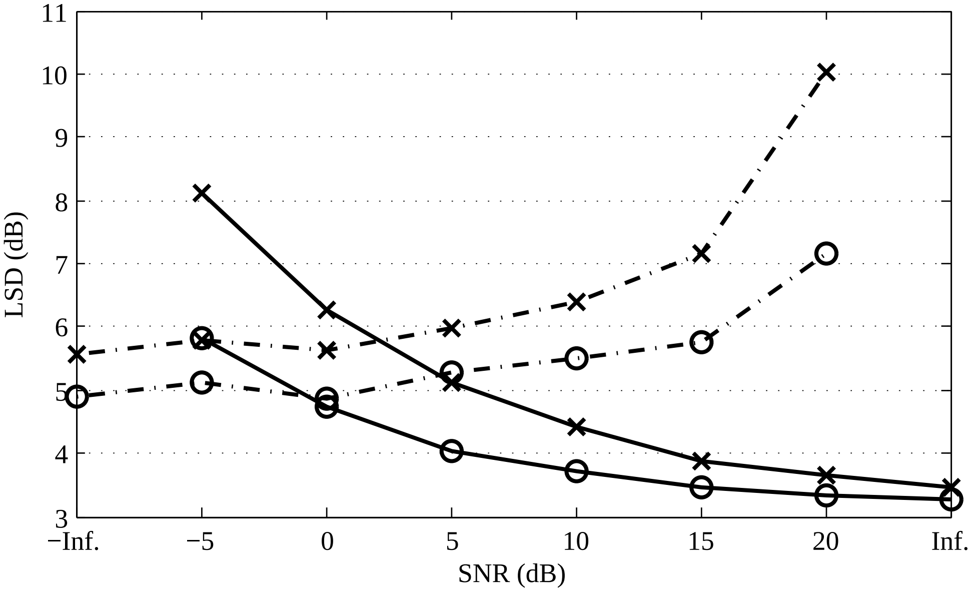

Figure 5.1: Log spectral distances for speech (solid) and factory noise (dash-dotted) obtained from NMF-based speech and noise separation of speech-factory noise mixture based on conventional (x) and DNN-based (o) encoding vector estimation.

In order to minimize the interference of the source subspaces, we propose a novel approach to estimate the encoding vectors of the NMF analysis. Even though the proposed approach is applied to NMF, it can be easily employed in other dictionary learning techniques with only slight modifications.

A major difficulty in speech enhancement with a concatenated basis matrix is how to estimate the encoding vector $\mathbf{h}$ when the orthogonality conditions among $\{W^n\}$ are not satisfied. In this case, minimizing $C(\mathbf{v}|W\mathbf{h})$ does not guarantee the successful separation of the individual sources.

A simple remedy to this is to learn the mapping from an input data vectors to the encoding vectors when we are given a set of mixture data. Under this framework, the problem of estimating the encoding vectors can be treated as a regression task where the input is the mixture data vectors and the output is the encoding vectors corresponding to separate sources. In this chapter, we apply the DNN which accommodates the inter-dependencies between basis matrices of the speech and interfering sources with a deep structure to estimate the NMF encoding vectors.

There are several approaches which apply DNN to estimate source data [36] or source probability [37] from the mixture data directly. Compared with these approaches, our approach which focuses on the encoding vector estimation of the dictionary learning can be more easily extended to adopt the advances in the dictionary learning area such as the update of bases [38]. Also, the number of estimated parameters for the proposed approach is smaller than that for [36].

The proposed technique consists of three parts: NMF training, DNN training, and speech enhancement stages. In the NMF training stage, the conventional NMF technique is applied for each source data matrix separately. The trained speech and noise basis matrix W_X and W_M are used in the following stages after forming a concatenated basis matrix.

In the DNN training stage, each data vector for the speech and interfering sources $\mathbf{x}$ and $\mathbf{m}$ are factorized using the conventional NMF technique with fixed basis matrices. Since this factorization is applied to each source independently, encoding vectors for the speech and noise sources are estimated without ambiguity caused by mixed sources. After obtaining $\{\mathbf{h}_S, \mathbf{h}_M\}$, we artificially generate a mixture data vector $\mathbf{v}$ with some arbitrary weights $\{\kappa_X, \kappa_M\} > 0$ as follows:

$$\mathbf{v} = \kappa_X \ \mathbf{x} + \kappa_M \ \mathbf{m}. \tag{5.1}$$

The optimal encoding vector $\mathbf{h}$ corresponding to this mixture data is then given by

$$\mathbf{h} = \left[\begin{array}{c:c} \kappa_X \mathbf{h}_X{}^{\dagger} & \kappa_M \mathbf{h}_M^{\dagger} \end{array} \right]^{\dagger} \tag{5.2}$$

where the † denotes the transpose of a matrix or a vector. After generating a collection of the artificial mixture data vectors, we can train a DNN where $\mathbf{v}$ in (5.1) is fed to the network as an input and $\mathbf{h}$ as defined in (5.2) is applied as the corresponding target output. Before $\mathbf{v}$ and $\mathbf{h}$ are applied to the DNN, it is useful to normalize them to be in the range of (0, 1). Multiplying weights $\{\kappa_X, \kappa_M\}$ and normalizing $\mathbf{v}$ and $\mathbf{h}$ do not hamper the relationship among the separate source components since the NMF analysis is scale-independent when the KL-divergence is used as an objective function.

It is widely known that DNNs provide a more proper structure than shallow neural networks for representing complicated functions or mappings. However, with randomly initialized parameters, the performance of DNNs is generally worse than that of shallow neural networks. To initialize DNN parameters, the stacked restricted Boltzmann machines accompanied with greedy layer-wise unsupervised learning is adopted to initialize DNN parameters as in [22], [39]. After this pre-training stage, a supervised learning algorithm using backpropagation and stochastic gradient descent is carried out in fine-tuning stage. The Euclidean distance is used as the training cost function in this stage as in [40]. In this chapter, the activation function used in each hidden unit of the DNN is the logistic sigmoid function [33]. The nonlinear activation function is usually used to deal with nontrivial problems with a small number of nodes [40].

A major weakness of the proposed training algorithm is that the number of possible source combinations for generating training data may increase rapidly as more separate sources are taken into account. However, the proposed approach can be considered to be a useful way if the goal is to extract a few target sources of our interest and all the remaining sources are treated as interferences. The DNN with enough number of hidden layers and nodes might learn the inter-dependencies between the target sources and various composite effect of the interfering sources simultaneously. From this aspect, the proposed algorithm can be applied to the speech enhancement or taret source separation tasks in which the number of interested source is one rather than arbitrary source separation task.

Finally, in the speech enhancement stage, an actual mixture data vector $\mathbf{v}$ is applied to the DNN with the standard feedforward processing. The output vector of the DNN is re-scaled to $\hat{\mathbf{h}}(\mathbf{v})$ such that the reconstructed mixture data vector $W\hat{\mathbf{h}}(\mathbf{v})$ has the same L2-norm as $\mathbf{v}$. The estimate for the n-th separate source is then given by the product of the basis matrix W_n with the corresponding part of $\hat{\mathbf{h}}(\mathbf{v})$. In the case of the speech enhancement task, $\mathbf{x}$ and $\mathbf{m}$ are estimated by the $W_X\hat{\mathbf{h}}_X(\mathbf{v})$ and$W_M\hat{\mathbf{h}}_M(\mathbf{v})$, respectively.

To verify the performance of the proposed algorithm, we carried out an experiment the same as in the previous subsection. Fig. 5.1 also shows the LSD obtained from the proposed DNN-based approach. From the result, we can see that in all the tested conditions, the performance was improved compared with that of the conventional technique.

5.3 Experiments

In order to evaluate the performance of the proposed algorithm, we conducted experiments on speech enhancement where all the noise components are considered as a interference source. Similar to the conventional NMF-based approach in the previous chapter, only the magnitude spectrum of the noisy input signal is modified to estimate the clean speech spectrum while the phase parts are kept intact. Let $\mathbf{v}$ denote a magnitude spectrum of the noisy input signal. Then the estimate of the magnitude spectrum of the corresponding clean speech, $\mathbf{x}$, is given by

$$\mathbf{x} = \frac{W_X \hat{\mathbf{h}}_X(\mathbf{v})}{W_X \hat{\mathbf{h}}_X(\mathbf{v}) + W_M \hat{\mathbf{h}}_M(\mathbf{v})} \otimes \mathbf{v} \tag{5.3}$$

where $\otimes$ and $\frac{(\ )}{(\ )}$ mean element-wise multiplication and division operations.

In the experiment, clean speech data was taken for the TIMIT database. The factory, babble, machinegun noises from NOISEX-92 DB [41] were used for training and test, and buccaneer, f16 noises from NOISEX-92 DB and Cafeteria noise from ITU-T recommendation P.501 [42] were used additionally for the test in mismatched condition. Each waveform was sampled at 16 kHz, and a 512-point Hamming window with 75% overlap was applied. We compared the quality of the enhanced speech obtained from the proposed algorithm with those from the traditional NMF-based speech separation algorithm [13] and discriminative NMF (DNMF) [31] in which the basis matrices are trained jointly to describe mixed as well as separate data vectors. The performance of DNN-based separation [36] which estimates the source data vector directly from mixture data through a DNN is also demonstrated.

In the NMF analysis, W_X was trained based on 10000 frames of the clean speech data and W_M was trained by using 9000 frames of noise data. The number of speech and noise basis vectors was set to be 40 each. For DNMF analysis, W_X and W_M were jointly trained over 9000 frames of speech and noise data pairs.

To train the DNN, a set of noisy speech utterances were artificially generated with SNRs from -5 to 20 dB with 5 dB step. For each SNR in this range, 23 different utterances of clean speech were added with the noise to generate noisy speech and corresponding encoding vectors. The DNN was constructed by stacking 3 hidden layers with 400 nodes each. To compare the separation performance of DNN with that of a shallow model, we also trained a shallow neural network (SNN) consisting of only one hidden layer with 1200 nodes. We ran 30 epochs for pre-training of each hidden layer and 300 epochs for fine-tuning to train DNN or SNN. The numbers of hidden layers and nodes in each hidden layer were determined empirically to describe complicated relation between input and output well enough while avoiding over-fitting. The DNN which estimates the source data vector directly [36] was constructed with the same configuration except the target output.

Ten utterances from 5 male and 5 female speakers were used for performance evaluation for each SNR value and noise type. The performance of the speech enhancement was evaluated in terms of the signal to distortion ratio (SDR), signal to interference ratio (SIR), signal to artifacts ratio (SAR) [43] and the perceptual evaluation of speech quality (PESQ) score [44]. Table 5.1 shows the SDR, SIR, and SAR values and Table 5.2. shows PESQ scores for enhanced speech obtained from various algorithms averaged over all noise types when each model was trained for

Table 5.1: SDR, SIR, and SAR values of enhanced speech with various source separation algorithms : models trained for a specific type of noise.

SNR	NMF [13]			DNMF [31]			DNN [36]			SNN-NMF			DNN-NMF		
(dB)	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR
-5	2.86	5.58	8.43	4.45	8.46	**8.78**	5.77	10.54	8.37	5.42	10.06	8.33	**5.88**	**10.68**	8.61
0	7.17	10.10	11.42	8.11	11.56	12.04	9.63	13.85	12.22	9.45	13.91	11.93	**9.79**	**14.06**	**12.44**
5	11.08	15.34	13.58	11.73	15.46	14.86	13.32	17.63	15.63	13.03	**17.69**	15.10	**13.42**	17.62	**15.80**
10	14.02	19.57	15.72	14.97	18.73	17.90	16.93	21.00	19.30	16.60	20.98	18.74	**17.04**	**21.00**	**19.48**
Aver.	8.78	12.65	12.29	9.81	13.56	13.39	11.41	15.75	13.88	11.13	15.66	13.52	**11.53**	**15.84**	**14.08**

Table 5.2: PESQ scores of enhanced speech with various source separation algorithms : models trained for a specific type of noise.

SNR (dB)	NMF [13]	DNMF [31]	DNN [36]	SNN-NMF	DNN-NMF
-5	1.92	1.98	2.06	2.04	**2.07**
0	2.29	2.31	2.48	2.47	**2.50**
5	2.60	2.65	2.85	2.82	**2.87**
10	2.92	2.95	3.20	3.17	**3.21**
Aver.	2.43	2.47	2.65	2.63	**2.66**

a specific type of noise. From the results, we can see that the proposed algorithm outperformed the conventional NMF-based techniques and DNN-based separation.

We performed an additional experiment in which the NMF, DNMF, SNN and DNN models were trained over all types of noises pooled together to examine whether the proposed algorithm can learn various types of source characteristics simultaneously. In this experiment, W_M was trained from 9000 frames of pooled noise data. It is noted that the number of basis vectors in W_M was also fixed to 40 even though

Table 5.3: SDR, SIR, and SAR values of enhanced speech with various source separation algorithms in matched conditions : models trained for pooled noise data.

SNR	NMF [13]			DNMF [31]			DNN [36]			SNN-NMF			DNN-NMF		
(dB)	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR
-5	1.92	5.34	7.03	3.82	7.35	**8.69**	5.38	9.54	8.43	5.27	10.03	8.23	**5.78**	**11.74**	7.82
0	5.99	9.87	9.84	7.65	10.97	11.78	9.43	13.31	**12.22**	9.35	13.99	11.83	**9.73**	**15.05**	11.71
5	9.62	14.96	11.90	11.28	14.89	14.58	13.18	17.10	**15.69**	12.95	17.75	15.04	**13.30**	**18.32**	15.19
10	12.36	19.34	13.76	14.46	18.36	17.29	16.78	20.59	**19.32**	16.53	21.10	18.66	**16.85**	**21.47**	18.90
Aver.	7.47	12.38	10.63	9.30	12.90	13.09	11.19	15.14	**13.91**	11.03	15.72	13.44	**11.41**	**16.65**	13.40

Table 5.4: PESQ scores of enhanced speech with various source separation algorithms in matched conditions : models trained for pooled noise data.

SNR (dB)	NMF [13]	DNMF [31]	DNN [36]	SNN-NMF	DNN-NMF
-5	1.83	1.94	**2.06**	2.06	2.06
0	2.16	2.29	2.45	2.45	**2.50**
5	2.46	2.56	2.83	2.80	**2.85**
10	2.80	2.88	3.16	3.14	**3.21**
Aver.	2.31	2.42	2.63	2.61	**2.66**

we pooled all types of noise together. All the training data used in the previous experiment were pooled together for the training of the SNN and DNN. In this experiment, we also tested in the mismatched conditions where the noises unseen in the training were mixed covering wider range of SNR.

Tables 5.3 and 5.5 show the SDR, SIR, SAR values averaged over all noise types in matched and mismatched conditions when the model was trained for pooled noise data. Comparing Tables 5.1 and 5.3, we can find that the performance of

Table 5.5: SDR, SIR, and SAR values of enhanced speech with various source separation algorithms in mismatched conditions : models trained for pooled noise data.

SNR	NMF [13]			DNMF [31]			DNN [36]			SNN-NMF			DNN-NMF		
(dB)	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR
-10	-7.06	-5.98	6.62	-6.72	-5.87	**8.52**	-6.01	-4.52	5.54	-5.57	-4.10	5.56	**-5.43**	**-3.24**	4.01
-5	-1.63	-0.39	7.82	-0.88	0.15	**10.17**	0.29	1.97	7.83	0.60	2.40	7.52	**0.98**	**3.46**	6.51
0	3.53	5.15	9.90	4.04	5.30	**12.53**	5.58	7.35	11.25	5.85	7.97	10.79	**6.01**	**8.55**	10.31
5	7.89	10.31	12.15	8.42	9.84	**15.77**	10.05	11.81	15.25	10.14	12.30	14.64	**10.38**	**12.72**	14.55
10	11.47	15.69	13.84	12.63	14.57	18.37	14.33	16.13	**19.20**	14.33	16.53	18.64	**14.46**	**16.71**	18.65
25	16.55	**29.35**	16.87	21.80	27.91	23.43	**26.23**	28.76	**30.10**	25.68	29.18	28.53	26.06	28.99	29.44
Aver.	5.13	9.02	11.20	6.55	8.65	14.80	8.41	10.24	**14.86**	8.51	10.71	14.28	**8.74**	**11.20**	13.91

NMF and DNMF deteriorated severely when the noise basis was trained over a pooled noise data. In contrast, there was only slight performance degradation in the case of SNN and DNN, which confirms the robustness of these techniques in various noise environments. Moreover, in most test conditions in the Tables 5.3 and 5.5, DNN-based encoding vector estimation showed better performances than other algorithms.

Table 5.4 and 5.6 also show the PESQ scores averaged over all noise types in matched and mismatched conditions when the model was trained for pooled noise data. From these tables, we can see that the enhanced speech with the proposed algorithm showed slightly better performance than other speech enhancement algorithms. The proposed algorithm showed better PESQ scores than those from other algorithms especially in high SNR conditions. In these conditions, the small deviation from the speech subspace which is caused by the confusion between the speech and noise basis matrices are reduced by the proposed algorithm.

Table 5.6: PESQ scores of enhanced speech with various source separation algorithms in mismatched conditions : models trained for pooled noise data.

SNR (dB)	NMF [13]	DNMF [31]	DNN [36]	SNN-NMF	DNN-NMF
-10	1.18	**1.23**	1.18	1.20	1.11
-5	1.51	1.56	1.53	**1.59**	1.52
0	1.83	1.88	1.97	**1.99**	1.98
5	2.17	2.20	2.36	2.36	**2.38**
10	2.52	2.52	2.72	2.71	**2.74**
25	3.42	3.52	3.66	3.66	**3.66**
Aver.	2.11	2.15	2.24	**2.25**	2.23

5.4 Summary

In this chapter, we have proposed a novel approach to estimate the encoding vectors based on DNN. The DNN-based framework for the speech enhancement is introduced and the detailed training database configuration and the DNN training scheme are presented. Through a series of experiments on speech enhancement, we have proved that the performance of the proposed algorithm outperforms the conventional NMF-based technique.

Chapter 6

DNN-Based Monaural Speech Enhancement with Temporal and Spectral Variations Equalization

6.1 Introduction

For a number of decades, monaural speech enhancement using a single microphone has been widely studied to improve various communication and signal processing systems [45]. Though considerable performance improvements have been achieved by various approaches, speech enhancement in realistic noise environments still remains a challenging problem.

In order to enhance the noisy speech in various noise environments, deep neu-

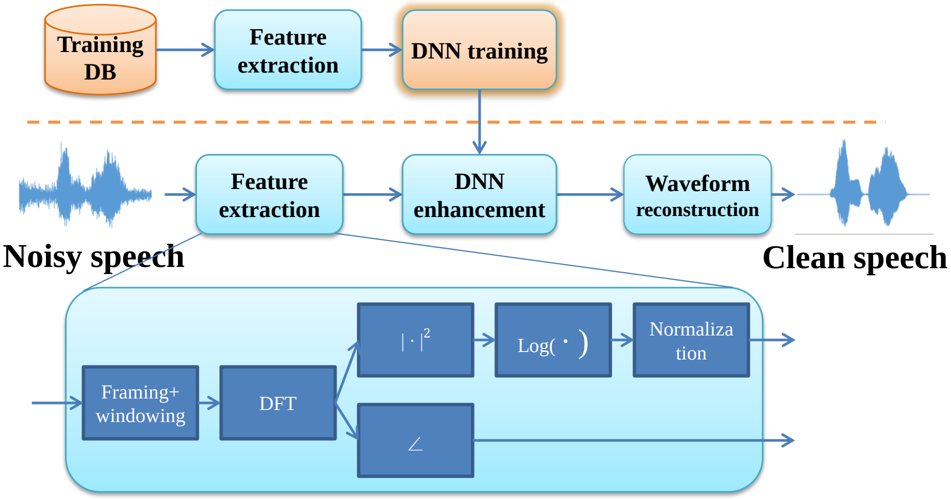

Figure 6.1: Scheme of the DNN-based speech enhancement algorithm.

ral networks (DNNs) which can learn complicated inter-dependencies between the input variables [19], [33], [34], [46] were successfully introduced to the speech enhancement area [47]. In these approaches, the DNN provides a mapping between consecutive noisy speech frames and the corresponding clean speech frame with its deep hidden structure. Fig. 6.1. shows the basic scheme of the DNN-based enhancement algorithm in which the magnitude spectrum of the noisy speech is modified by the network. In thisz figure, the DNN estimates the clean log-spectar from the corresponding noisy log-spectra. Furthermore, in [48], global variance (GV) equalization post-filter, dropout training, and noise-aware training techniques were incorporated to DNN-based speech enhancement to improve the speech quality in mismatched noise conditions.

Many studies have applied the DNN-based approach to speech enhancement and target speaker separation with various new ideas. Huang et al. proposed a technique to jointly optimize all the sources with a discriminative objective function for DNN and recurrent neural network (RNN) [49]. Han et al. applied a DNN-based method for joint dereverberation and denoising followed by iterative signal reconstruction [50]. The training targets of the DNNs were studied in [51] and the complex ratio masking was also proposed by Williamson et al. [52]. Zhang et al. investigated the performance of the mapping- and masking-based training targets both theoretically and experimentally in [53] where they also proposed the multi-context stacking networks for deep ensemble learning.

It is well-known that the estimated speech trajectories obtained from the DNN-based algorithms are usually over-smoothed compared to those of the clean speech. Conventional DNN-based speech enhancement algorithms generally apply the objective functions which are related to the mean square error between the enhanced and clean speech features. However, since these measures are derived from each time-frequency bin separately rather than from whole spectral trajectory, the enhanced speech obtained from the DNN could be over-smoothed compared to the original speech. The speech generated from these enhancement algorithms may result in muffled sound quality and decreased intelligibility [48], [54], [55]. In addition, the mean square error calculated in the linear frequency scale does not match the human auditory perception where the sensitivity follows the Mel-frequency scale. The perceptual quality of the enhanced speech would be improved if the DNN can calculate the errors based on this non-linear frequency scale.

In this chapter, we propose a novel DNN-based speech enhancement algorithm which computes the gradients based on a perceptually motivated non-linear frequency scale and alleviates the over-smoothness problem by equalizing temporal and spectral variations of the enhanced speech to match those of the clean speech. The main contributions of the proposed algorithm are summarized as follows:

First, we apply the Mel-scale weight to fit the objective function to the critical frequency bands of hearing. Similar to the human auditory perception, the network trained using the Mel-scaled gradients is more sensitive to the perceptually important frequency bins. The Mel-frequency scale was adopted to speech enhancement in [56] to smooth the gain function over spectral coefficients. In contrast, the Mel-scale is introduced to prioritize the gradients according to the perceptual importance in the proposed algorithm.

Second, the objective function for DNN training is modified to incorporate the temporal and spectral variation similarities between the enhanced and clean speech. By equalizing the temporal and spectral variations, the enhanced speech could have the spectral peaks and valleys distributed similarly to those of the clean speech. The proposed objective functions are motivated by the relation between the human intelligibility and short-time analysis on one-third octave band trajectory [57]. We adopt variation similarity over short-time trajectories and spectral coefficients into the DNN-based speech enhancement framework and analyze their effect on the naturalness and intelligibility of the enhanced speech.

The proposed objective function is also related to the sequence-discriminative training technique originally developed for automatic speech recognition (ASR) [58], [59]. In these studies, various sequence-discriminative criteria were proposed to train the DNN-based acoustic models for the speech recognizer. Instead of adopting criteria that are based on mutual information or phone error, the proposed algorithm targets human auditory perception and introduces the variation similarity which are found to be closely related to the perceptual quality of the enhanced speech.

6.2 Conventional DNN-Based Speech Enhancement

The task of DNN-based speech enhancement can be divided into the training and test stages. In the training stage, the noisy speech features and the corresponding clean speech features are respectively fed to the input and output nodes of the DNN, and the network is optimized to minimize the mean square error between the enhanced and clean speech features. After the training stage, the clean speech features are estimated from the noisy speech features through the DNN and a GV equalization post-filter is applied to compensate the over-smoothed output trajectory. In this section, we present the feature structures and training scheme of the conventional DNN-based speech enhancement algorithm.

6.2.1 Training Stage

In the training stage, the input and output features of the DNN are respectively extracted from the noisy speech utterances and corresponding clean speech utterances. The input and output features of the DNN are usually normalized to have zero mean and unit variance before being fed to the network.

For the input and output features, we extract log-power spectra of the noisy and clean speech [48], [50]. Let us denote l_F-dimensional normalized log-power spectra of the noisy speech and clean speech at the t-th frame as $\mathbf{z}_t$ and $\mathbf{y}_t$, respectively. Then, the input feature vector $\mathbf{v}_t^0$ is generally constructed as follows:

$$\mathbf{v}_t^0 = [\mathbf{z}_{t-K}^\dagger, \mathbf{z}_{t-K+1}^\dagger, \cdots, \mathbf{z}_{t+K}^\dagger]^\dagger \tag{6.1}$$

where K denotes an input context expansion parameter and $\mathbf{z}_t^\dagger$ denotes the transpose of a vector $\mathbf{z}_t$.

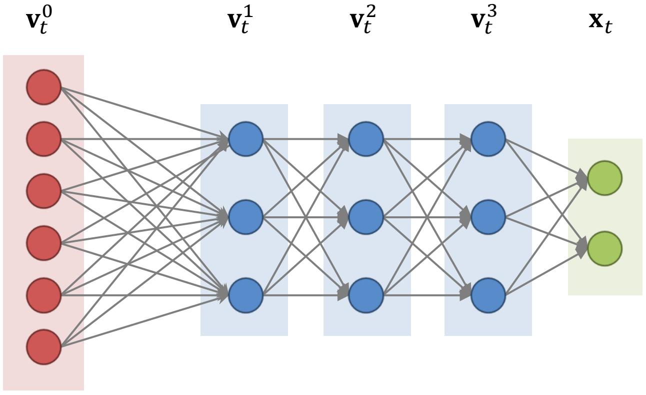

Figure 6.2: Scheme of the DNN with three hidden layers.

Fig. 6.2 shows the structure of a typical DNN with three hidden layers. The DNN consists of an input layer, a few hidden layers and an output layer which are fully connected to their adjacent layers. In this chapter, all hidden layers of the DNN are assumed to use the rectified linear function which is defined as follows:

$$g(\mathbf{a}_t^l(i)) = max(\mathbf{a}_t^l(i), 0). \tag{6.2}$$

After all the hidden layer activations are computed, the l_F-dimensional output vector $\mathbf{x}_t$ is produced by

$$\mathbf{x}_t = W^{L+1}\mathbf{v}_t^L + \mathbf{b}^{L+1}. \tag{6.3}$$

In this chapter, the parameters of the DNN are initialized randomly [60] and optimized using the stochastic gradient descent algorithm. In the training stage, the mean square error between the network output $\mathbf{x}_t$ and the target feature $\mathbf{y}_t$ is minimized, which is given by

$$C_{mse} = \frac{1}{TF}\sum_{t=1}^{T}\sum_{f=1}^{F}(\mathbf{x}_t(f) - \mathbf{y}_t(f))^2 \tag{6.4}$$

where T denotes the total number of frames of the given training data.

6.2.2 Test Stage

In the test stage, the clean speech estimate $\mathbf{x}_t$ is obtained from $\mathbf{v}_t^0$ through the standard feedforward processing. In the speech enhancement algorithm without GV equalization, $\mathbf{x}_t$ is de-normalized to $\bar{\mathbf{x}}_t$ as follows:

$$\bar{\mathbf{x}}_t = \mathbf{x}_t \otimes \mathbf{s} + \mathbf{d} \tag{6.5}$$

where $\mathbf{d}$ and $\mathbf{s}$ are respectively the mean and standard deviation vectors used to normalize the output feature of the DNN, and $\otimes$ denotes element-wise multiplication between two vectors. In this chapter, only the magnitude spectrum of the speech is estimated while the phase parts of the noisy speech are kept intact.

One of the significant drawbacks of the conventional DNN-based speech enhancement algorithm is that it usually results in over-smoothed spectral trajectories of the enhanced speech. In order to alleviate this phenomenon, the GV equalization post-filter which modifies the variance of $\mathbf{x}_t$ to match that of $\mathbf{y}_t$ is usually employed. In this chapter, the frequency-independent GV equalization method which has been known to perform better than the frequency-dependent approach [48] is applied as a conventional post-filtering technique.

In the frequency-independent GV equalization, the global variances of $\mathbf{x}_t$ and $\mathbf{y}_t$ are computed as follows:

$$GV(\mathbf{x}) = \frac{1}{TF}\sum_{t=1}^{T}\sum_{f=1}^{F}(\mathbf{x}_t(f) - \frac{1}{TF}\sum_{t=1}^{T}\sum_{f=1}^{F}\mathbf{x}_t(f))^2, \tag{6.6}$$

$$GV(\mathbf{y}) = \frac{1}{TF}\sum_{t=1}^{T}\sum_{f=1}^{F}(\mathbf{y}_t(f) - \frac{1}{TF}\sum_{t=1}^{T}\sum_{f=1}^{F}\mathbf{y}_t(f))^2. \tag{6.7}$$

Based on (6.6) and (6.7), the frequency-independent GV factor α is given by

$$\alpha = \sqrt{\frac{GV(\mathbf{y})}{GV(\mathbf{x})}}, \tag{6.8}$$

and it is multiplied to $\mathbf{x}_t$ before de-normalization as follows:

$$\bar{\mathbf{x}}_t = \alpha\ \mathbf{x}_t \otimes \mathbf{s} + \mathbf{d}. \tag{6.9}$$

In the GV equalization post-filter, multiplying the GV factor to the output feature can be viewed as imposing an exponential factor in the linear spectral magnitude domain. By this post-filter, the variance of the spectral trajectory is enlarged or diminished depending on the value of α. In most cases, α is bigger than 1 and the lack of dynamics in $\mathbf{x}_t$ is alleviated to some extent.

After the network output is de-normalized, they are transformed to the linear spectra and the clean speech waveform is generated by the conventional overlap-add method.

6.3 Perceptually-Motivated Criteria

In this section, we propose a novel speech enhancement algorithm that is based on DNN. We introduce the proposed objective function which consists of the Mel-scale weighted mean square error, and the temporal and spectral variation similarities between the enhanced and clean speech over adjacent frames or frequency bins.

6.3.1 Perceptually Motivated Objective Function

Our framework to incorporate the perceptually motivated criteria is to replace the conventional mean square error C_{mse} given in (6.4) by a modified objective function C defined as

$$C = C_{wmse} + \lambda_1(1 - \rho_1) + \lambda_2(1 - \rho_2) \tag{6.10}$$

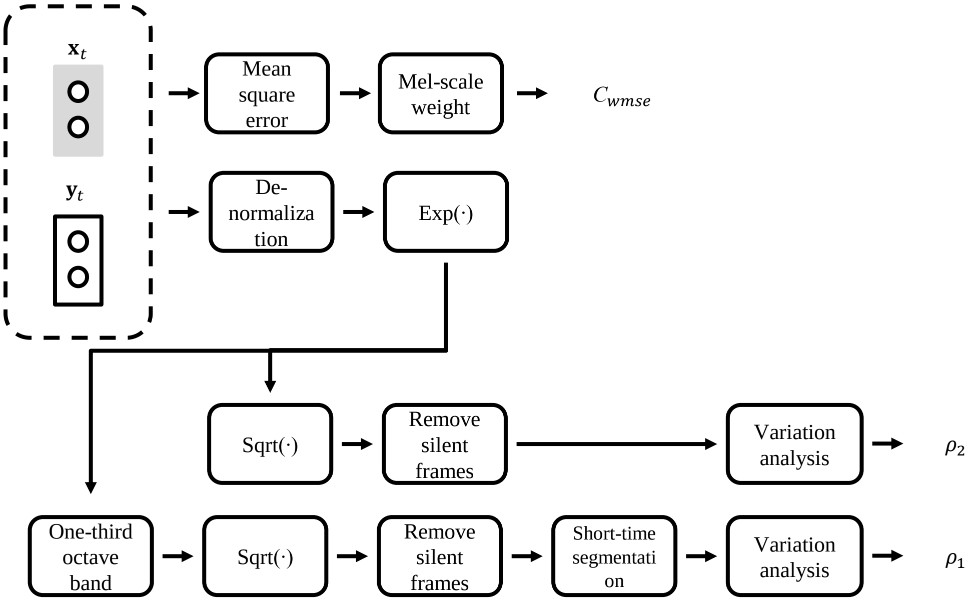

Figure 6.3: Scheme of the proposed objective function which incorporates Mel-scale weighted mean square error, temporal and spectral variation similarities.

where λ_1 and λ_2 denote the weights controlling the contributions of the three separate sub-costs, C_{wmse}, $(1 - \rho_1)$, and $(1 - \rho_2)$. Fig. 6.3 shows the procedures for computing these three sub-costs given $\{\mathbf{x}_t\}$ and $\{\mathbf{y}_t\}$.

In the training stage, parameters of the network are optimized so as to minimize C via the stochastic gradient descent algorithm. The test stage of the DNN remains the same to that of the conventional DNN approach. Note that the only difference of the proposed method from the conventional DNN-based speech enhancement algorithm is that it applies a new objective function for DNN training. Now, we will give the detail on how to derive C_{wmse}, ρ_1, and ρ_2 which jointly specify the objective function.

6.3.2 Mel-Scale Weighted Mean Square Error C_{wmse}

We modify the original mean square error C_{mse} to take the Mel-frequency scale into consideration. The Mel-frequency is defined as follows [61]:

$$\varpi = 2595 \log_{10}(1 + \frac{\zeta}{700}) \tag{6.11}$$

where ϖ and ζ denote the Mel-frequency and the corresponding linear frequency, respectively. The relative importance of each spectral coefficient can be determined by the derivative of the Mel-frequency at the corresponding frequency, i.e.,

$$d(f) = min\left(\frac{d\varpi}{d\zeta}|_{\zeta=f}, \eta\right) \tag{6.12}$$

where η is a constant setting the minimum weight value. Then, the Mel-scale weighted mean square error C_{wmse} is defined by multiplying the normalized weight $w(f)$ with each element of C_{mse} as follows:

$$w(f) = \frac{d(f)}{\sum_{f=1}^{F} d(f)}, \tag{6.13}$$

$$C_{wmse} = \frac{1}{T}\sum_{t=1}^{T}\sum_{f=1}^{F} w(f)(\mathbf{x}_t(f) - \mathbf{y}_t(f))^2. \tag{6.14}$$

Fig 6.4. shows the values of $w(f)$ over frequency axis. Compared to C_{mse}, C_{wmse} emphasizes the error in the low-frequency bins which are crucial for speech naturalness and intelligibility.

6.3.3 Temporal Variation Similarity ρ_1

It has been known that the similarity in frequency band trajectories between the enhanced and clean speech is related to the intelligibility of the enhanced speech [54], [55]. In [57], a speech intelligibility metric using temporal variation over the one-third octave band trajectory is presented. Motivated by these studies, we attempt

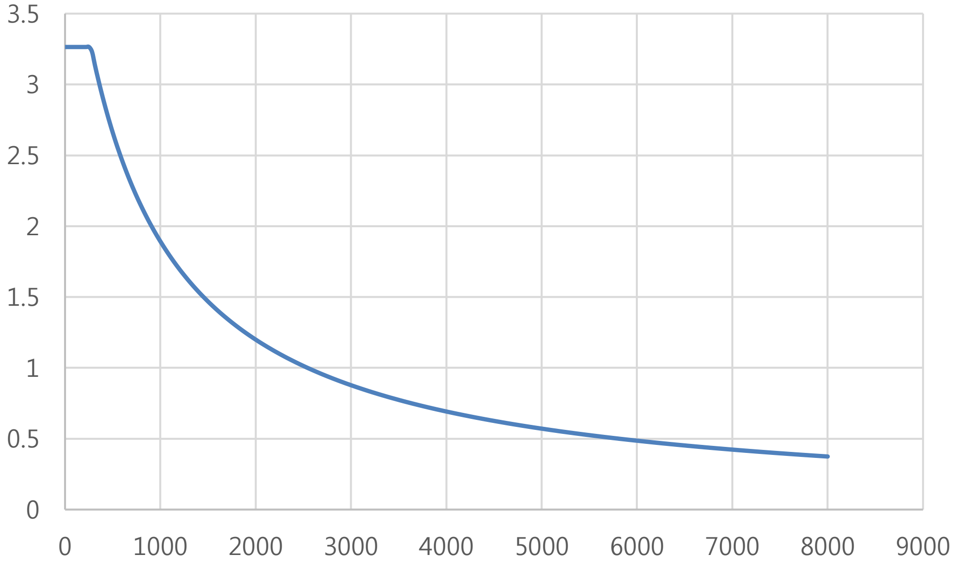

Figure 6.4: Results of preference test (%) between the conventional and proposed algorithms in the speech quality in various SNR values.

to equalize the temporal variation of the one-third octave band trajectory of the enhanced speech during the DNN training session.

The comparison in temporal variation between the enhanced and clean speech is performed similarly to [57]. In the DNN training stage, the output feature vectors are transformed into the one-third octave band domain before the short-time segmentation and variation analysis are performed to obtain the temporal variation similarity for each slice of frames. Then, we incorporate the temporal variation similarity values to the objective function and compute the gradients from them.

The enhanced and clean log-power spectra $\bar{\mathbf{x}}_t$ and $\bar{\mathbf{y}}_t$ are transformed to the

l_H-dimensional one-third octave band vectors $\tilde{\mathbf{x}}_t$ and $\tilde{\mathbf{y}}_t$ as follows:

$$\tilde{\mathbf{x}}_t = \sqrt{B\ exp(\bar{\mathbf{x}}_t)}, \tag{6.15}$$

$$\tilde{\mathbf{y}}_t = \sqrt{B\ exp(\bar{\mathbf{y}}_t)} \tag{6.16}$$

where B denotes the $l_B \times l_F$-dimensional one-third octave band matrix, and $exp(\mathbf{x})$ and $\sqrt{\mathbf{x}}$ denote the element-wise exponential and square root functions of a vector $\mathbf{x}$, respectively. The temporal variation similarity is computed only for the speech active frames. To remove the speech absence frames from the variation analysis, a simple decision rule is applied to $\tilde{\mathbf{y}}_t$ as in [57].

The variation analysis is performed for each one-third octave band and each slice of N speech active frames. Let us denote the vectors stacking the h-th one-third octave band coefficients from the t-th frame to the $t+N-1$-th frame of the enhanced and clean speech as $\tilde{\mathbf{X}}_{t,h}$ and $\tilde{\mathbf{Y}}_{t,h}$. Then, the temporal variation similarity between $\tilde{\mathbf{X}}_{t,h}$ and $\tilde{\mathbf{Y}}_{t,h}$ is defined as follows:

$$\rho_1(t,h) = \frac{(\tilde{\mathbf{X}}_{t,h} - \mu_{\tilde{\mathbf{X}}_{t,h}} \mathbf{1}_N)^{\dagger} (\tilde{\mathbf{Y}}_{t,h} - \mu_{\tilde{\mathbf{Y}}_{t,h}} \mathbf{1}_N)}{||\tilde{\mathbf{X}}_{t,h} - \mu_{\tilde{\mathbf{X}}_{t,h}} \mathbf{1}_N||\ ||\tilde{\mathbf{Y}}_{t,h} - \mu_{\tilde{\mathbf{Y}}_{t,h}} \mathbf{1}_N||} \tag{6.17}$$

where $\mathbf{1}_N$ denotes an N-dimensional vector with all elements being 1 and

$$\mu_{\tilde{\mathbf{X}}_{t,h}} = \frac{1}{N} \sum_{i=1}^{N} \tilde{\mathbf{X}}_{t,h}(i), \tag{6.18}$$

$$\mu_{\tilde{\mathbf{Y}}_{t,h}} = \frac{1}{N} \sum_{i=1}^{N} \tilde{\mathbf{Y}}_{t,h}(i). \tag{6.19}$$

The proposed objective function incorporates the variation similarity $\rho_1(t,h)$ averaged over a time-frequency window as given by

$$\rho_1 = \frac{1}{(T-N+1)H} \sum_{t=1}^{T-N+1} \sum_{h=1}^{H} \rho_1(t,h). \tag{6.20}$$

By training the DNN while considering this sub-cost, the short-time trajectories of the enhance speech would have temporal variation more similar to those of the clean speech.

6.3.4 Spectral Variation Similarity ρ_2

The speech generated by the enhancement algorithms would suffer from the muffled effect when the spectral peaks and valleys are over-smoothed [48]. In order to improve the spectral dynamics of the enhanced speech, we also introduce a variation over the frequency bins, which results in a better contrast between the spectral peaks and valleys.

The spectral variation similarity ρ_2 is derived in a similar manner to ρ_1. However, compared to ρ_1, ρ_2 is different in two aspects. First, ρ_2 is derived in the spectral magnitude domain without the one-third octave band analysis. Second, while ρ_1 considers speech trajectory and disregards the variation over different frequency bins, ρ_2 aims to adjust the spectral peaks and valleys in the same time frame.

The l_F-dimensional enhanced and clean speech magnitude spectra $\tilde{\mathbf{x}}'_t$ and $\tilde{\mathbf{y}}'_t$ are obtained as follows:

$$\begin{aligned} \tilde{\mathbf{x}}'_t &= \sqrt{exp(\bar{\mathbf{x}}_t)}, && (6.21) \\ \tilde{\mathbf{y}}'_t &= \sqrt{exp(\bar{\mathbf{y}}_t)}. && (6.22) \end{aligned}$$

The spectral variation similarity is computed only over the speech active frames. The variation similarity $\rho_2(t)$ computed at the t-th frame is given by

$$\rho_2(t) = \frac{(\tilde{\mathbf{x}}'_t - \mu_{\tilde{\mathbf{x}}'_t}\mathbf{1}_F)^\dagger(\tilde{\mathbf{y}}'_t - \mu_{\tilde{\mathbf{y}}'_t}\mathbf{1}_F)}{||\tilde{\mathbf{x}}'_t - \mu_{\tilde{\mathbf{x}}'_t}\mathbf{1}_F||\ ||\tilde{\mathbf{y}}'_t - \mu_{\tilde{\mathbf{y}}'_t}\mathbf{1}_F||} \tag{6.23}$$

with

$$\mu_{\tilde{\mathbf{x}}'_t} = \frac{1}{F}\sum_{i=1}^{F}\tilde{\mathbf{x}}'_t(i), \tag{6.24}$$

$$\mu_{\tilde{\mathbf{y}}'_t} = \frac{1}{F}\sum_{i=1}^{F}\tilde{\mathbf{y}}'_t(i). \tag{6.25}$$

Then, ρ_2 is obtained by averaging $\rho_2(t)$ over all frames i.e.,

$$\rho_2 = \frac{1}{T}\sum_{t=1}^{T}\rho_2(t). \tag{6.26}$$

6.3.5 DNN Training with the Proposed Objective Function

In the training stage with the proposed objective function, the derivative of the objective function with respect to each network output $\frac{\partial C}{\partial \mathbf{x}_t(f)}$ is computed and used to derive the gradient with respect to each parameter through back-propagation. In Appendix, we provide the details on the derivation of $\frac{\partial C}{\partial \mathbf{x}_t(f)}$.

6.4 Experiments

In order to evaluate the performance of the proposed algorithm, we conducted experiments in matched and mismatched noise conditions. In the experiments, 4,620 utterances of clean speech data were taken from the TIMIT training database to train the DNN. The {con_mono_1, met_mono_1, off_mono_1, car_mono_1, rai_mono_1, res_mono_1, train, traffic} noises from ITU-T recommendation P.501 database [42] and the {white, factory, babble, machinegun} noises from NOISEX-92 database [41] were used for training. Each noise waveform was re-sampled to 16 kHz, and we chose the left channel of the binaural noise recordings in ITU-T recommendation P.501 database. For each pair of the clean speech utterance and noise waveform, a noisy speech utterance was artificially generated with an SNR value randomly chosen from {-5, 0, 5, 10, 15, 20} dB. A 512-point Hamming window with 50% overlap was applied. F and τ were fixed to 257 and 5, respectively (feature vectors extracted from 11 consecutive frames were concatenated similarly to [48], [50]).

For the test set, 30 utterances of clean speech data were taken randomly from the TIMIT test database. The {con_mono_1, res_mono_1} noises from ITU-T recommendation P.501 database and the white noise from NOISEX-92 DB were used for the experiment in matched noise conditions. For the experiment in mismatched noise conditions, the {cafeteria, kids, street} noises from ITU-T recommendation P.501 DB were chosen. For each pair of the clean speech utterance and the noise waveform, the noisy speech utterances were artificially generated with the SNR ranging from -5 to 10 dB with 5 dB step.

The DNNs were implemented using the Theano neural network toolkit [24]. The DNNs were constructed by stacking 3 hidden layers with 2048 nodes each. The numbers of the input and output nodes were $257 \times 11 = 2827$ and 257, respectively. All networks were trained through 50 epochs. The learning rate was fixed to 0.003 in the first 10 epochs and decreased by 10% after each subsequent epoch. The momentum rate was 0.5 for the first 5 epochs and increased to 0.9 afterward. The dropout rates of the input layer and all hidden layers were set to 0.1 and 0.2, respectively. The mini-batch size was equal to the number of frames in each utterance. The average value of mini-batch size was 190.6, and N was fixed to 30.

In the experiments, the enhanced speech signals obtained from DNNs with various training objective functions and GV equalization post-filter were compared in both objective measures and subjective test. The performance of the DNN-based algorithm with the proposed techniques was compared to that with the conventional mean square error and frequency-independent GV equalization post-filter [48].

The perceptual evaluation of speech quality (PESQ) score [44] and the short-time objective intelligibility (STOI) value [57] were used for the objective measures. For the subjective measures, a preference test was conducted with the enhanced speech obtained in the mismatched noise conditions.

6.4.1 Performance Evaluation with Varying Weight Parameters

First, we evaluated how the variation similarities ρ_1 and ρ_2 affect the performance of the enhancement algorithm by varying weight parameters λ_1 and λ_2 in (6.10). In this experiment, we measured PESQ scores and STOI values of the enhanced speech while varying λ_1 and λ_2.

Table 6.1 shows the PESQ scores and STOI values averaged over all SNR values and noise types in the matched noise conditions. The results show that both the PESQ scores and STOI values gradually increased as λ_1 and λ_2 became larger. From the results, we can see that the proposed variation similarities are useful for the DNN to generate more natural and intelligible speech. In all the following experiments, we fixed λ_1 and λ_2 to 5 which demonstrated a good performance.

6.4.2 Performance Evaluation in Matched Noise Conditions

In this experiment, the performances of various configurations of the DNN were compared in the matched noise conditions. Tables 6.2 and 6.3 show the PESQ scores and STOI values obtained in matched noise conditions. From the results, it is shown that employing the Mel-scale weighted mean square error improved both the perceptual quality and intelligibility of the enhanced speech. This result demonstrates that adopting perceptually motivated non-linear frequency scale to the objective function

Table 6.1: Results of average PESQ scores and STOI values of the proposed objective function C and GV equalization post-filter with varying weight parameters in matched noise conditions.

	λ_1	λ_2	PESQ	STOI
With C_{mse}	0	0	2.55	0.80
With C_{wmse}	0	0	2.64	0.81
	0.5	0.5	2.72	0.84
	1	1	2.76	0.84
	2	2	2.78	**0.85**
	5	5	**2.80**	**0.85**
	10	10	**2.80**	**0.85**
Without C_{wmse}	10	10	**2.80**	**0.85**

improves the quality of the enhanced speech.

Moreover, incorporating the variation similarities into the DNN training objective function further improved the performance in terms of both PESQ score and STOI value. In the case of PESQ score, the performance of the DNN was improved with the use of ρ_1 and ρ_2. On the other hand, it turned out that ρ_1 played more important role to improve the STOI values particularly in low SNR conditions than ρ_2. These results were consistent with the previous studies which reported that the temporal variation is more important in speech intelligibility.

6.4.3 Performance Evaluation in Mismatched Noise Conditions

In this experiment, the performances of various algorithms were compared in the mismatched noise conditions. Tables 6.4 and 6.5 show the PESQ scores and STOI values obtained in various mismatched noise conditions. From the results, we can see that the amount of improvement in both the quality and intelligibility was less than that achieved in the matched noise condition. However, the DNN-based speech enhancement algorithm with the proposed objective function still outperformed the conventional algorithm in unseen noise conditions. The PESQ scores of the enhanced speech were improved by employing the Mel-scale weighted mean square error and variation similarities.

It is interesting to see that the incorporation of the spectral variation similarity ρ_2 slightly decreased the enhancement performance in the Kids noise environment. This may be due to the characteristics of the Kids noise which has similar spectral shape to the target speech. Since ρ_2 emphasized the spectral peaks and valleys, it also made the speech-like noise slightly more noticeable after speech enhancement.

As shown in Table 6.5, the STOI values were enhanced by incorporating ρ_1 to the objective function while other techniques did not show any significant effects on intelligibility prediction score. This result once again confirms that the temporal variation of the enhanced speech is more crucial than the spectral variation in terms of the speech intelligibility.

6.4.4 Comparison Between Variation Analysis Method

Next, we compared the performance of the proposed objective function with Pearson correlation coefficients (6.17) and (6.23) and those with the mean square error which are defined as follows:

$$\begin{aligned}
\rho_1(t,h) &= \mathbf{1}_N^{\dagger}\left(\frac{\tilde{\mathbf{X}}_{t,h}-\mu_{\tilde{\mathbf{X}}_{t,h}}\mathbf{1}_N}{||\tilde{\mathbf{X}}_{t,h}-\mu_{\tilde{\mathbf{X}}_{t,h}}\mathbf{1}_N||}-\frac{\tilde{\mathbf{Y}}_{t,h}-\mu_{\tilde{\mathbf{Y}}_{t,h}}\mathbf{1}_N}{||\tilde{\mathbf{Y}}_{t,h}-\mu_{\tilde{\mathbf{Y}}_{t,h}}\mathbf{1}_N||}\right)^2 \qquad (6.27)\\
\rho_2(t) &= \mathbf{1}_F^{\dagger}\left(\frac{\tilde{\mathbf{x}}'_t-\mu_{\tilde{\mathbf{x}}'_t}\mathbf{1}_F}{||\tilde{\mathbf{x}}'_t-\mu_{\tilde{\mathbf{x}}'_t}\mathbf{1}_F||}-\frac{\tilde{\mathbf{y}}'_t-\mu_{\tilde{\mathbf{y}}'_t}\mathbf{1}_F}{||\tilde{\mathbf{y}}'_t-\mu_{\tilde{\mathbf{y}}'_t}\mathbf{1}_F||}\right)^2 \qquad (6.28)
\end{aligned}$$

where $\mathbf{x}^2$ denotes the element-wise square function of the vector $\mathbf{x}$. We compared the average PESQ scores and STOI values of the proposed algorithm with various analysis method in the mismatched noise environments.

Table 6.6 and 6.7 shows the performance of the proposed algorithm with various parameters and analysis methods in the mismatched noise environments. From the results, we can see that the performance of the proposed algorithm with mean square error analysis showed similar results with those of the correlation analysis. Since the derivative of correlation and mean square error functions are similar, the performances of the objective functions would be also similar when the related parameters were optimized. However, we chose the correlation function since it showed slightly better results than the mean square error function.

6.4.5 Subjective Test Results

Finally, we performed a subjective listening test to compare the performance of the proposed techniques with that of the conventional objective function. Ten listeners participated and were presented with 45 randomly selected sentences in the SNR range of {-5, 0, 5} dB corrupted by the {cafeteria, kids, street} noises. In the test, each listener was provided with speech samples enhanced by the network with or without the proposed technique. The listeners were asked to choose the preferred one for each pair of speech samples in terms of perceptual speech quality. Two samples in each pair were given in arbitrary order.

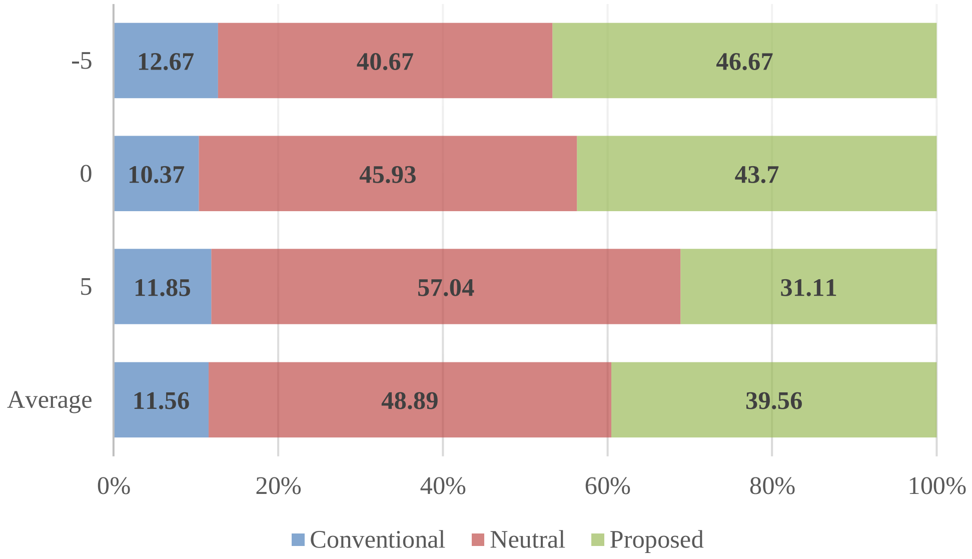

Figure 6.5: Results of preference test (%) between the conventional and proposed algorithms in the speech quality in various SNR values.

The results are shown in Fig. 6.4——. It can be seen that the quality of the speech enhanced by the proposed algorithm was better than that using the conventional algorithm in all SNR values. This result implies that the proposed algorithm provides enhanced speech which is more comfortable to the human listener. From these experiments, we can conclude that the proposed objective function is effective for improving not only the objective but also the subjective speech quality.

6.5 Summary

In this chapter, we have proposed a novel objective function for DNN-based speech enhancement to equalize the temporal and spectral variations of the enhanced speech. The proposed algorithm incorporates the perceptually motivated non-linear frequency weight and variation similarities between the enhanced and clean speech spectral trajectories. From the experimental results, it has been found that the proposed algorithm outperformed the conventional DNN-based speech enhancement algorithm in terms of the objective measures as well as the subjective listening quality.

The future work will focus on employing novel model structures and training techniques. The recent studies show that the performance of the deep learning models could be further improved by the proper training scheme [62]–[64]. Also, as in speech recognition, the sequence-to-sequence model such as the long-short term memory (LSTM) or gated recurrent unit (GRU) [65], [66] may be better to describe the speech characteristics. Finally, the spatial information of the target and background noise will be useful to improve the speech quality in the interfering speaker environments such as the Kids noise.

Table 6.2: Results of PESQ scores of various algorithms in matched noise conditions.

SNR (dB)		-5			0		
Objective function	Post-filter	White	Res.	Con.	White	Res.	Con.
unprocessed		1.24	1.32	1.28	1.53	1.64	1.63
C_{mse}	-	1.91	1.70	2.06	2.41	2.14	2.48
C_{mse}	GV	1.91	1.69	2.07	2.45	2.19	2.54
$C\ (\lambda_1 = 0, \lambda_2 = 0)$	GV	2.03	1.77	2.15	2.55	2.31	2.64
$C\ (\lambda_1 = 5, \lambda_2 = 0)$	GV	2.13	1.91	2.29	2.58	2.40	2.72
$C\ (\lambda_1 = 0, \lambda_2 = 5)$	GV	2.11	1.88	2.22	2.62	2.43	2.71
$C\ (\lambda_1 = 5, \lambda_2 = 5)$	GV	**2.18**	**2.02**	**2.38**	**2.65**	**2.52**	**2.82**
SNR (dB)		5			10		
Objective function	Post-filter	White	Res.	Con.	White	Res.	Con.
unprocessed		1.87	2.04	2.08	2.23	2.42	2.39
C_{mse}	-	2.76	2.55	2.84	3.03	2.88	3.04
C_{mse}	GV	2.85	2.64	2.95	3.15	3.01	3.15
$C\ (\lambda_1 = 0, \lambda_2 = 0)$	GV	2.91	2.74	3.04	3.22	3.11	3.25
$C\ (\lambda_1 = 5, \lambda_2 = 0)$	GV	2.93	2.81	3.07	3.23	3.14	3.28
$C\ (\lambda_1 = 0, \lambda_2 = 5)$	GV	2.97	2.84	3.15	3.31	3.22	3.34
$C\ (\lambda_1 = 5, \lambda_2 = 5)$	GV	**3.00**	**2.92**	**3.16**	**3.32**	**3.24**	**3.39**

Table 6.3: Results of STOI values of various algorithms in matched noise conditions.

SNR (dB)		-5			0		
Objective function	Post-filter	White	Res.	Con.	White	Res.	Con.
unprocessed		0.58	0.51	0.56	0.71	0.64	0.67
C_{mse}	-	0.66	0.61	0.69	0.78	0.74	0.81
C_{mse}	*GV*	0.66	0.61	0.69	0.79	0.75	0.82
$C\ (\lambda_1 = 0, \lambda_2 = 0)$	*GV*	0.68	0.62	0.70	0.80	0.77	0.83
$C\ (\lambda_1 = 5, \lambda_2 = 0)$	*GV*	**0.75**	0.71	0.76	0.83	0.81	0.86
$C\ (\lambda_1 = 0, \lambda_2 = 5)$	*GV*	0.71	0.63	0.72	0.82	0.78	0.84
$C\ (\lambda_1 = 5, \lambda_2 = 5)$	*GV*	**0.75**	**0.71**	**0.77**	**0.84**	**0.82**	**0.87**
SNR (dB)		5			10		
Objective function	Post-filter	White	Res.	Con.	White	Res.	Con.
unprocessed		0.82	0.75	0.78	0.91	0.85	0.85
C_{mse}	-	0.86	0.83	0.86	0.91	0.88	0.90
C_{mse}	*GV*	0.87	0.84	0.87	0.92	0.89	0.91
$C\ (\lambda_1 = 0, \lambda_2 = 0)$	GV	0.88	0.85	0.89	0.93	0.91	0.92
$C\ (\lambda_1 = 5, \lambda_2 = 0)$	*GV*	**0.90**	0.87	0.90	**0.94**	**0.92**	0.93
$C\ (\lambda_1 = 0, \lambda_2 = 5)$	*GV*	0.89	0.86	0.90	**0.94**	**0.92**	0.93
$C\ (\lambda_1 = 5, \lambda_2 = 5)$	*GV*	**0.90**	**0.88**	**0.91**	**0.94**	**0.92**	**0.94**

Table 6.4: Results of PESQ scores of various algorithms in mismatched noise conditions.

SNR (dB)		-5			0		
Objective function	Post-filter	Cafe.	Kids	Str.	Cafe.	Kids	Str.
unprocessed		1.43	1.27	1.66	1.75	1.71	1.99
C_{mse}	-	1.61	1.66	1.86	2.04	1.98	2.35
C_{mse}	*GV*	1.63	1.68	1.87	2.09	2.06	2.42
$C\ (\lambda_1 = 0, \lambda_2 = 0)$	*GV*	1.69	1.74	1.95	2.21	2.11	2.51
$C\ (\lambda_1 = 5, \lambda_2 = 0)$	*GV*	1.72	**1.78**	2.04	2.24	**2.14**	2.56
$C\ (\lambda_1 = 0, \lambda_2 = 5)$	*GV*	1.74	1.72	1.91	2.28	2.08	2.55
$C\ (\lambda_1 = 5, \lambda_2 = 5)$	*GV*	**1.80**	1.73	**2.09**	**2.31**	2.11	**2.62**
SNR (dB)		5			10		
Objective function	Post-filter	Cafe.	Kids	Str.	Cafe.	Kids	Str.
unprocessed		2.13	2.11	2.25	2.50	2.38	2.63
C_{mse}	-	2.50	2.46	2.63	2.86	2.68	2.96
C_{mse}	*GV*	2.59	2.55	2.74	2.97	2.79	3.08
$C\ (\lambda_1 = 0, \lambda_2 = 0)$	*GV*	2.67	2.62	2.83	3.06	2.86	3.18
$C\ (\lambda_1 = 5, \lambda_2 = 0)$	*GV*	2.66	**2.59**	2.87	3.03	**2.86**	3.20
$C\ (\lambda_1 = 0, \lambda_2 = 5)$	*GV*	2.73	2.56	2.89	**3.12**	2.85	**3.26**
$C\ (\lambda_1 = 5, \lambda_2 = 5)$	*GV*	**2.74**	2.54	**2.91**	3.10	2.84	**3.26**

Table 6.5: Results of STOI values of various algorithms in mismatched noise conditions.

SNR (dB)		-5			0		
Objective function	Post-filter	Cafe.	Kids	Str.	Cafe.	Kids	Str.
unprocessed		0.54	0.57	0.66	0.65	0.70	0.76
C_{mse}	-	0.58	0.65	0.65	0.72	0.77	0.80
C_{mse}	*GV*	0.58	0.65	0.66	0.73	0.78	0.81
$C\ (\lambda_1 = 0, \lambda_2 = 0)$	*GV*	0.59	0.66	0.67	0.74	0.79	0.82
$C\ (\lambda_1 = 5, \lambda_2 = 0)$	*GV*	0.63	**0.67**	**0.74**	0.77	**0.80**	0.85
$C\ (\lambda_1 = 0, \lambda_2 = 5)$	*GV*	0.60	0.65	0.68	0.76	0.79	0.83
$C\ (\lambda_1 = 5, \lambda_2 = 5)$	*GV*	**0.64**	0.66	**0.74**	**0.78**	0.79	**0.86**
SNR (dB)		5			10		
Objective function	Post-filter	Cafe.	Kids	Str.	Cafe.	Kids	Str.
unprocessed		0.77	0.83	0.83	0.85	0.89	0.90
C_{mse}	-	0.82	0.86	0.85	0.88	0.90	0.91
C_{mse}	*GV*	0.83	0.88	0.86	0.89	0.91	0.92
$C\ (\lambda_1 = 0, \lambda_2 = 0)$	*GV*	0.84	**0.89**	0.88	0.90	**0.93**	0.93
$C\ (\lambda_1 = 5, \lambda_2 = 0)$	*GV*	0.86	**0.89**	0.89	0.91	**0.93**	**0.94**
$C\ (\lambda_1 = 0, \lambda_2 = 5)$	*GV*	0.86	**0.89**	0.89	0.91	**0.93**	**0.94**
$C\ (\lambda_1 = 5, \lambda_2 = 5)$	*GV*	**0.87**	**0.89**	**0.90**	**0.92**	**0.93**	**0.94**

Table 6.6: Results of PESQ scores of various analysis methods averaged over various mismatched noise conditions.

		SNR (dB)			
Objective function	Post-filter	-5	0	5	10
$C\ (\lambda_1 = 1, \lambda_2 = 1)$	MSE	1.81	2.29	2.72	3.04
$C\ (\lambda_1 = 5, \lambda_2 = 5)$	MSE	1.83	2.31	2.72	3.05
$C\ (\lambda_1 = 20, \lambda_2 = 20)$	MSE	**1.87**	2.33	**2.73**	3.06
$C\ (\lambda_1 = 5, \lambda_2 = 5)$	Corr.	**1.87**	**2.35**	**2.73**	**3.07**

Table 6.7: Results of STOI values of various analysis methods averaged over various mismatched noise conditions.

		SNR (dB)			
Objective function	Post-filter	-5	0	5	10
$C\ (\lambda_1 = 1, \lambda_2 = 1)$	MSE	0.65	0.79	0.87	0.92
$C\ (\lambda_1 = 5, \lambda_2 = 5)$	MSE	0.67	0.80	0.88	**0.93**
$C\ (\lambda_1 = 20, \lambda_2 = 20)$	MSE	**0.68**	0.80	0.88	**0.93**
$C\ (\lambda_1 = 5, \lambda_2 = 5)$	Corr.	**0.68**	**0.81**	**0.89**	**0.93**

Chapter 7

Spectral Variance Equalization Post-filter for DNN-Based Speech Enhancement

7.1 Introduction

The speech generated from the conventional DNN-based speech enhancement algorithm is suffered by the muffled effect which is caused by the over-smoothed spectrum from the network output. In order to improve the perceptual quality of the enhanced speech, early studies applied the global variance (GV) equalization post-filter which roughly matches the standard deviation of the DNN output with that of the original speech [48], [67]. In [67], the authors showed that the GV of the enhanced feature is lower than that of the original feature and the problem would get worse in the low-SNR conditions. The enhanced speech generated from the GV equalization post-filter showed better performance with affordable computational cost.

While the GV equalization post-filters partly compensate the lack of variation in the DNN output, their ability to recover the original contrast between the spectral peaks and valleys are not fully studied. In this chapter, we propose the spectral variance (SV) equalization post-filter which directly adjusts the variance over frequency bins in the log-power spectra domain. By adjusting the variance over frequency bins, the mismatch in spectral dynamics, the mismatch between spectral contrasts of the estimated and clean speech which cannot be fully compensated by the perceptually-motivated objective function is further reduced.

7.2 GV Equalization Post-Filter

As explained in the previous chapter, the GV equalization post-filter can be divided into the frequency-dependent and the frequency-independent algorithm. In the frequency-independent GV equalization, the GV factor is defined as (6.6) to (6.8).

In the frequency dependent GV, the GV for each dimension $\alpha_{dep}(d)$ is derived as follows:

$$GV_{dep}(\mathbf{x}, d) = \frac{1}{T}\sum_{t=1}^{T}(\mathbf{x}_t(d) - \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t(d))^2, \tag{7.1}$$

$$GV_{dep}(\mathbf{y}, d) = \frac{1}{T}\sum_{t=1}^{T}(\mathbf{y}_t(d) - \frac{1}{T}\sum_{t=1}^{T}\mathbf{y}_t(d))^2, \tag{7.2}$$

$$\alpha_{dep}(d) = \sqrt{\frac{GV_{dep}(\mathbf{y}, d)}{GV_{dep}(\mathbf{x}, d)}}. \tag{7.3}$$

In the case of the frequency-dependent GV, the GV factor vector is multiplied to the network output element-wisely.

Since the value of the elements in α_{dep} could be fluctuant especially in the very high or low frequency bands, the average of the frequency-dependent GV factor $\bar{\alpha}_{dep}$ could be used as a GV factor rather than using α_{dep} directly. Previous studies on the GV equalization post-filter evaluated these three GV factors in various SNR conditions [67]. They showed that while the post-filter with $\bar{\alpha}_{dep}$ was slightly better than other post-filters in terms of PESQ scores [44], the performance difference among these three algorithms were not significant.

7.3 Spectral Variance (SV) Equalization Post-Filter

The conventional frequency-dependent and -independent GV equalization post-filters enhance the over-smoothed feature trajectory by appropriately scaling the output vectors. In these post-filter algorithms, the GV factors are obtained from the frequency-dependent or -independent variances of the output feature. While the conventional frequency-dependent and -independent GV algorithms partly alleviate the over-smoothing problem of the DNN [48], these GV factors do not consider the spectral dynamics over frequency bins. To consider the contrast between spectral peaks and valleys in each enhanced speech frames, we propose the spectral variance (SV) equalization post-filter which matches the variance over spectral coefficients in the log-power spectra domain.

Similar to the procedures for calculating the temporal and spectral variation similarities in the last chapter, the proposed SV post-filter algorithm discards the speech absence frames from its derivation for SV factor. The SV factor α_{SV} is defined by the average ratio of the variation in the log-power spectral domain as given by

$$\alpha_{SV} = \frac{1}{T}\sum_{t=1}^{T} \frac{||\bar{\mathbf{y}}_t - \mu_{\bar{\mathbf{y}}_t}||}{||\bar{\mathbf{x}}_t - \mu_{\bar{\mathbf{x}}_t}||}. \tag{7.4}$$

In the test stage, the proposed post-filter is applied after $\mathbf{x}_t$ is transformed to the log-power spectra domain as follows:

$$\bar{\mathbf{x}}_t = \alpha_{SV}(\mathbf{x}_t \otimes \mathbf{s} + \mathbf{m}). \tag{7.5}$$

Similar to the conventional GV factors, the SV factor in the log-power spectra domain operates as an exponential factor in the spectral magnitude domain.

7.4 Experiments

In order to evaluate the performance of the proposed post-filter, we conducted experiments on the speech generated from the DNN with the perceptually-motivated objective function proposed in the previous chapter. The enhanced speech without the post-filter, GV and SV equalization post-filters are compared in terms of the PESQ scores and subjective preference test.

The DNN structure, training scheme, and the training and test dataset are kept to be same with the experiment in Chapter 6. In the experiments, the GV and SV factors of the post-filters are estimated from the DNN training database. We consider the baseline model as the DNN trained by perceptually-motivated criteria and the conventional GV equalization filter. The proposed post-filter is compared to the baseline model to show if further objective and subjective performance improvement could be achieved by the proposed post-processing algorithm.

7.4.1 Objective Test Results

First, we evaluated the performance of the conventional and proposed post-filters in the matched and mismatched conditions in terms of the PESQ scores. The STOI values of the post-processed speech are not presented since both the GV and SV

Table 7.1: Results of PESQ scores of various algorithms in matched noise conditions.

SNR (dB)	-5			0			5			10		
Post-filter	White	Res.	Con.	White	Res.	Con.	White	Res.	Con.	White	Res.	Con.
–	2.12	1.98	2.33	2.55	2.45	2.74	2.89	2.81	3.06	3.19	3.13	3.28
GV	2.18	2.02	2.38	2.65	2.52	2.82	3.00	2.92	3.16	3.32	3.24	3.39
SV	**2.23**	**2.03**	**2.42**	**2.70**	**2.54**	**2.86**	**3.05**	**2.95**	**3.21**	**3.36**	**3.27**	**3.40**

algorithm since they did not enhance the STOI values of the enhanced speech. This results were consistent with the previous studies [54], [55] that the intelligibility of the speech is highly related to the temporal variation of the enhanced speech.

Tables 7.1 and 7.2 show the performance of PESQ scores in the matched and mismatched conditions. From the experiments, we can see that the perceptual quality of the enhanced speech is increased by adopting post-filters to the DNN-based enhancement algorithm. Also, the performance of the proposed post-filter outperformed the conventional GV equalization post-filter in all SNR values and noise conditions. From the results, we can see that the proposed SV equalization post-filter can compensate the lack of spectral variance from the DNN-based algorithm more effectively.

7.4.2 Subjective Test Results

Next, we performed a subjective listening test to compare the performance of the proposed techniques with that of the conventional objective function and post-filter. Ten listeners participated and were presented with 45 randomly selected sentences in the SNR range of { 0, 5, 10} dB corrupted by the {cafeteria, kids, street} noises. In

Table 7.2: Results of PESQ scores of various algorithms in mismatched noise conditions.

SNR (dB)	-5			0			5			10		
Post-filter	Cafe.	Kids	Str.	Cafe.	Kids	Str.	Cafe.	Kids	Str.	Cafe.	Kids	Str.
–	1.78	1.72	2.06	2.26	2.06	2.54	2.67	2.50	2.81	3.02	2.78	3.14
GV	1.80	1.73	2.09	2.31	2.11	2.62	2.74	2.54	2.91	3.10	2.84	3.26
SV	**1.83**	**1.79**	**2.11**	**2.36**	**2.17**	**2.66**	**2.78**	**2.62**	**2.96**	**3.15**	**2.92**	**3.31**

the test, each listener was asked to compare the speech samples enhanced by network with the objective function proposed in the previous chapter and SV equalization post-filter to that with the conventional objective function and GV equalization post-filter. Two samples in each pair were given in arbitrary order. The listers chose the perceptually better sound samples or neutral when the quality of two sound samples are similar.

The results are shown in Fig. 7.1. It can be seen that the quality of the speech enhanced by the proposed algorithm was better than that using the conventional algorithm in all SNR values. Also, comparing the results in Figures 6.4 and 7.1, the proposed post-filter further improved the performance of the proposed algorithm compared to those without the SV equalization post-filter or the perceptually motivated objective function. This results shows that the improvement from the perceptually-motivated objective function and the SV equalization post-filter can be easily combined for perceptually natural enhanced speech.

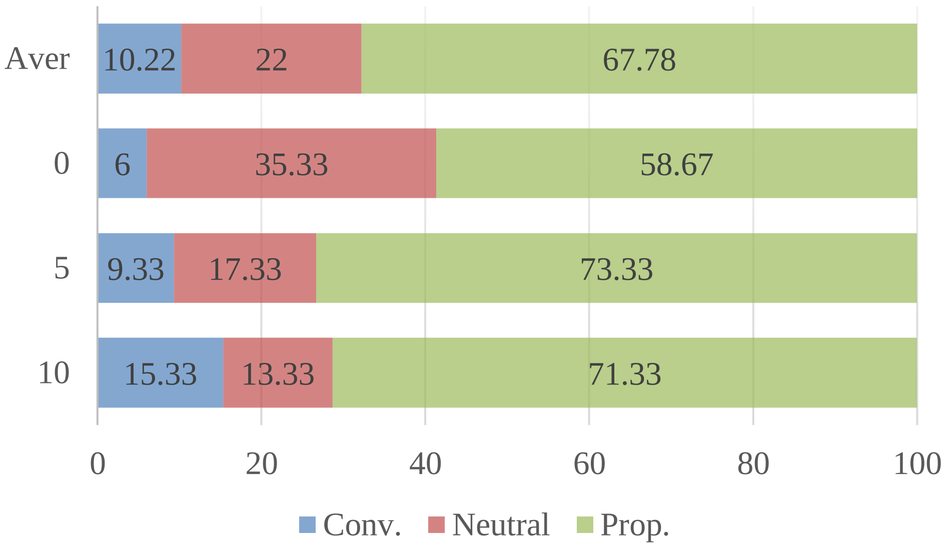

Figure 7.1: Results of preference test (%) between the conventional and proposed algorithms in the speech quality in various SNR values.

7.5 Summary

In this chapter, we have proposed the SV equalization post-filter for DNN-based speech enhancement algorithm. The proposed post-filter directly matches the variance over frequency bins in log-spectral domains to recover the spectral variation of the enhanced speech. In the experiment, the proposed algorithm is compared with the conventional GV equalization post-filter and showed better performance. The experiment also shows that combining the perceptually-motivated objective function and the SV equalization post-filter can further improve the perceptual speech quality.

Chapter 8

Conclusions

In this thesis, the deep learning-based approaches for the robust VAD and speech enhancement have been proposed. The scheme of the DNN-based VAD and speech enhancement algorithms which estimate the speech presence interval and the corresponding clean speech waveform, respectively, are introduced. In the VAD area, the performance degradation of the DNN-based VAD in the mismatched noise environments is compensated by adopting multi-task learning in the training stage. In the speech enhancement area, the DNN-based algorithm is defined and the knowledge on the human auditory perception is adopted to the DNN objective function and post-filter.

Firstly, we have proposed a novel approach to increase the robustness of the DNN-based VAD in the mismatched noise environments. To learn the general denoising function which could be applied to various noise environments, the subsidiary feature enhancement task is combined to the conventional DNN-based VAD and the parameters of the DNN are optimized by the gradients from both the conventional VAD and feature enhancement tasks. From the experiment results, the performance of the DNN with the MTL framework showed better performance than the conventional DNN in the mismatched noise environments.

Secondly, we have proposed a speech enhancement algorithm which estimate the encoding vectors of the NMF analysis using DNN. The noisy speech waveforms are artificially generated and the corresponding speech and noise encoding vectors are estimated with the DNN. In the proposed algorithm, the complicated nonlinear relations between the speech and noise basis vectors are represented by the DNN. From the experiment results, the performance of the proposed algorithm outperformed the conventional and discriminative NMF-based algorithms.

Thirdly, we have incorporated the nonlinear characteristics of human auditory perception in the DNN objective function. Also, we alleviated the over-smoothness and improved the human intelligibility of the enhanced speech by considering temporal and spectral variations. In the experiments, we showed that the proposed objective function improved the performance of the DNN-based speech enhancement algorithm in terms of both the objective and subjective measures.

Finally, we have proposed a novel post-filter algorithm which matches the spectral variance of the enhanced speech to that of the clean speech. The proposed post-filter emphasizes the spectral peaks and valleys of the enhanced speech to reduce the muffled effect. In the experiments, the proposed post-filter showed better performance than the conventional GV equalization post-filter in terms of the PESQ scores. The subjective results showed that the proposed perceptually-motivated objective function and post-filters could be combined to generate more perceptually comfortable speech.

Bibliography

[1] Y. Ephraim and D. Malah, "Speech enhancement using minimum mean square log spectral amplitude estimator," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-33, no. 2, pp. 443–445, Apr. 1985.

[2] J. Sohn, N. S. Kim, and W. Sung, "A statistical model based voice activity detection," *IEEE Signal Process. Lett.*, vol. 6, no. 1, pp. 1–3, Jan. 1999.

[3] I. Cohen and B. Berdugo, "Speech enhancement for non-stationary noise environments," *Signal Process.*, vol. 81, no. 11, pp. 2403–2418, 2001.

[4] I. Cohen, "Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging," *IEEE Trans. Speech, Audio Process.*, vol. 11, no. 5, pp. 465–475, Nov. 2003.

[5] D. D. Lee and H. S. Seung, "Learning the parts of objevts by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[6] K. W. Wilson, B. Raj, P. Smargadis, and A. Divakaran, "Speech denoising using nonnegative matrix factorization with priors," in *Proc. ICASSP*, 2008, pp. 4029–4032.

[7] R. A. Caruana, "Multitask connectionist learning," in *Connectionist Models Summer School*, 1993, pp. 372–379.

[8] P. Bell and S. Renals, "Regularization of context-dependent deep neural networks with context-independent multi-task learning," in *Proc. ICASSP*, 2015, pp. 4290–4294.

[9] R. Giri, M. L. Seltzer, J. Droppo, and D. Yu, "Improving speech recognition in reverberation using a room-aware deep neural network and multi-task learning," in *Proc. ICASSP*, 2015, pp. 5014–5018.

[10] D. Chen and B. K. W. Mak, "Multitask learning of deep neural networks for low-resource speech recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 7, pp. 1172–1183, Jul. 2015.

[11] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Advances in Neural Inform. Process. Syst.*, 2000, pp. 556–562.

[12] F. Weninger, J. Geiger, M. Wöllmer, B. Schuller, and G. Rigoll, "The munich 2011 chime challenge contribution: Nmf-blstm speech enhancement and recognition for reverberated multisource environments," in *Proc. 1st Int. Workshop on Mach. Listening in Multisource Environments (CHiME)*, 2011, pp. 24–29.

[13] E. M. Grais and H. Erdogan, "Single channel speech music separation using nonnegative matrix factorization and spectral masks," in *Int. Conf. on Digital Signal Process.*, 2011, pp. 1–6.

[14] G. Hinton, N. Srivastava, and K. Swersky, "Lecture 6: Overview of mini-batch gradient descent," in *Coursera Lecture slides.* [Online]. Available: https://class.coursera.org/neuralnets-2012-001/lecture

[15] J. W. Shin, J.-H. Chang, and N. S. Kim, "Voice activity detection based on statistical models and machine learning approaches," *Computer Speech Lang.*, vol. 24, no. 3, pp. 515–530, Jul. 2010.

[16] D. Enqing, L. Guizhong, and Z. Y. andZ. Xiaodi, "Applying support vector machines to voice activity detection," in *Proc. ICASSP*, 2002, pp. 1124–1127.

[17] J. Wu and X.-L. Zhang, "Efficient multiple kernel support vector machine based voice activity detection," *IEEE Signal Process. Lett.*, vol. 18, no. 8, pp. 466–469, Aug. 2011.

[18] D. Ying, Y. Yan, J. Dang, and F. K. Soong, "Voice activity detection basd on an unsupervised learning framework," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 8, pp. 2624–2633, Nov. 2011.

[19] X.-L. Zhang and J. Wu, "Deep belief network based voice activity detection," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 4, pp. 697–710, Apr. 2013.

[20] T. G. Kang and N. S. Kim, "DNN-based voice activity detection with multitask learning," *IEICE Trans. Inf. Syst.*, vol. E99-D, no. 2, pp. 550–553, Feb. 2016.

[21] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Process. Lett.*, vol. 6, no. 12, pp. 1–3, Jan. 1999.

[22] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[23] D. Pearce and H.-G. Hirsch, “The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions,” in *Proc. ISCA ITRW ASR*, Sept. 2000, pp. 29–32.

[24] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, “Theano: A CPU and GPU math complier in Python,” in *Proc. Scientific Comput. with Python Conf. (SciPy)*, Jul. 2010, pp. 3–9.

[25] C. Wang, L. Lan, Y. Zhang, and M. Gu, “Face recognition based on principle component analysis and support vector machine,” in *Proc. Int. Workshop on Intelligent Syst. and Applicat.*, 2011, pp. 1–4.

[26] P. O. Hoyer and A. Hyvärinen, “Independent component analysis applied to feature extraction from colour and stereo images,” *Network: Comput. Neural Syst.*, vol. 11, no. 3, pp. 191–210, Jan. 2000.

[27] R. Ramanath, W. E. Snyder, and H. Qi, “Eigenviews for object recognition in multispectral imaging system,” in *Proc. Applied Imagery Pattern Recognition (AIPR) Workshop.*, 2003, pp. 33–38.

[28] Y. G. Jin and N. S. Kim, “On detecting target acoustic signals based on nonnegative matrix factorization,” *IEICE Trans. on Inform. and Syst.*, vol. E93-D, no. 4, pp. 922–925, Apr. 2010.

[29] S. Choi, “Algorithms for orthogonal nonnegative matrix factorization,” in *Proc. IEEE Int. Joint Conf. Neural Networks*, 2008, pp. 1828–1832.

[30] I. Tosic and P. Frossard, “Dictionary learning,” *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 27–38, Mar. 2011.

[31] Z. Wang and F. Sha, "Discriminative non-negative matrix factorization for single-channel speech separation," in *Proc. ICASSP*, 2014, pp. 3777–3781.

[32] T. G. Kang, K. Kwon, J. W. Shin, and N. S. Kim, "NMF-based target source separation using deep neural network," *IEEE Signal Process. Lett.*, vol. 22, no. 2, pp. 229–233, Feb. 2015.

[33] A. Mohamed, G. E. Dahl, and G. E. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.

[34] R. Salakhutdinov and G. Hinton, "Using deep belief nets to learn covariance kernels for Gaussian processes," in *Proc. Advances in Neural Inform. Process. Syst.*, 2007, pp. 1–8.

[35] J. Du and Q. Huo, "A speech enhancement approach using piecewise linear approximation of an explicit model of environmental distortions," in *Proc. Interspeech*, 2008, pp. 569–572.

[36] P. S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "'deep learning for monaural speech separation," in *Proc. ICASSP*, 2014, pp. 1581–1585.

[37] E. M. Grais, M. U. Sen, , and H. Erdogan, "Deep neural networks for single channel source separation," in *Proc. ICASSP*, 2014, pp. 3762–3766.

[38] G. Cabras, S. Canazza, P. L. Montessoro, and R. Rinaldo, "Restoration of audio documents with low snr: a nmf parameter estimation and perceptually motivated bayesian suppression rule," in *Proc. Sound and Music Computing Conf.*, 2010, pp. 314–321.

[39] R. B. Palm, “Prediction as a candidate for learning deep hierarchical models of data,” in *Technical University of Denmark*, 2012. [Online]. Available: http://www2.imm.dtu.dk/pubdb/views/publication$_$details.php?id=628

[40] Y. Qian, Y. Fan, W. Hu, and F. K. Soong, “On the training aspects of deep neural network (DNN) for parametric synthesis,” in *Proc. ICASSP*, 2014, pp. 3857–3861.

[41] A. Varga and H. J. M. 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 Commun.*, vol. 12, no. 3, pp. 247–251, Jul. 1993.

[42] ITU, “Test signals for use in telephonometry,” ITU-T Rec. P. 501, 2012.

[43] E. Vincent, R. Gribonval, and C. Fevotte, “Performance measurement in blind audio source separation,” *IEEE Trans. Audio, Speech, and Language Process.*, vol. 14, no. 4, pp. 1462–1469, Jul. 2006.

[44] ITU, “Perceptual evalulation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs,” ITU-T Rec. P. 862, 2000.

[45] P. C. Loizou, *Speech Enhancement: Theory and Practice*, Boca Raton, FL, USA: CRC, 2013.

[46] F. Seide, G. Li, and D. Yu, “Conversational speech transcription using context-depdendent deep neural networks,” in *Proc. Interspeech*, 2011, pp. 437–440.

[47] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Process. Lett.*, vol. 21, no. 1, pp. 65–68, Jan. 2014.

[48] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 1, pp. 7–19, Jan. 2015.

[49] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smargadis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 12, pp. 2136–2147, Dec. 2015.

[50] K. Han, Y. Wang, D. L. Wang, W. S. Woods, I. Merks, and T. Zhang, "Learning spectral mapping for speech dereverberation and denoising," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 6, pp. 982–992, Jun. 2015.

[51] Y. Wang, A. Narayanan, and D. L. Wang, "On training targets for supervised speech separation," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 12, pp. 1849–1858, Dec. 2014.

[52] D. S. Williamson, Y. Wnag, and D. L. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 3, pp. 483–492, Mar. 2016.

[53] X.-L. Zhang and D. L. Wang, "A deep ensemble learning method for monaural speech separation," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. To be appeared.

[54] R. L. Goldsworthy and J. E. Greenberg, "Analysis of speech-based speech transmission index methods with implications for nonlinear operations," *J. Acoust. Soc. Amer.*, vol. 116, no. 6, pp. 3679–3689, 2004.

[55] J. M. Kates and K. H. Arehart, "Coherence and the speech intelligibility index," *J. Acoust. Soc. Amer.*, vol. 117, no. 4, pp. 2224–2237, 2005.

[56] H. S. Kim, Y. M. Cho, and H.-J. Kim, "Speech enhancement via mel-scale Wiener filtering with a frequency-wise voice activity detector," *J. Mech. Sci. Technology*, vol. 21, no. 5, pp. 708–722, 2007.

[57] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time-frequency weighted noisy speech," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 7, pp. 2125–2136, 2011.

[58] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. Interspeech*, 2013, pp. 2345–2349.

[59] A. Narayanan and D. L. Wang, "Improving robustness of deep neural network acoustic models via speech separation and joint adaptive training," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 1, pp. 92–101, Jan. 2015.

[60] Y. Bengio, "Practical recommendation for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*, G. Montavon, G. Orr, and K.-R. Müller, Eds. Springer-Verlag, 2012, pp. 437–478.

[61] T. F. Quatieri, *Discrete-Time Speech Signal Processing: Principles and Practice*, Pearson Education, 2013.

[62] Y. Dauphin, R. Pascanu, Ç. Gülçehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," *arXiv*, vol. abs/1406.2572, 2014.

[63] L. J. Ba and B. J. Frey, "Adaptive dropout for training deep neural networks," in *Advances in Neural Information Processing Systems*, 2013, pp. 3084–3092.

[64] L. Wan, M. D. Zeiler, S. Zhang, Y. LeCun, and R. Fergus, "Regularization of neural networks using dropconnect," in *Int. Conf. Mach. Learning,*, 2013, pp. 1058–1066.

[65] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," Université de Montréal, Tech. Rep. Arxiv report 1412.3555, 2014, presented at the Deep Learning workshop at NIPS2014.

[66] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, ser. Studies in Computational Intelligence. Springer, 2012, vol. 385.

[67] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "Global variance equalization for improving deep neural network based speech enhancement," in *ChinaSIP,*, 2014, pp. 71–71.

Appendix

In this Appendix we present the detail on deriving $\frac{\partial C}{\partial \mathbf{x}_t(f)}$. The gradient of the proposed objective function is given by the sum of the separate gradients of the three sub-costs as follows:

$$\frac{\partial C}{\partial \mathbf{x}_t(f)} = \frac{\partial C_{wmse}}{\partial \mathbf{x}_t(f)} - \lambda_1 \frac{\partial \rho_1}{\partial \mathbf{x}_t(f)} - \lambda_2 \frac{\partial \rho_2}{\partial \mathbf{x}_t(f)}. \tag{8.1}$$

Similar to the conventional mean square error, $\frac{\partial C_{wmse}}{\partial \mathbf{x}_t(f)}$ is given by

$$\frac{\partial C_{wmse}}{\partial \mathbf{x}_t(f)} = \frac{2}{T} w(f)(\mathbf{x}_t(f) - \mathbf{y}_t(f)). \tag{8.2}$$

The gradients of the second and third sub-costs are given by (8.3)–(8.5) and (8.6)–(8.8), respectively.

After $\{\frac{\partial C}{\partial \mathbf{x}_t(f)}\}$ are derived, the gradient of the proposed objective function with respect to each network parameter θ is obtained as

$$\frac{\partial C}{\partial \theta} = \sum_{t=1}^{T} \sum_{f=1}^{F} \frac{\partial C}{\partial \mathbf{x}_t(f)} \frac{\partial \mathbf{x}_t(f)}{\partial \theta} \tag{8.9}$$

and the usual back-propagation algorithm is applied.

$$\begin{aligned}\frac{\partial\rho_1}{\partial\mathbf{x}_t(f)} &= \frac{1}{(T-N+1)H}\sum_{\tau=1}^{T-N+1}\sum_{h=1}^{H}\frac{\partial\rho_1(\tau,h)}{\partial\mathbf{x}_t(f)} \\ &= \frac{1}{(T-N+1)H}\sum_{\tau=1}^{T-N+1}\sum_{h=1}^{H}\frac{\partial\rho_1(\tau,h)}{\partial\tilde{\mathbf{x}}_t(h)}\frac{\partial\tilde{\mathbf{x}}_t(h)}{\partial\mathbf{x}_t(f)} \qquad (8.3)\end{aligned}$$

$$\frac{\partial\rho_1(\tau,h)}{\partial\tilde{\mathbf{x}}_t(h)} = \begin{cases}\frac{\tilde{\mathbf{y}}_t(h)-\mu_{\tilde{\mathbf{Y}}_{\tau,h}}}{||\tilde{\mathbf{X}}_{\tau,h}-\mu_{\tilde{\mathbf{X}}_{\tau,h}}\mathbf{1}_N||\ ||\tilde{\mathbf{Y}}_{\tau,h}-\mu_{\tilde{\mathbf{Y}}_{\tau,h}}\mathbf{1}_N||} - \rho_1(\tau,h)\frac{\tilde{\mathbf{x}}_t(h)-\mu_{\tilde{\mathbf{X}}_{\tau,h}}}{||\tilde{\mathbf{X}}_{\tau,h}-\mu_{\tilde{\mathbf{X}}_{\tau,h}}\mathbf{1}_N||^2}, & \text{if } \tilde{\mathbf{x}}_t(h)\in\tilde{\mathbf{X}}_{\tau,h}, \\ 0, & \text{otherwise.}\end{cases} \qquad (8.4)$$

$$\frac{\partial\tilde{\mathbf{x}}_t(h)}{\partial\mathbf{x}_t(f)} = \begin{cases}\frac{1}{2\tilde{\mathbf{x}}_t(h)}\ \mathbf{s}(f)\ exp(\mathbf{x}_t(f)\mathbf{s}(f)+\mathbf{m}(f)), & B(h,t)=1, \\ 0, & B(h,t)=0.\end{cases} \qquad (8.5)$$

$$\frac{\partial\rho_2}{\partial\mathbf{x}_t(f)} = \frac{1}{T}\frac{\partial\rho_2(t)}{\partial\tilde{\mathbf{x}}'_t(f)}\frac{\partial\tilde{\mathbf{x}}'_t(f)}{\partial\mathbf{x}_t(f)}, \qquad (8.6)$$

$$\frac{\partial\rho_2(t)}{\partial\tilde{\mathbf{x}}'_t(f)} = \frac{\tilde{\mathbf{y}}'_t(f)-\mu_{\tilde{\mathbf{y}}'_t}}{||\tilde{\mathbf{x}}'_t-\mu_{\tilde{\mathbf{x}}'_t}\mathbf{1}_F||\ ||\tilde{\mathbf{y}}'_t-\mu_{\tilde{\mathbf{y}}'_t}\mathbf{1}_F||} - \rho_2(t)\frac{\tilde{\mathbf{x}}'_t(f)-\mu_{\tilde{\mathbf{x}}'_t}}{||\tilde{\mathbf{x}}'_t-\mu_{\tilde{\mathbf{x}}'_t}\mathbf{1}_F||^2}, \qquad (8.7)$$

$$\frac{\partial\tilde{\mathbf{x}}'_t(f)}{\partial\mathbf{x}_t(f)} = \frac{1}{2\tilde{\mathbf{x}}'_t(f)}\ \mathbf{s}(f)\ exp(\mathbf{x}_t(f)\mathbf{s}(f)+\mathbf{m}(f)). \qquad (8.8)$$

요 약

본 논문에서는 딥 러닝 기법을 적용한 강인한 음성구간 검출 및 음성 향상 알고리즘들을 제안한다. 강인한 음성구간검출 알고리즘에서는 잡음환경의 특성 차이로 인한 일부 환경에서의 DNN 기반 알고리즘의 성능 감소를 보상하기 위하여 기존의 VAD를 위한 DNN과 음성 피쳐 향상을 위한 DNN의 일부 은닉층을 연결하여 학습하는 MTL 기법을 적용하였다. 제안된 MTL 기법은 DNN이 단순히 입력과 출력 사이의 관계를 외우듯이 학습하지 않고 입력 피쳐에서 잡음의 영향을 제거하는 일반적인 잡음 환경에 적응 가능한 함수를 학습하도록 한다. 제안된 기법은 학습 DB에 포함되지 않은 잡음 환경에서 기존 기법보다 더 나은 성능을 보였다.

음성 향상 분야에서는 기존의 NMF 알고리즘의 한계점을 분석하고, 이러한 한계점을 보안하기 위해 제안되었던 DNMF 알고리즘에 대해서 언급한 뒤 NMF 기법에 필요한 encoding vector를 DNN을 이용해 추정하는 기법을 제안한다. 제안된 알고리즘은 음성과 잡음의 특성이 비슷하거나 음성 및 잡음을 묘사하는 basis vector 들간의 비선형적인 관계를 포착할 수 있다. 제안된 기법은 다양한 잡음 환경에서 기존 기법들과 성능이 비교되어 더 나은 성능을 보였다.

두 번째로는 제안된 DNN을 적용한 음성 향상 기법의 일반적인 프레임워크에 대해 사람이 주관적으로 느끼는 성능을 향상시키기 위해 주관적인 음성 품질에 큰 영향을 미치는 것으로 보고된 특징들을 DNN이 보상하도록 하는 목적함수를 제안

한다. 이를 통해 DNN은 사람이 보다 중요하게 생각하는 주파수 대역에서 정확한 음성을 추정하는 데 보다 집중하게 되며, 음성과 잡음 간의 오차를 일대일 관계가 아닌 전체 음성의 흐름 안에서 판단하게 되어 보다 자연스럽고 인지하기 쉬운 음성을 학습하도록 한다. 제안된 기법은 기존의 DNN 기반 기법과 비교되어 객관적, 주관적으로 보다 나은 성능을 보였다.

마지막으로 DNN을 이용해 추정한 음성 파라미터의 over-smoothing을 막기 위한 후처리 필터로 사용되는 GV 기반 피쳐를 대신하는 Spectral variance 후처리 필터를 제안하였다. 제안된 후처리 필터는 각각의 주파수 대역에서의 추정 값을 독립적으로 생각하지 않고 그들 간의 대조를 보다 뚜렷하게 하여 향상된 음성에서의 formant 구조를 더욱 두드러지게 한다. 제안된 후처리 기법은 앞에서 제안된 목적 함수에 대한 연구와 결합하여서 기존 기법에 비해 향상된 음성의 품질을 더욱 향상시킨다. 제안된 기법은 기존의 후처리 필터와 비교되어 주관적, 객관적 품질 평가에서 더 좋은 성능을 보였다.

주요어: 강인한 음성 구간 검출, 음성 향상, deep neural network (DNN), multi-task learning (MTL), nonnegative matrix factorization (NMF), 목적 함수, 후처리 필터

학 번: 2012-30189