2018 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 17–20, 2018, AALBORG, DENMARK

NOISE-ADAPTIVE DEEP NEURAL NETWORK FOR SINGLE-CHANNEL SPEECH ENHANCEMENT

Hanwook Chung[1]*, Taesup Kim*[2]*, Eric Plourde*[3] *and Benoit Champagne*[1]

[1] Dept. of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada
[2] MILA, Université de Montreal, Montreal, QC, Canada
[3] Dept. of Electrical and Computer Engineering, Sherbrooke University, Sherbrooke, QC, Canada
email: hanwook.chung@mail.mcgill.ca, taesup.kim@umontreal.ca, eric.plourde@usherbrooke.ca, benoit.champagne@mcgill.ca

ABSTRACT

We introduce a noise-adaptive feed-forward deep neural network (DNN) for single-channel speech enhancement. The goal is to better exploit individual noise characteristics while training a spectral mapping DNN. To this end, we employ noise-dependent adaptation vectors, which are obtained based on the output of an auxiliary noise classification DNN, to adjust the weights and biases of the spectral mapping DNN. The parameters of the spectral mapping DNN, noise classification DNN and adaptation vectors are estimated jointly during the training stage. During the enhancement stage, we combine a classical unsupervised speech enhancement algorithm with the proposed DNN-based approach to further improve the enhanced speech quality. Experiments show that the proposed method provides better enhancement performance than the selected benchmark algorithms.

Index Terms— Single-channel speech enhancement, deep neural network, classification

1. INTRODUCTION

The general objective of speech enhancement algorithms is to remove the background noise from a noisy speech signal to improve its quality and/or intelligibility. These algorithms find diverse applications including mobile telephony, hearing aids, speech coding and automatic speech recognition, to name a few. A considerable amount of research efforts have been made in the past decades, leading to various approaches, such as: minimum mean-square error (MMSE) estimation [1], spectral subtraction [2] and subspace decomposition method [3]. However, these classical methods were originally introduced by using a minimal amount of *a priori* information about the speech and noise. Consequently, they tend to provide limited enhancement performance, especially when the speech is corrupted by adverse noise, such as under low input signal-to-noise ratio (SNR) or non-stationary noise conditions.

To overcome these limitations, machine learning techniques have been applied to the speech enhancement task and have shown remarkable improvement in recent years. One of the widely considered technique is the non-negative matrix factorization (NMF) method, e.g., [4, 5], which decomposes a given matrix into basis and activation matrices with non-negative elements [6]. In a supervised framework, the basis vectors (also referred to as a codebook or dictionary) of the speech and noise sources are obtained *a priori* from the training data, and subsequently used during the enhancement stage.

In recent years, deep neural network (DNN) algorithms have gained enormous interest [7], and found diverse applications such as image classification [8], automatic speech recognition [9] and speech enhancement [10]. Supervised DNN training aims at estimating the nonlinear mapping function, specified by the weights and biases of the hidden layers of a processing network, that relates the input features to the target output features. Various DNN structures, such as deep autoencoder [11], feed-forward DNN [10] and convolutional neural network (CNN) [12] have been employed for speech enhancement. A recurrent neural network (RNN) has been employed in [13, 14], to better capture the temporal dynamics of audio signals. References [15, 16] consider a long-short term memory (LSTM) network, which is a modified version of the RNN proposed to lessen its effect of the long-term temporal dependency. A combination of the NMF and DNN frameworks has been introduced in [17, 18], while perceptually-motivated DNN algorithms have been introduced in [19, 20]. Reference [10] proposes a noise-aware training (NAT) framework, where the noise information is utilized during the DNN training by augmenting the estimated noise features with the DNN input features. However, the noise features are obtained by averaging first few frames of the given noisy speech spectrum and fixed over the utterance, which may limit the ability of the DNN in capturing the noise characteristics.

In this paper, we introduce a novel noise-adaptive feed-forward DNN for single-channel speech enhancement. The goal is to better exploit individual noise characteristics while training the spectral mapping DNN. To this end, motivated by [21], we adjust the weights and biases of the spectral mapping DNN via noise-dependent adaptation vectors. The latter are obtained based on the output of an auxiliary noise classification DNN. The parameters of the spectral mapping DNN, noise classification DNN and the adaptation vectors are estimated jointly during the training stage. During the enhancement stage, the clean speech signal is estimated from the spectral DNN, which is dynamically adapted to an arbitrary noise type via noise adaptation vectors without any additional fine-tuning step. Moreover, we combine a classical unsupervised speech enhancement algorithm with the DNN-based approach to further improve the enhanced speech quality. Experiments focusing on different performance metrics of interest show that the proposed method provides better enhancement performance than the selected benchmarks.

Funding for this work was provided by a CRD grant from NSERC (Govt. of Canada), with sponsorship from Microsemi Corporation (Ottawa, Canada).

978-1-5386-5477-4/18/$31.00

2. DNN-BASED SPEECH ENHANCEMENT

In single-channel speech enhancement problem, the noisy speech spectrum, obtained via short-time Fourier transform (STFT), can be expressed as

$$Y_{kl} = S_{kl} + N_{kl} \tag{1}$$

where Y_{kl}, S_{kl} and N_{kl} respectively denote the STFT coefficients of the noisy speech, clean speech and noise at the frequency bin $k \in \{1, ..., K\}$ and time frame $l \in \{1, ..., L\}$. The clean speech spectrum can be estimated via Wiener filtering (WF) [22], that is,

$$\hat{S}_{kl} = \frac{\hat{P}^S_{kl}}{\hat{P}^S_{kl} + \hat{P}^N_{kl}} Y_{kl} = \hat{G}_{kl} Y_{kl} \tag{2}$$

where $\hat{P}^S_{kl}$ and $\hat{P}^N_{kl}$ respectively denote the estimated power spectral densities (PSD) of the clean speech and noise. The latter are typically obtained via temporal smoothing of the periodograms of the clean speech and noise spectral estimates as

$$\hat{P}^S_{kl} = \tau_S \hat{P}^S_{k,l-1} + (1 - \tau_S)|\widetilde{S}_{kl}|^2 \tag{3}$$

$$\hat{P}^N_{kl} = \tau_N \hat{P}^N_{k,l-1} + (1 - \tau_N)|\widetilde{N}_{kl}|^2 \tag{4}$$

where τ_S and τ_N are the smoothing factors for the clean speech and noise. The estimates $|\widetilde{S}_{kl}|$ and $|\widetilde{N}_{kl}|$ can be obtained via a DNN approach (e.g., [23]), as explained below.

Supervised DNN training aims at estimating the nonlinear mapping function, specified by the weights and biases of the hidden layers of a processing network, that relates the input features to the target output features. Let $m \in \{1, ..., M\}$ denote the *hidden* layer index, where each layer consists of I_m neurons. For a feed-forward DNN, the output of the m-th hidden layer can be expressed in a vector form as

$$\mathbf{h}^{(m)} = f\big(\mathbf{W}^{(m)}\mathbf{h}^{(m-1)} + \mathbf{b}^{(m)}\big) \tag{5}$$

where $\mathbf{h}^{(m)} \in \mathbb{R}^{I_m}$ is the output of the m-th hidden layer, $\mathbf{W}^{(m)} \in \mathbb{R}^{I_m \times I_{m-1}}$ is the weight matrix, $\mathbf{b}^{(m)} \in \mathbb{R}^{I_m}$ is the bias vector, and $f(\cdot)$ represents an activation function that operates component-wise. Note that $\mathbf{h}^{(0)} \in \mathbb{R}^{I_0}$ and $\mathbf{h}^{(M+1)} \in \mathbb{R}^{I_{M+1}}$ respectively denote the DNN input and output values.

In DNN-based speech enhancement, various types of input and/or output features can be used, such as time-domain signal samples, mel-frequency cepstral coefficients (MFCC), magnitude or power spectral coefficients, ideal binary mask (IBM) or ideal ratio mask (IRM). In this work, we consider the log-power spectral coefficients (LPS) [10, 24]. That is, we employ the LPS of the noisy speech (i.e., $\mathbf{v}^Y_l \triangleq [\ln|Y_{kl}|^2] \in \mathbb{R}^K$) as the input and the corresponding LPS of the clean speech and noise (i.e., $\mathbf{v}^S_l \triangleq [\ln|S_{kl}|^2] \in \mathbb{R}^K$ and $\mathbf{v}^N_l \triangleq [\ln|N_{kl}|^2] \in \mathbb{R}^K$) as the target output features. Specifically, we formulate the DNN target output as $[(\mathbf{v}^S_l)^T\ (\mathbf{v}^N_l)^T]^T$ [13, 18, 24], where superscript T is transpose. The main DNN that maps the input noisy speech spectral features to the output clean speech and noise spectral features will be referred to as spectral mapping DNN.

For given training data sets $[\mathbf{v}^Y_l]$, $[\mathbf{v}^S_l]$ and $[\mathbf{v}^N_l]$ ($\in \mathbb{R}^{K \times L}$), the DNN parameters $\mathbf{W} = \{\mathbf{W}^{(m)}\}$ and $\mathbf{b} = \{\mathbf{b}^{(m)}\}$ are estimated by minimizing the mean-square error (MSE):

$$E = \frac{1}{KL}\sum_{l=1}^{L}\Big[\parallel \mathbf{v}^S_l - \widetilde{\mathbf{v}}^S_l \parallel^2_F + \parallel \mathbf{v}^N_l - \widetilde{\mathbf{v}}^N_l \parallel^2_F\Big] \tag{6}$$

where $\widetilde{\mathbf{v}}^S_l \triangleq [\ln|\widetilde{S}_{kl}|^2] \in \mathbb{R}^K$ and $\widetilde{\mathbf{v}}^N_l \triangleq [\ln|\widetilde{N}_{kl}|^2] \in \mathbb{R}^K$ are the DNN outputs and $\parallel \cdot \parallel_F$ is the Frobenius norm. During the enhancement stage, once we obtain the DNN outputs $\widetilde{\mathbf{v}}^S_l$ and $\widetilde{\mathbf{v}}^N_l$ from $\mathbf{v}^Y_l$, the PSDs are computed based on (3) and (4), and the clean speech spectrum is then estimated via (2). Finally, the enhanced speech signal in the time-domain is reconstructed by applying the inverse STFT to $\hat{S}_{kl}$, followed by the overlap-add method.

3. PROPOSED NOISE-ADAPTIVE DNN

The structure of the proposed noise-adaptive feed-forward DNN, which will be referred to as NA-DNN in the sequel, is illustrated in Fig. 1. It consists of a modified version of the spectral mapping DNN introduced Section 2, along with a noise classification DNN. The output of the latter is used while computing the noise-dependent weight adaptation and bias vectors, which in turn are used to adjust the parameters of the spectral mapping DNN.

3.1. Internal structure of NA-DNN

Let us denote by $j \in \{1, ..., J\}$ the noise class index, and by $m' \in \{1, ..., M'\}$ the hidden layer index of the noise classification DNN. The structure of the latter is similar to that of the spectral mapping DNN in Section 2. The output of the m'-th hidden layer is written as

$$\mathbf{h}^{(m')}_d = f\big(\mathbf{W}^{(m')}_d \mathbf{h}^{(m'-1)}_d + \mathbf{b}^{(m')}_d\big). \tag{7}$$

where $\mathbf{W}^{(m')}_d$ and $\mathbf{b}^{(m')}_d$ respectively denote the weight matrix and bias vector of the noise classification DNN. The input feature is $\mathbf{v}^Y_l$ and the target output is a class label vector $\mathbf{d} = [d_j] \in \mathbb{R}^J$ with $d_j \in \{0, 1\}$, such that $\sum_j d_j = 1$.

Motivated by [21], we modify the weight matrix and bias vectors of the spectral mapping DNN $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$, and replace the output of the m-th layer in (5) by the following hidden layer structure:

$$\mathbf{h}^{(m)} = f\big(\mathrm{diag}\{\mathbf{w}^{(m)}_a\}\mathbf{W}^{(m)}\mathbf{h}^{(m-1)} + \mathbf{b}^{(m)}_a\big) \tag{8}$$

where $\mathbf{w}^{(m)}_a \in \mathbb{R}^{I_m}$ and $\mathbf{b}^{(m)}_a \in \mathbb{R}^{I_m}$ are the noise-dependent weight adaptation vector and bias vector. The latter are given by

$$\mathbf{w}^{(m)}_a = \tanh\big(\mathbf{W}^{(m)}_w \widetilde{\mathbf{d}} + \mathbf{b}^{(m)}_w\big) \tag{9}$$

$$\mathbf{b}^{(m)}_a = \tanh\big(\mathbf{W}^{(m)}_b \widetilde{\mathbf{d}} + \mathbf{b}^{(m)}_b\big) \tag{10}$$

where $\tanh(\cdot)$ is the hyperbolic tangent function that operates component-wise, and $\widetilde{\mathbf{d}} = [\widetilde{d}_j] \in \mathbb{R}^J$ is the noise classification DNN output. Note that $\widetilde{d}_j$ represents the estimated likelihood of class j. The processes in (9) and (10) can be considered as mapping the input feature $\widetilde{\mathbf{d}}$ to $\mathbf{w}^{(m)}_a$ and $\mathbf{w}^{(m)}_b$, where the mapping functions are parameterized by $\mathbf{W}^{(m)}_w \in \mathbb{R}^{I_m \times J}$, $\mathbf{b}^{(m)}_w \in \mathbb{R}^{I_m}$, $\mathbf{W}^{(m)}_b \in \mathbb{R}^{I_m \times J}$ and $\mathbf{b}^{(m)}_b \in \mathbb{R}^{I_m}$.

In the proposed framework, we use rectified linear units (ReLU) as the activation function for all hidden layers m and m'. We use the linear and sigmoid activation functions for the output layers of the spectral mapping DNN and noise classification DNN[1], respectively.

[1] In a classification task, the softmax activation function is commonly used for the output layer, and the DNN parameters are estimated by minimizing the cross-entropy [25]. However, we empirically found that using the sigmoid function and considering the MSE-based cost function provided better enhancement performance in the proposed framework.

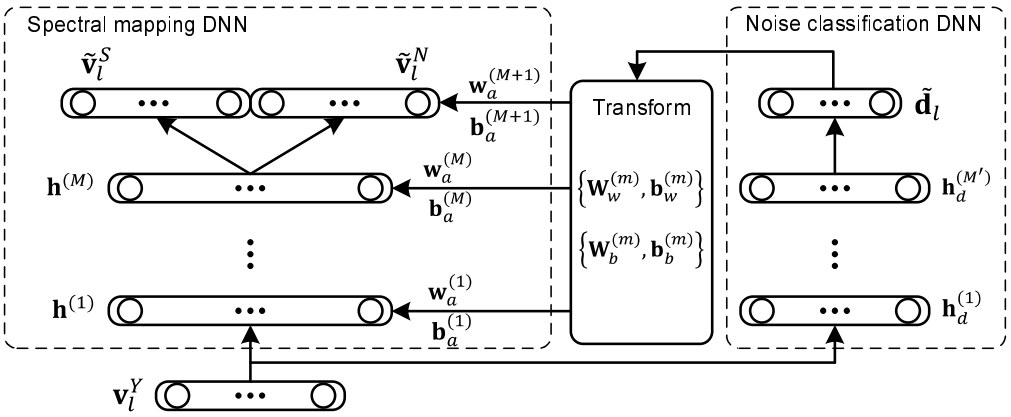

Fig. 1. The architecture of the proposed noise-adaptive feed-forward DNN.

3.2. Training of NA-DNN

The parameters of the proposed NA-DNN, $\mathbf{W} = \{\mathbf{W}^{(m)}, \mathbf{W}_w^{(m)}, \mathbf{W}_b^{(m)}, \mathbf{W}_d^{(m')}\}$ and $\mathbf{b} = \{\mathbf{b}_w^{(m)}, \mathbf{b}_b^{(m)}, \mathbf{b}_d^{(m')}\}$, are estimated jointly during the training stage. The proposed cost function is written as

$$
\begin{aligned}
E = & \frac{1}{KL}\sum_{l=1}^{L}\left[\|\mathbf{v}_l^S - \widetilde{\mathbf{v}}_l^S\|_F^2 + \|\mathbf{v}_l^N - \widetilde{\mathbf{v}}_l^N\|_F^2\right] + \frac{1}{JL}\sum_{l=1}^{L}\|\mathbf{d}_l - \widetilde{\mathbf{d}}_l\|_F^2 \\
& + \lambda_1 \sum_{m=1}^{M+1} \frac{1}{(J+1)^2} \parallel [\mathbf{b}_w^{(m)} \quad \mathbf{W}_w^{(m)}]^T [\mathbf{b}_w^{(m)} \quad \mathbf{W}_w^{(m)}] \parallel_F^2 \\
& + \lambda_2 \sum_{m=1}^{M+1} \frac{1}{(J+1)^2} \parallel [\mathbf{b}_b^{(m)} \quad \mathbf{W}_b^{(m)}]^T [\mathbf{b}_b^{(m)} \quad \mathbf{W}_b^{(m)}] \parallel_F^2 \\
& + \lambda_3 \sum_{m=1}^{M+1} \frac{1}{I_{m-1}} \parallel (\mathbf{w}_a^{(m)})^T \mathbf{W}^{(m)} \parallel_F^2
\end{aligned} \tag{11}
$$

where $\mathbf{d}_l = [d_{jl}] \in \mathbb{R}^J$ and $\widetilde{\mathbf{d}}_l = [\widetilde{d}_{jl}] \in \mathbb{R}^J$ are the target and output of the noise classification DNN for the input feature $\mathbf{v}_l^Y$. The first and second terms in the first line in (11) correspond to the MSE-based costs for the spectral mapping and noise classification DNN outputs, respectively. The last three lines in (11) represent regularization terms with regularization coefficients $\lambda_1, \lambda_2, \lambda_3 > 0$. The motivation of using the proposed regularization is explained as follows. As we can see in (9) and (10), the internal computations of the mapping functions can be interpreted as a linear combination of the columns of the weight matrices and the bias vectors (i.e., $[\mathbf{b}\ \mathbf{W}][1\ \widetilde{\mathbf{d}}^T]^T$). Each column of $\mathbf{W}_w^{(m)}$ and $\mathbf{W}_b^{(m)}$ corresponds to a specific noise type, and $\mathbf{b}_w^{(m)}$ and $\mathbf{b}_b^{(m)}$ can be considered as common vectors to the noise types. Based on these aspects, we add regularization terms to the MSE-based costs to enforce the columns to be distinct, to better exploit individual noise characteristics while computing $\mathbf{w}_a^{(m)}$ and $\mathbf{w}_b^{(m)}$. To this end, we consider the orthogonality between: i) the columns of $[\mathbf{b}_w^{(m)} \quad \mathbf{W}_w^{(m)}] \in \mathbb{R}^{I_m \times (J+1)}$, ii) the columns of $[\mathbf{b}_b^{(m)} \quad \mathbf{W}_b^{(m)}] \in \mathbb{R}^{I_m \times (J+1)}$, and iii) the weight-adaptation vector $\mathbf{w}_a^{(m)}$ and the columns of the common weight matrix $\mathbf{W}^{(m)}$. The DNN parameters are updated iteratively via error back-propagation with a stochastic gradient descent method [25].

3.3. Enhancement stage of NA-DNN

During the enhancement stage, we consider buffer processing [5]. That is, we aim at enhancing the noisy speech $\mathbf{Y}_{l_b} = [Y_{kl}] \in \mathbb{C}^{K \times L_b}$ obtained from consecutive time frames $l \in \{(l_b - 1)L_b + 1, ..., l_b L_b\} \triangleq \mathcal{C}_b$, where $l_b = 1, 2, ...$ is the buffer frame index and L_b is the buffer size. For a given l_b-th buffer frame, the spectral mapping DNN outputs are computed frame-by-frame. In contrast, the noise classification DNN output $\widetilde{\mathbf{d}}$ (and the resulting weight adaptation vector $\mathbf{w}_a$ and bias $\mathbf{b}_a$) is obtained based on the average noisy speech features, i.e., $(1/L_b)\sum_{l \in \mathcal{C}_b} \mathbf{v}_l^Y$. The purpose of such an implementation is to avoid rapid change of the estimated noise type that may deteriorate the enhanced speech quality. Once the spectral mapping DNN outputs are obtained, the clean speech spectrum can be estimated via Wiener filtering as explained in Section 2.

To further improve the enhanced speech quality, motivated by [5, 26], we combine a classical speech enhancement algorithm with the DNN-based framework. The basic idea is to use the classical method as a pre-processor to first remove some stationary background noise, and subsequently apply the DNN-based algorithm to further remove the remaining noise components. In this work, we we use the well-known MMSE short-time spectral amplitude (STSA) estimator [1] as the pre-processor, where the noise PSD is estimated based on [27].

Let us denote by $\bar{\mathbf{Y}}_{l_b} = [\bar{Y}_{kl}] \in \mathbb{C}^{K \times L_b}$ the pre-processed noisy speech for the l_b-th buffer frame. We estimate the clean speech spectrum where the magnitude components are obtained via the geometric mean (GM) of the magnitude spectra of: i) the signal obtained via WF (as explained in Section 2), ii) the pre-processed noisy speech signal, and iii) the Wiener-filtered pre-processed noisy speech signal. By taking the phase from the noisy speech signal, the proposed enhanced speech spectrum can be written as

$$
\hat{S}_{kl} = \left|\hat{G}_{kl} Y_{kl}\right|^{1/3} \left|\bar{Y}_{kl}\right|^{1/3} \left|\hat{G}_{kl} \bar{Y}_{kl}\right|^{1/3} e^{\jmath \angle Y_{kl}} \tag{12}
$$

where $\jmath = \sqrt{-1}$. The underlying motivation of using the GM is to compensate for the over-reduced clean speech components. That is, although Wiener filtering the pre-processed noisy speech can significantly remove the background noise components, the clean speech signal can be distorted as well. In contrast, the classical method tends to provide a limited performance on reducing the background noise especially for a low input SNR, but the enhanced speech signal contains most of the clean speech components.

4. EXPERIMENTS

In this section, after describing the data sets and general methodology, we present and discuss the experimental results.

4.1. Data sets

We conducted the experiments using the clean speech from the TIMIT corpus [28] and the noises from the NOISEX database [29], where the sampling rate of all signals was adjusted to 16 kHz. The speech and noise files were divided into three *disjoint* groups: i) *training data*, used for estimating the DNN parameters, ii) *validation data*, used for selecting tuning parameters such as regularization coefficients, and iii) *test data*, used during the enhancement stage to evaluate the enhancement performance. Regarding the clean speech training data, we selected 128 speakers with 2 utterances from each speaker from the TIMIT training set. For the noise training data, we selected Babble, Factory1, HF-Channel, Buccaneer1, Destroyerops, Leopard noises, where each noise type consisted of 3 minute long signal. The noisy speech training data were artificially generated by adding the noise training data to the clean speech training data to obtain input SNRs of -10, -5, 0, 5 and 10 dB, assuming that a single noise type is included in the noisy speech signal.

Regarding the test data, we selected 100 utterances from the TIMIT test set for the reference clean speech. The noise test data was catagorized into two groups, referred to as: *matched* and *unmatched* cases. The matched case assumes that the noise type is also presented in the training data, whereas the purpose of the unmatched case is to evaluate the performance for an unseen noise type. To this end, we additionally selected Destroyerengine, Pink and F16 noises from the NOISEX database for the unmatched case. For both the matched and unmatched cases, we considered two scenarios: *single-noise* and *multiple-noise* conditions. The former scenario assumes that a single noise type is included in the noisy speech signal. To examine the latter scenario, we generated a test noise signal for each matched and unmatched case, by summing all noise signals which were adjusted to have same variances in the time domain. For all cases, the noisy speech test signals were generated by adding the noise test signal to the reference clean speech test signal to obtain input SNRs of -10, -5, 0, 5, and 10 dB. The single-noise condition was considered while generating the noisy speech training data.

Regarding the validation data, we selected 50 utterances from the TIMIT test set. We considered the matched noise case for selecting noise validation data, and single-noise condition while generating the noisy speech validation signals.

4.2. Methodolody

Regarding the implementation, a Hanning window of 512 samples with 50% overlap was employed for the STFT analysis. The spectral mapping DNN was consisted of 3 hidden layers with 512 neurons for each layer, and the noise classification DNN was consisted of 2 hidden layers with 128 neurons for each layer. The weight matrix for each hidden layer were initialized by generating Gaussian random numbers with zero mean and standard deviation of 0.1 divided by the number of the neurons, and all biases were initialized to zero. Regarding the stochastic gradient descent method, we used the mini-batch size of 128 and adaptive moment estimation (Adam) optimizer [30] for 200 epochs. The initial learning rate was set to 0.001, which decreased by 10% for every 10 epochs. We adopted dropout training while estimating the DNN parameters to avoid overfitting [31], where the rate was set to 0.8. The DNN input features $\mathbf{v}_l^Y$ were normalized to have zero mean and unit variance across the time frames l. We used $\tau_S = 0.4$ and $\tau_N = 0.9$ while computing the smoothed PSDs in (3) and (4). The regularization coefficients were set to $\lambda_1 = \lambda_2 = \lambda_3 = 0.001$, and the buffer size L_b was set to 8. A smoothing factor of 0.85 was used in the decision-directed method to estimate the *a priori* SNR in the MMSE-STSA estimator [1], and a factor of 0.9 was used in the noise PSD estimation [27].

To evaluate the performance of the proposed method, we implemented several benchmark algorithms: i) MMSE-STSA [1] with noise PSD estimated based on [27], ii) Bayesian NMF (BNMF) [32], iii) feed-forward DNN introduced in Section 2, and iv) dynamic NAT (dNAT) [33]. For the BNMF, we trained 120 speaker-independent basis vectors as well as 120 noise-independent basis vectors. The implementation of the BNMF algorithm for single-channel speech enhancement can be found in [5]. The dNAT method, which was proposed to overcome the limitation of the static noise feature-based NAT method, considers the frame-wise estimated noise features. Among several realizations of dNAT in [33], we implemented the one referred to as dNAT1, which utilizes the noise features obtained via [27]. Basic settings such as the STFT analysis and synthesis, buffer size and DNN structure were kept identical when applicable.

We considered the perceptual evaluation of speech quality (PESQ) [34], source-to-distortion ratio (SDR) [35] and extended short-time objective intelligibility (ESTOI) [36] as the objective measures of the enhancement performance. For all measures, a higher value indicates a better result.

4.3. Results

Tables 1 and 2 show the average results over all utterances and all noises for the matched and unmatched cases regarding the single-noise condition. Tables 3 and 4 show the average results over all utterances for the matched and unmatched cases regarding the multiple-noise condition. The values in bold indicate the best performance along the corresponding row. The objective results of the BNMF, DNN and dNAT methods were computed using the WF-based reconstruction method explained in Section 2. For all cases, it can be observed that the proposed NA-DNN with the GM-based reconstruction method exhibited the best performance. Comparing between the DNN, dNAT and NA-DNN-WF methods, we can see that the proposed NA-DNN-WF method gave better results than the DNN and dNAT methods not only for the single-noise condition but also the multiple-noise condition, in general. Comparing between the WF and GM-based reconstruction methods (i.e., NA-DNN-WF versus NA-DNN-GM), the latter method provided better performance, which validates that employing the classical unsupervised method can further improve the enhanced speech quality, especially for the unmatched noise cases.

In the following, we comment on some additional experimental results, which we did not report in this paper. Regarding the single-noise condition, we found that the proposed NA-DNN-WF method provided better results than the DNN and dNAT methods for each noise type, in general. In addition, we examined the BNMF, DNN and dNAT methods using the proposed GM-based clean speech estimation given by (12), and we found that the GM-based method showed better results than the WF-based method. Comparing between the BNMF-GM, DNN-GM, dNAT-GM and NA-DNN-GM methods, we observed that the proposed NA-DNN-GM method exhibited the best performance, following similar patterns to those of the results based on WF shown in Tables 1 to 4.

5. CONCLUSION AND FUTURE WORKS

We introduced a noise-adaptive feed-forward DNN for single-channel speech enhancement. The goal was to better exploit in-

Table 1. Average results for the matched single-noise condition

Input SNR	Eval.	Noisy	STSA	BNMF	DNN	dNAT	NA-DNN WF	NA-DNN GM
-10 dB	PESQ	1.06	1.22	1.51	1.71	1.69	1.75	**1.81**
	SDR	-9.56	-6.49	-0.66	3.12	2.96	3.60	**3.69**
	ESTOI	0.21	0.23	0.24	0.36	0.36	**0.37**	**0.37**
-5 dB	PESQ	1.35	1.55	1.85	2.05	2.03	2.09	**2.17**
	SDR	-4.83	-1.19	3.91	6.02	6.04	6.55	**6.75**
	ESTOI	0.31	0.34	0.36	0.46	0.46	0.48	**0.49**
0 dB	PESQ	1.66	1.91	2.14	2.37	2.37	2.43	**2.49**
	SDR	0.08	3.87	7.54	8.79	9.01	9.46	**9.70**
	ESTOI	0.43	0.46	0.47	0.57	0.57	**0.60**	**0.60**
5 dB	PESQ	2.01	2.27	2.42	2.68	2.68	2.73	**2.77**
	SDR	5.05	8.66	10.64	11.57	12.05	12.40	**12.67**
	ESTOI	0.56	0.60	0.59	0.67	0.68	**0.70**	**0.70**
10 dB	PESQ	2.36	2.62	2.71	2.95	2.97	3.00	**3.03**
	SDR	10.04	13.21	13.22	14.30	15.10	15.34	**15.68**
	ESTOI	0.69	0.73	0.69	0.75	0.77	0.78	**0.79**

Table 2. Average results for the unmatched single-noise condition

Input SNR	Eval.	Noisy	STSA	BNMF	DNN	dNAT	NA-DNN WF	NA-DNN GM
-10 dB	PESQ	1.04	1.16	1.39	1.48	1.51	1.55	**1.68**
	SDR	-9.59	-5.98	-3.53	1.65	1.73	1.99	**2.60**
	ESTOI	0.16	0.20	0.19	0.30	0.31	0.32	**0.33**
-5 dB	PESQ	1.28	1.51	1.69	1.78	1.82	1.86	**2.01**
	SDR	-4.84	-0.64	1.49	4.45	4.65	4.94	**5.61**
	ESTOI	0.27	0.32	0.31	0.40	0.41	**0.43**	**0.43**
0 dB	PESQ	1.58	1.90	1.97	2.11	2.14	2.18	**2.32**
	SDR	0.07	4.37	5.63	7.17	7.50	7.87	**8.58**
	ESTOI	0.41	0.46	0.44	0.51	0.53	0.54	**0.55**
5 dB	PESQ	1.93	2.27	2.25	2.42	2.47	2.50	**2.62**
	SDR	5.04	9.07	9.32	10.02	10.53	10.91	**11.65**
	ESTOI	0.56	0.62	0.57	0.63	0.64	0.66	**0.67**
10 dB	PESQ	2.29	2.63	2.56	2.71	2.77	2.78	**2.89**
	SDR	10.04	13.55	12.52	12.93	13.76	13.99	**14.79**
	ESTOI	0.71	0.76	0.69	0.72	0.74	0.75	**0.78**

Table 3. Average results for the matched multiple-noise condition

Input SNR	Eval.	Noisy	STSA	BNMF	DNN	dNAT	NA-DNN WF	NA-DNN GM
-10 dB	PESQ	0.87	1.02	1.29	1.37	1.38	1.37	**1.49**
	SDR	-9.55	-6.59	-2.32	-0.98	-0.36	-0.50	**0.35**
	ESTOI	0.13	0.15	0.17	0.24	0.25	**0.26**	**0.26**
-5 dB	PESQ	1.12	1.37	1.60	1.68	1.70	1.69	**1.86**
	SDR	-4.83	-1.20	2.56	3.47	3.80	3.98	**4.57**
	ESTOI	0.23	0.26	0.28	0.35	0.36	**0.37**	**0.37**
0 dB	PESQ	1.44	1.77	1.90	2.01	2.03	2.02	**2.20**
	SDR	0.07	3.87	6.39	6.61	6.97	7.16	**7.72**
	ESTOI	0.36	0.40	0.41	0.47	0.48	**0.50**	**0.50**
5 dB	PESQ	1.82	2.16	2.20	2.34	2.37	2.39	**2.53**
	SDR	5.04	8.59	9.81	9.57	10.12	10.28	**10.83**
	ESTOI	0.50	0.55	0.54	0.59	0.60	0.62	**0.63**
10 dB	PESQ	2.20	2.53	2.52	2.66	2.71	2.72	**2.83**
	SDR	10.03	13.03	12.80	12.58	13.39	13.42	**13.99**
	ESTOI	0.65	0.70	0.66	0.70	0.72	0.73	**0.74**

Table 4. Average results for the unmatched multiple-noise condition

Input SNR	Eval.	Noisy	STSA	BNMF	DNN	dNAT	NA-DNN WF	NA-DNN GM
-10 dB	PESQ	1.01	1.13	1.40	1.45	1.48	1.52	**1.65**
	SDR	-9.60	-6.05	-4.81	1.93	2.05	2.39	**2.89**
	ESTOI	0.15	0.18	0.17	0.29	**0.30**	**0.30**	**0.30**
-5 dB	PESQ	1.24	1.47	1.64	1.76	1.76	1.80	**1.98**
	SDR	-4.85	-0.69	0.75	4.53	4.71	5.06	**5.63**
	ESTOI	0.25	0.30	0.29	0.39	0.40	**0.41**	**0.41**
0 dB	PESQ	1.53	1.85	1.88	2.07	2.08	2.15	**2.30**
	SDR	0.06	4.32	5.30	7.13	7.41	7.93	**8.51**
	ESTOI	0.39	0.44	0.41	0.50	0.51	0.53	**0.54**
5 dB	PESQ	1.87	2.23	2.16	2.38	2.41	2.48	**2.61**
	SDR	5.03	9.00	9.20	9.86	10.37	10.90	**11.50**
	ESTOI	0.54	0.60	0.55	0.61	0.63	0.65	**0.66**
10 dB	PESQ	2.23	2.59	2.49	2.68	2.73	2.77	**2.88**
	SDR	10.02	13.43	12.53	12.67	13.49	13.85	**14.52**
	ESTOI	0.69	0.74	0.68	0.72	0.73	0.75	**0.76**

dividual noise characteristics while training the spectral mapping DNN. To this end, we employed noise-dependent adaptation vectors, obtained based on the output of an auxiliary noise classification DNN, to adjust the weights and biases of the spectral mapping DNN. The parameters of the spectral mapping DNN, noise classification DNN and the adaptation vectors were estimated jointly during the training stage. In addition to the WF-based clean speech reconstruction during the enhancement stage, we introduced a method that combines a classical speech enhancement algorithm and the DNN-based approach to further improve the enhanced speech quality. Experiments showed that using the proposed NA-DNN structure provided better enhancement performance than the selected benchmark algorithms. Specifically, the GM-based reconstruction method exhibited further improvements of the performance, especially for the unseen noise types.

Finally, we comment on some interesting research avenue for our future works. First, we will extend the proposed NA-DNN to a LSTM-RNN to better capture the temporal dynamics. Second, we will additionally employ a speaker-dependent feature (e.g., [37]) to adjust the weights and biases of the spectral mapping DNN.

6. REFERENCES

[1] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Trans. Acoustics, Speech, and Signal Process.*, vol. 32 ,no. 6, pp. 1109-1121, Dec. 1984.

[2] N. Virag, "Single channel speech enhancement based on masking properties of the human auditory system," *IEEE Trans. Speech and Audio Process.*, vol. 7. no. 2, pp. 126-137, Mar. 1999.

[3] F. Jabloun and B. Champagne, "Incorporating human hearing properties in the signal subspace approach for speech enhancement," *IEEE Trans. Speech and Audio Process.*, vol. 11, no. 6, pp. 700-708, Nov. 2003.

[4] N. Mohammadiha, P. Smaragdis and A. Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 21, no. 10, pp. 2140-2151, Oct. 2013.

[5] H. Chung. R. Badeau, E. Plourde and B. Champagne, "Training and compensation of class-conditioned NMF bases for speech enhancement," *Neurocomputing*, vol. 284, pp. 107-118, Apr. 2018.

[6] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Advances in Neural Information Process. Systems*, pp. 556-562, 2001.

[7] Y. Bengio, A. Courville and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 35, no 8, pp. 1798-1828, Aug. 2013.

[8] D. Ciregan, U Meier and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. Computer*

Vision and Pattern Recognition (CVPR), pp 3642-3649, June 2012.

[9] L. Deng, G. Hinton and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: an overview," in *Proc. ICASSP*, pp. 8599-8603, May 2013.

[10] 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, and Language Process.*, vol. 23, no. 1, pp. 7-19, Jan. 2015.

[11] X. Lu, Y. Tsao, S. Matsuda and C. Hori, "Speech enhancement based on deep denoising autoencoder," in *Proc. Interspeech*, pp. 436-440, Aug. 2013.

[12] S. -W. Fu, Y. Tsao and X. Lu, "SNR-aware convolutional neural network modeling for speech enhancement," in *Proc. Interspeech*, pp. 3768-3772, Sep. 2016.

[13] P. -S. Huang, M. Kim, M. Hasegawa-Johnson and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Trans. Audio, Speech, and Language Process.*, vol. 23, no. 12, pp. 2136-217, Dec. 2015.

[14] G. -X. Wang, C. -C. Hsu and J. -T. Chien, "Discriminative deep recurrent neural networks for monaural speech separation," in *Proc. ICASSP*, pp. 2544-2548, Mar. 2016.

[15] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. L. Roux, J. R. Hershey and B. Schuller, "Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR," in *Proc. Int. Conf. on Latent Variable Analysis and Signal Separation*, pp. 91-99, Aug. 2015.

[16] M. Kolbaek, D. Yu, Z. -H. Tan and J. Jensen, "Joint separation and denoising of noisy multi-talker speech using recurrent neural networks and permutation invariant training," in *Proc. MLSP*, six pages, Sep. 2017.

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

[18] S. Nie, S. Liang, H. Li, X. Zhang, Z. Zhang, W. J. Liu and L. K. Dong, "Exploiting spectro-temporal structures using NMF for DNN-based supervised speech separation," in *Proc. ICASSP*, pp. 469-473, Mar. 2016.

[19] A. Kumar and D. Florencio, "Speech enhancement in multiple noise conditions using deep neural networks," in *Proc. Interspeech*, pp. 7-19, Sep. 2016.

[20] W. Han, X. Zhang, M. Sun, W. Shi, X. Chen and Y. Hu, "Perceptual improvement of deep neural networks for monaural speech enhancement," in *Proc. Int. Workshop on Acoustic Signal Enhancement*, five pages, Sep. 2016.

[21] T. Kim, I. Song and Y. Bengio, "Dynamic layer normalization for adaptive neural acoustic modeling in speech recognition," in *Proc. Interspeech*, pp. 2411-2415, Aug. 2017.

[22] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proc. IEEE*, vol. 67, no. 12, pp. 1586-1604, Dec. 1979.

[23] R. Razani, H. Chung, Y. Attabi and B. Champagne, "A reduced complexity MFCC-based deep neural network approach for speech enhancement," in *Proc. IEEE Symposium on Signal Process. and Information Tech.*, six pages, Dec. 2017.

[24] T. Gao, J. Du, L. -R. Dai and C. -H. Lee, "A unified DNN approach to speaker-dependent simultaneous speech enhancement and speech separation in low SNR environment," *Speech Communication*, vol. 95, pp. 28-39, Dec. 2017.

[25] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016.

[26] M. Sun, Y. Li, J. F. Gemmeke and X. Zhang, "Speech enhancement under low SNR conditions via noise estimation using sparse and low-rank NMF with Kullback-Leibler divergence," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 23, no. 7, pp. 1233-1242, July 2015.

[27] T. Gerkmann and R. C. Hendriks, "Unbiased MMSE-based noise power estimation with low complexity and low tracking delay," *IEEE Trans. Audio. Speech, and Language Process.*, vol. 20, no. 4, pp. 1383-1393, May 2012.

[28] J. S. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren and V. Zue, *TIMIT: acoustic-phonetic continuous speech corpus*, Linguistic Data Consortium, 1993.

[29] A. Varga and H. J. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: a database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication*, vol. 12, no. 3, pp. 247-251, July 1993.

[30] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," *arXiv: 1412.6980*, Dec. 2014.

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958, Jan. 2014.

[32] A. T. Cemgil, "Bayesian inference for nonnegative matrix factorization models," *Computational Intelligence and Neuroscience*, no. 4 Article ID 785152, pp. 1-17, 2009.

[33] Y. Xu, J. Du, L. -R. Dai and C. -H. Lee, "Dynamic noise aware training for speech enhancement based on deep neural networks," in *Proc. Interspeech*, pp. 2670-2674, Sep. 2014.

[34] ITU-T, *Recommendation P.862, Perceptual evaluation of speech quality (PESQ): and objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs*, Tech. Rep., 2001.

[35] E. Vincenet, 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, July 2006.

[36] J. Jensen and C. H. Taal, "An algorithm for predicting the intelligibility of speech masked by modulated noise maskers," *IEEE/ACM Trans. Audio, Speech, and Language Process.*, vol. 24, no. 11, pp. 2009-2022, Nov. 2016

[37] T. Tan, Y. Qian, D. Yu, S. Kundu, L. Lu, K. C. Sim, X. Xiao and Y. Zhang, "Speaker-aware training of LSTM-RNNs for acoustic modeling," in *Proc. ICASSP*, pp. 5280-5284, Mar. 2016.