



Robust Speech Recognition using DNN-HMM Acoustic Model Combining Noise-aware training with Spectral Subtraction

Akihiro Abe, Kazumasa Yamamoto, Seiichi Nakagawa

Department of Computer Science and Engineering, Toyohashi University of Technology, Japan

{abe, kyama, nakagawa}@slp.cs.tut.ac.jp

Abstract

Recently, acoustic models based on deep neural networks (DNNs) have been introduced and showed dramatic improvements over acoustic models based on GMM in a variety of tasks. In this paper, we considered the improvement of noise robustness of DNN. Inspired by Missing Feature Theory and static noise aware training, we proposed an approach that uses a noise-suppressed acoustic feature and estimated noise information as input of DNN. We used simple Spectral Subtraction as noise-suppression. As noise estimation, we used estimation per utterance or frame. In noisy speech recognition experiments, we compared the proposed method with other methods and the proposed method showed the superior performance than the other approaches. For noise estimation per utterance with log Mel Filterbank, we obtained 28.6% word error rate reduction compared with multi condition training, 5.9% reduction compared with noise adaptive training.

Index Terms: speech recognition, deep neural network, noise robustness, adaptive training, noise aware training, Spectral Subtraction, Missing Feature Theory

1. Introduction

In recent years, studies of speech recognition technology using the DNN-HMM acoustic models have been actively carried out [1], and they have shown that DNN-based acoustic models outperformed traditional GMM-HMM acoustic models. Improving noise robustness in the DNN-HMM acoustic model has also been studied [2]. To improve the robustness of conventional GMM-HMM acoustic models, the following methods have commonly been used: (1) performing noise suppression with the spectral domain or feature domain on the input speech by using, for example, Spectral Subtraction [3], (2) using robust acoustic features such as PNCC (Power-Normalized Cepstral Coefficients) [4], and (3) performing model adaptation such as MLLR (Maximum Likelihood Linear Regression), on the acoustic models. Various studies have been performed using each method [5]. Also, the Missing Feature Theory (MFT) has been proposed to improve robustness [6]. The MFT masks the parts of the acoustic features that were noise-corrupted to avoid the effect of noise.

For DNN-HMM acoustic models, as with GMM-HMM acoustic models, such methods for performing noise suppression using robust acoustic features are used. In particular, noise suppression with a neural network called the Denoising Autoencoder [7], which maps noisy features/spectra to clean them, has often been used. Although there have been several studies of noise suppression approaches with DNN, there have been few of noise adaptive approaches.

Under these circumstances, Seltzer *et al.* proposed a tech-

nique called *noise-aware training* [8], which uses noisy acoustic features and estimated noise information as DNN input. In the log-spectral domain, the relationship between noisy speech, clean speech, and noise is non-linear. The DNN consists of multiple layers of non-linear processing. Thus, by inputting noise information as additional information into the DNN, noise-aware training allows the DNN to learn the non-linear relationship between noisy speech and estimated noise information.

In this study, inspired by static noise-aware training and Missing Feature Theory, we proposed a method using noise-suppressed acoustic features and estimated noise information as the input of the DNN-HMM acoustic model. We intended that input noise information denotes the reliability of acoustic features. While static noise-aware training enables the learnings of various relationships between noisy speech and noise in the DNN, our proposed approach processes the linear relationship outside the DNN, and other relationships are learned inside the DNN to improve the DNN robustness. To evaluate this approach, we experimented with noisy speech recognition and showed that the proposed approach outperforms other approaches including static noise-aware training.

2. Related previous works

2.1. Noise-aware training

Static noise-aware training proposed by Seltzer [8] is a model learning approach for the DNN acoustic model that include noise adaptation. Figure 1 shows the architecture of static noise-aware training. As the input of the DNN, the static noise-aware training uses not only features extracted from noisy speech but also noise information estimated from the same noisy speech. The update interval of noise estimation was only for the beginning of the utterance, and noise information is input into only the input layer of DNN [8]. By using this approach, the robustness of the DNN-HMM acoustic model improved more than when using noise adaptive training [5]. In [9], noise information is input into not only the input layer, which is less reliable as a feature extractor, but also the output layer, which is more reliable as a feature extractor.

In [10], the same approach has been applied to speaker adaptation using “speaker code” instead of noise information, and the additional information has been input into not only the input layer but also the hidden and output layers.

Also, dynamic noise-aware training has been proposed in [16] for speech enhancement that performs noise estimation in the DNN as well.

2.2. Missing Feature Theory

Missing Feature Theory (MFT) is an approach to improve noise robustness [6]. This approach detects the component whose

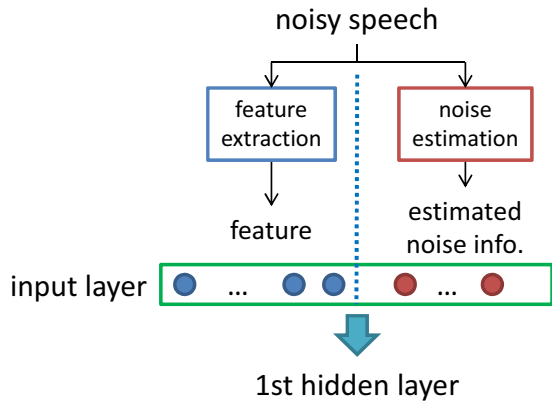


Figure 1: Architecture of noise aware training

quality is reduced by noise from an input speech as the Missing Feature. To avoid the bad effect of the Missing Feature, MFT generates a soft mask or hard mask and applies the mask to the acoustic features in speech recognition. Hence, it is possible to remove the effect of unreliable components on performance in the acoustic features. Moreover, the mask is computed in each channel and each frame, so MFT can deal with non-stationary noises. To compute the mask, both clean speech and noisy speech are required [18]. For example, when making a binary mask, if the absolute difference between the component of the feature from clean speech and the corresponding component of the features from noisy speech exceeds a threshold, this component will be masked.

3. Proposed approach

In this study, as the input of the DNN-HMM acoustics model, we propose using not only noise-suppressed acoustic features but also estimated noise information to make the feature reliable for the DNN. We aimed to mask the component corrupted by noise in acoustic features in the network.

Figure 2 shows the architecture of the proposed approach. Firstly, we estimated noise information. Next, we extracted noise-suppressed acoustic feature from the noisy speech using the estimated noise information. When inputting it into the DNN acoustic model, we used both the acoustic features and the estimated noise information.

We aim to improve the noise robustness of the DNN by inputting noise-suppressed features. Moreover, by inputting the estimated noise information that was used for noise-suppression as additional information, we consider that noise information works as a mask of “*implicit Missing Feature Theory*” within the DNN, which masks the noisy speech features with high noise information and enhances the features with low noise. The “implicit” means we do not performed masking to noise-suppressed acoustic features, and this task has been left to the DNN. In other words, noise information helps to adapt the DNN to noise. Thus, the DNN requires more complex architecture to compare the common DNNs without noise information, such as more hidden units or more hidden layers to perform noise adaptation.

3.1. Noise estimation and suppression

As noise suppression, we used simple Spectral Subtraction [3].

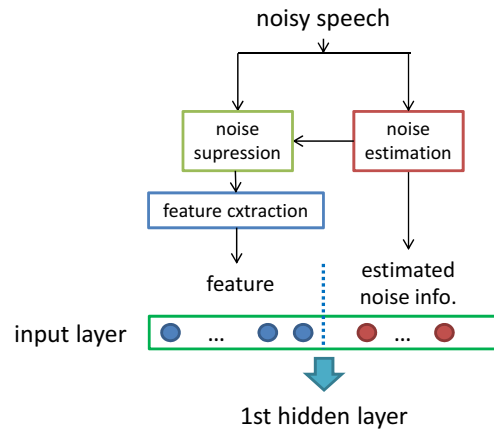


Figure 2: Architecture of proposed approach

$$S(k, m) = \begin{cases} X(k, m) - \alpha N(k, m) & \text{if } X(k, m) - \alpha N(k, m) \geq 0.0 \\ \beta X(k, m) & \text{otherwise,} \end{cases} \quad (1)$$

where $S(k, m)$ is noise-suppressed m -th frame, k -th filterbank channel output, $X(k, m)$ is filterbank channel output from noisy speech, $N(k, m)$ is estimated noise spectrum from $X(k, m)$, α is the subtraction factor, and β is the flooring factor.

3.1.1. Noise estimation using top frames

To estimate the noise per utterance, we used Eq.(2) from traditional Spectral Subtraction(SS) [3].

$$N(k) = \frac{1}{M} \sum_{m=1}^M X(k, m), \quad (2)$$

where M is the number of noise frames used for noise estimation.

3.1.2. Noise estimation using Minimum statistics

Deng *et al.* [13] proposed an approach to track the background noise in speech based on the Minimum Statistics (MS). The MS are calculated in the spectral domain, and the spectra are smoothed in both the time and frequency direction. However, we calculated the MS in the filterbank domain in this paper. Hence, the filterbank outputs have already been smoothed for the frequency direction. We performed smoothing only for the time direction and calculating the MS using Eq.(3).

$$N(k, m) = \begin{cases} \gamma N(k, m-1) + \frac{1-\gamma}{1-\beta_{MS}} (P(k, m) - \beta P(k, m-1)) & \text{if } N(k, m-1) < P(k, m) \\ P(k, m) & \text{otherwise,} \end{cases} \quad (3)$$

$$P(k, m) = \alpha_s P(k, m-1) + (1 - \alpha_s) X(k, m), \quad (4)$$

where γ and β_{MS} are the control parameters, $P(k, m)$ is time-smoothed $X(k, m)$, and α_s is the time smoothing parameter.

In [8], the noise estimation has been performed per utterance only; therefore, the estimated noise information has been

input into the DNN one frame at a time. However, we can consider the noise estimation per frame. Hence, when inputting the noise information into the DNN, we could input it several number of frames at a time along with the features.

4. Experiments and Results

4.1. Experimental setup

In this paper, we performed noisy speech recognition experiments on the CENSREC-1 (a.k.a. AURORA-2J, continuous Japanese digits) database [14]. We used multi-condition training data during training acoustic models and set B during the evaluation. The training set consisted of 8440 utterances (~ 5 hours) from 110 speakers and included clean speech and noisy speech with one of four different noises added (subway, babble, car, and exhibition) at different SNRs (20, 15, 10, and 5). The evaluation set included clean speech and noisy speech with one of four different noises added (restaurant, street, airport, and station) at six different SNRs (20, 15, 10, 5, 0, and -5) and consisted of 1001 utterances from 52 speakers for each condition.

The acoustic models were context-independent monophone models (the number of monophones was 19 because CENSREC-1 is a digit speech database), and the training was: (1) multi-condition training (baseline), (2) noise adaptive training, (3) static noise aware training, and (4) the proposed method. Each monophone was modeled by a left-to-right HMM, and each HMM has three output states. For (2), (3), and (4), we evaluated each utterance and each frame noise estimation.

The DNNs had several hidden layers from 3 to 8, and each hidden layer had 1024 units. The activation function was sigmoid at the hidden layer and softmax at the output layer. Before fine-tuning the DNNs, layer-wise pre-training was performed. Fine-tuning was performed by back-propagation with cross entropy criterion. The alignment label was made by monophone GMM-HMM. GMM-HMM used 38MFCCs (Mel-Frequency Cepstral Coefficients) as acoustic features and modeled 16 Gaussians per state.

We used MFCCs, log mel-filterbank output (FBANK), and gammatone filterbank output (GFBANK) as acoustic features. Each acoustic feature was extracted by using an analysis window of 25 ms and frame shift of 10 ms. The number of filterbank was unified 30 among MFCC, FBANK, and GFBANK. MFCC was 38 dimensional ($12MFCC + 13\Delta(MFCC + power) + 13\Delta\Delta(MFCC + power)$). FBANK and GFBANK have only static features. We performed log compression in MFCC and FBANK and root compression ($\cdot^{0.3}$) in GFBANK on each output of filterbank. When using acoustic features as the input of the DNN, the input of the DNN was normalized by the mean and variance of 0 and 1, and 11 consecutive frames (5 frames before and 5 frames after the current frame) we used as input at a time, so FBANK and GFBANK had a dimensional input vector of $30 \times 11 = 330$, and MFCC had a dimensional input vector of $38 \times 11 = 418$. GFBANK was extracted by multiplying the FFT spectrum by the impulse response of the gammatone filter.

We used Spectral Subtraction as a noise suppression technique. We set the subtraction factor $\alpha = 2.0$, and the flooring factor $\beta = 0.0$. For noise estimation per utterance, we set $M = 30$. When noise estimation per frame is based on minimum statistics, we set $\gamma = 0.995$, $\beta_{MS} = 0.8$, and $\alpha_s = 0.5$. After estimating the noise information, it was log compressed or root compressed in the same way as the acoustic features. The

Table 1: Optimal number of hidden layers at each condition: (1) multi condition training, (2) noise adaptive training, (3) static noise aware training, (4) proposed method

	feature	(1)	(2)	(3)	(4)
utter.	MFCC	3	7	3	5
	FBANK	5	3	3	3
	GFBANK	6	3	5	6
frame	MFCC	3	4	3	6
	FBANK	5	3	3	4
	GFBANK	6	3	7	7

estimated noise information was obtained in the filterbank domain; therefore, Discrete Cosine Transform (DCT) was applied to the estimated noise information when adding the information to MFCC.

When the noise estimation interval was frame, we could input 11 frames at a time into the DNN in the same way as the acoustic features. However, in the preliminary experiment, we compared inputting 11 frames and one frame of noise information at a time; inputting one frame at a time showed better performance than inputting 11 frames at a time. Therefore in these experiments we used inputting noise information one frame at a time.

When decoding, we used the WFST SPOJUS decoder [17].

4.2. Results

4.2.1. Comparison of each technique with noise estimation per utterance

Figure 3 shows the average digit accuracy for each approach. We expected that the optimal number of hidden layers is different in each condition; therefore, we made DNNs with different depths. Notice that Figure 3 shows the digit accuracy when the number of hidden layers is optimal at each condition. For FBANK, the proposed approach using noise estimation per utterance showed the best performance in the approaches we used. On the other hand, for MFCC and GFBANK, noise adaptive training showed the best performance. The reason for this may be the difference in feature extraction. As shown in the previous section, FBANK and its noise information was log compressed, and GFBANK and its noise information was root compressed. As a result of this difference, the complexity of the relationship between the feature and its noise information might have changed. As evidence of this, the optimal number of hidden layers was three in FBANK and six in GFBANK in the proposed approach using noise estimation per utterance (Table 1). Moreover, the optimal numbers of hidden layers in (2), (3), and (4) were all three in FBANK. By contrast, in GFBANK, the optimal number of hidden layers of approach (2) that use only speech features was three, and that of approach (3) and (4) that use both speech features and noise information was five and six. Therefore, it is thought that the DNN with features using root compression requires more deep architecture than that with features using log compression. A larger training set has been required to learn in a deeper network, but the database used in this study had a small training set. Hence, the deeper network could not learn well; consequently, approach (2), which was a sufficiently learned shallow network, had the best performance. MFCC performed DCT in FBANK and was similarly inclined; therefore, with DNN and MFCC, there could not be a good learnt relationship between speech features and noise information due to the effect of DCT.

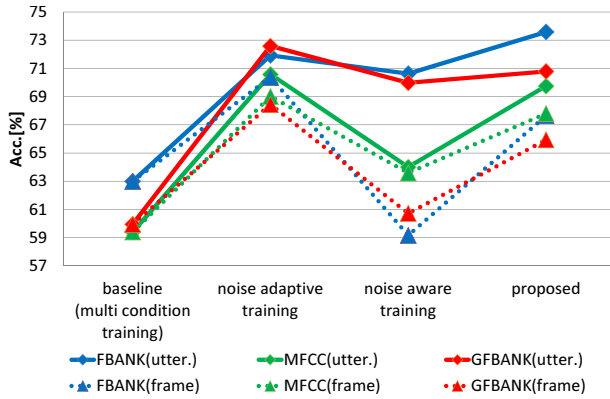


Figure 3: Average of digit accuracy for all noise conditions in CENSREC-1

4.2.2. Comparison between noise estimation per utterance and noise estimation per frame

In accordance with Figure 3, the approaches with noise estimation per utterance showed better performance than the approaches with noise estimation per frame. Table 2 shows the digit accuracy at each SNR with FBANK for further comparison between per utterance and per frame. When the SNR was high, the approaches with noise estimation per utterance had better performance than those per frame, but when the SNR was lower, the performance was significantly reduced. Noise estimation per utterance used the top M frames to estimate noise, and most of these frames were only noise. Moreover, the database we used included almost stationary noise. Hence, it was deduced that the noise estimation per utterance worked well. On another front, discriminating between speech and noise was very difficult at low SNR; therefore, it was guessed that the noise estimation per frame does not work well. Approaches (3) and (4) input estimated noise information into the DNN directory, so their performance with noise estimation per frame was worse than (2).

Table 2: Digit accuracy at each SNR with FBANK: (2) noise adaptive training, (4) proposed method

SNR	utterance		frame	
	(2)	(4)	(2)	(4)
20dB	97.38	98.03	99.09	97.84
15dB	95.29	96.25	97.61	94.65
10dB	90.25	91.32	92.77	87.24
5dB	79.33	79.50	78.03	71.48
0dB	51.89	53.94	48.55	43.45
-5dB	17.45	22.48	8.15	11.23
ave.	71.93	73.59	70.70	67.45

4.3. Experiments using oracle noise information

To confirm the utility of the proposed approach with noise estimation per frame, we conducted the same experiment by giving oracle noise information. To obtain the oracle noise information, we re-created the database with the same configuration as CENSREC-1. For the training set, we added the G.712 filtered noises used in set A of CENSREC-1 to the clean speech training data at each SNRs (20, 15, 10, and 5). When adding noise to speech, we used the same program, named FaNT [15], as AURORA-2, AURORA-4, and CENSREC-1. Similarly, we re-

created the evaluation set by adding noise to clean speech under the same conditions as set B of CENSREC-1. Notice that we could not compare the result of this experiment and that of the previous experiment strictly. We changed the sigmoid function to the Rectified Linear function as a hidden layer activation function for speeding up training, so normalization of the mean and variance were not performed. We used the estimated noise information when performing noise-suppression and used (1) the estimated noise information or (2) the oracle noise information as input noise information. We compared the digit accuracy of (1) and (2) with the noise estimation per frame.

Table 3 shows the digit accuracy at each SNR using the proposed approaches. It could be seen that the recognition accuracy was improved by giving oracle noise information to the DNN. Hence, it was shown that the proposed approach with noise information per frame is effective if the noise information is given to the DNN appropriately.

Table 3: Digit accuracy at each SNR with FBANK in proposed approach

SNR	utterance	frame	
	estimated	estimated	oracle
20dB	96.40	95.64	96.10
15dB	93.88	92.92	94.04
10dB	89.64	87.59	90.32
5dB	77.46	76.97	80.40
0dB	52.16	54.44	57.99
-5dB	20.39	21.86	23.66
ave.	71.66	71.57	73.45

5. Conclusions

In this paper, we proposed an approach to improve the noise robustness of a DNN acoustic model. Both the noise-suppressed acoustic features and the estimated noise information were input into the DNN. In the noisy speech recognition task, we compared the proposed approach and other approaches such as noise adaptive training and static noise-aware training. For noise estimation per utterance with FBANK, the proposed approach showed better performance than the other noise robust approach, and we obtained a 28.62% word error rate reduction compared with multi-condition training, and a 5.9% word error rate reduction compared with noise adaptive training. It was shown that giving the estimated noise information per utterance to the DNN helps improve the robustness of the DNN. Also, in the experiments using oracle noise information, it was shown that using the noise information per frame is effective.

The experiments were performed by a simple task with a small training set and relatively stationary noises. It was suggested that the DNN might not learn well. As future work, we will evaluate the performance of the proposed approach on large tasks in more non-stationary noise environments. Also, we will develop an accurate noise estimation technique for the proposed approach.

6. Acknowledgment

This work was partly supported by JSPS KAKENHI Grant Number 24500201.

7. References

- [1] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath and B. Kingsbury. "Deep Neural Networks for Acoustic Modeling in Speech Recognition", *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82-97, 2012.
- [2] D. Yu, M. L. Seltzer, J. Li, J-T. Huang and F. Seide. "Feature Learning in Deep Neural Networks - Studies on Speech Recognition Tasks", *CoRR*, vol.abs/1302.3605, 2013.
- [3] S. F. Boll. "Suppression of Acoustic Noise in Speech Using Spectral Subtraction", *IEEE Trans, Acoustic, Speech, Signal Processing*, vol.AASP-27, no. 2, pp. 113-120, 1979.
- [4] C. Kim and R. M. Stern. "Power-Normalized Cepstral Coefficients (PNCC) for Robust Speech Recognition", *Proc. ICASSP*, pp. 4101-4104, 2012.
- [5] L. Deng, A. Acero, M. Plumpe and X. Huang. "Large-vocabulary Speech Recognition Under Adverse Acoustic Environments", *Proc. ICSLP*, pp. 806-809, 2000.
- [6] M. Cooke, P. Green, L. Josifovski and A. Vizinho. "Robust Automatic Speech Recognition with Missing and Unreliable acoustic data", *Speech Communication*, vol. 34, no. 3, pp. 267-285, 2001.
- [7] P. Vincent, H. Larochelle, Y. Bengio and P-A. Manzagol. "Extracting and Composing Robust Features with Denoising Autoencoders", *Proc ICML*, pp. 1096-1103, 2008.
- [8] M.L. Seltzer, D. Yu and Y. Wang. "An Investigation of Deep Neural Networks for Noise Robust Speech Recognition", *Proc. ICASSP*, pp. 7398-7402, 2013.
- [9] B. Li and K. C. Sim. "An Ideal Hidden-Activation Mask for Deep Neural Networks based Noise-Robust Speech Recognition", *Proc. ICASSP*, pp. 200-204, 2014.
- [10] S. Xue, O. A-Hamid, H. Jiang and L. Dai. "Direct Adaptation of Hybrid DNN/HMM Model for Fast Speaker Adaptation in LVCSR based on Speaker Code", *Proc. ICASSP*, pp. 200-204, 2014.
- [13] S. Deng, J. Han, C. Zhang, T. Zheng and G. Zheng. "Robust Minimum Statistics Project Coefficients Feature for Acoustic Environment Recognition", *Proc.ICASSP*, pp. 8282-8286, 2014.
- [14] S. Nakamura, K. Takeda, K. Yamamoto, T. Yamada, S. Kuroiwa, N. Kitaoka, T. Nishimura, A. Sasou, M. Mizumachi, C. Miyajima, M. Fujimoto and T. Endo. "AURORA-2J: An Evaluation Framework for Japanese Noisy Speech Recognition", *IEICE transactions on Information and Systems*, vol. E88-D, No. 3, pp. 535-544, 2005.
- [15] H-G. Hirsch. FaNT: Filtering and Noise-Adding Tool, <http://dnt.kr.hs-niederrhein.de/download.html>
- [16] Y. Xu, J. Du, L.-R. Dai and C.-H. Lee. "Dynamic Noise Aware Training for Speech Enhancement Based on Deep Neural Networks", *Proc. Interspeech*, pp. 2670-2674,
- [17] Y. Fujii, K. Yamamoto, S. Nakagawa. "Large Vocabulary Speech Recognition System: SPOJUS++", *Proc. MUSP*, pp. 110-118, 2011.
- [18] K. Palomaki, G. J. Brown and J. Barker. "Missing data speech recognition in reverbrant condition", *Proc. ICASSP*, pp. 65-68, 2002.