DEEP AUTOENCODERS AUGMENTED WITH PHONE-CLASS FEATURE FOR REVERBERANT SPEECH RECOGNITION

Masato Mimura, Shinsuke Sakai, and Tatsuya Kawahara

Academic Center for Computing and Media Studies, Kyoto University
Sakyo-ku, Kyoto 606-8501, Japan
{mimura, sakai, kawahara}@ar.media.kyoto-u.ac.jp

ABSTRACT

This paper addresses reverberant speech recognition based on front-end processing using DAE (Deep AutoEncoder) coupled with DNN (Deep Neural Network) acoustic model. DAE can effectively and flexibly learn mapping from corrupted speech to the original clean speech based on the deep learning scheme. While this mapping is conventionally conducted only with the acoustic information, we presume the mapping is also dependent on the phone information. Therefore, we propose a new scheme (pDAE), which augments a phone-class feature to the standard acoustic features as input. Two types of the phone-class feature are investigated. One is the hard recognition result of monophones, and the other is a soft representation derived from the posterior outputs of monophone DNN. In the evaluation on the Reverb Challenge 2014 task, the augmented feature in either type results in a significant improvement (7-8% relative) from the standard DAE. It is also shown that using the soft representation in the training phase is critical.

Index Terms— Reverberant speech recognition, Deep Neural Networks (DNN), Deep Autoencoder (DAE)

1. INTRODUCTION

In recent years, the speech recognition technology based on statistical techniques achieved a remarkable progress supported by the ever increasing training data and the improvements in the computing resources. Applications such as voice search are now being used in our daily life. However, speech recognition accuracy in adverse environments such as those with reverberation and background noise is still at low levels. A key breakthrough for the speech recognition technology to be accepted widely in the society will be the methodology for hands-free input. This is critical for realizing conversational robots. Speech reverberation adversely influences the recognition accuracy in such conditions and various efforts have been made to solve this problem.

Reverberant speech recognition has been tackled by feature enhancement at the front-end and model adaptation at the back-end. One of the simplest approaches to feature enhancement is the cepstral mean normalization (CMN) [1]. However, since reverberation time is usually longer than the frame window length for feature extraction, its effectiveness is limited. A major back-end approach is the use of maximum-likelihood linear regression (MLLR) [2] that adapts the acoustic model parameters to the corrupted speech.

More sophisticated enhancement techniques for speech recognition have been investigated. Speech enhancement techniques include deconvolution approaches that reconstruct clean speech by inverse-filtering reverberant speech [3][4][5] and spectral enhancement approaches that estimate and remove the influences of the late reverberation [6][7]. Since an improvement measured by SNR may not be directly related to the speech recognition accuracy, there also are approaches to speech enhancement based on speech recognition likelihoods in the back-end [8].

Recently, following the great success of deep neural networks (DNN), dereverberation by deep autoencoders (DAE) has been investigated [9][10][11][12]. In these works, DAEs are trained using reverberant speech features as input and the clean speech features as target so that they recover the clean speech from corrupted speech in the recognition stage. DAE can effectively and flexibly learn mapping from corrupted speech to the original clean speech based on the deep learning scheme.

While this mapping is conventionally conducted only with the acoustic information, we presume the mapping is also dependent on the phone information. Since each dimension of the acoustic feature such as filterbank output has a different range of values depending on phones, the information on "which phone-class the current speech frame belongs to" should be helpful for DAE to recover the clean speech from reverberant speech. In this paper, we propose a new scheme of DAE, which incorporates a phone-class feature as additional input. We investigate two types of the phone-class feature: soft and hard features. We evaluate the effect of these features in the training and recognition stage through the ASR task of the Reverb Challenge 2014 [13].

After a brief review on DNNs for reverberant speech recognition (DAE front-end and DNN-HMM back-end) in Section 2, the detail of the proposed method is explained in Section 3. Experimental evaluations of the method are presented in Section 4 before the conclusion in Section 5.

2. DNN FOR REVERBERANT SPEECH RECOGNITION

Deep neural hidden Markov models (DNN-HMMs) have outperformed GMM-HMMs drastically in the wide range of speech recognition tasks [14][15][16][17] and become a state-of-the-art acoustic modeling method. One of the advantages of the DNN-HMMs is that they are good at exploiting multiple frames, which is vital especially for reverberant speech recognition where we need to handle long-term artifacts.

DNNs are also applied to front-end feature enhancement in robust speech recognition area [18][9][10][11][19][20]. DNNs used for regression tasks such as speech enhancement are often called DAEs [21]. Unlike DNNs for classification, DAEs are typically trained to reconstruct signals usually using error backpropagation with the mean squared error as the loss function [22]. DAEs for speech enhancement are trained using the clean speech features as

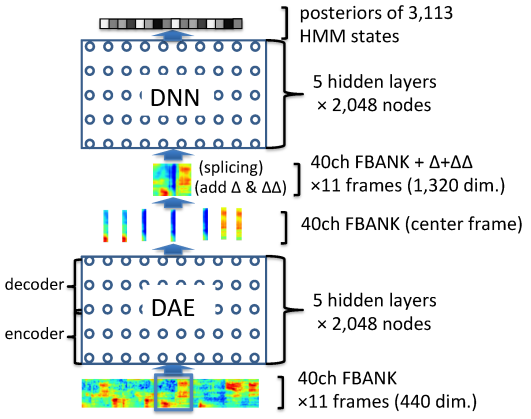

Fig. 1. Baseline system combining DAE front-end and DNN-HMM back-end

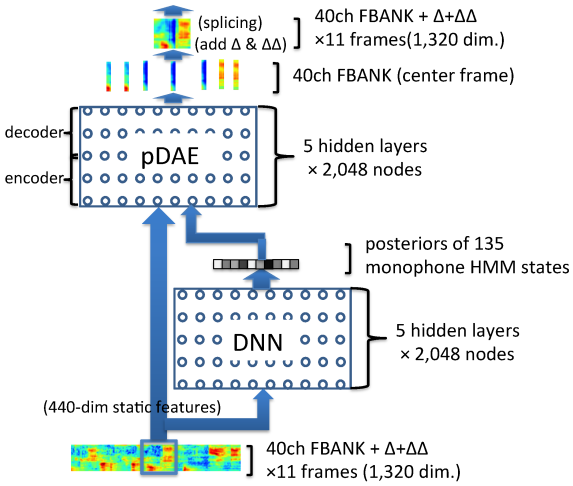

Fig. 2. Feature enhancement by pDAE using PC_{soft} features

target and the corrupted speech features as input (denoising autoencoders [23]). DAE-based dereverberation has also been investigated recently [9][10][11][12].

In [12], we proposed to use deep learning both in front-end (DAE-based dereverberation) and back-end (DNN-HMM acoustic model) in a reverberant speech recognition system. Since the dereverberation using the DAE is performed on not the STFT level ([9]) but the feature level ([10][11]) in our system, we can directly feed the DAE output to the DNN-HMM acoustic model. While the DNN-HMM trained using the multi-condition data significantly outperformed the MLLR-adapted baseline GMM-HMM system even when used alone, the combination of the multi-condition DNN-HMM and the DAE for dereverberation complementarily achieved further improvement in very adverse reverberant conditions. In this study, we also adopt this framework illustrated in Fig. 1. Our DNN-HMM models are built using the standard recipe described in [16]. The DAE has a vertically symmetric network structure and each layer is initialized by an RBM in the same manner as in [21]. We use the identity function as the output function of the DAE. The input feature vector of both of DNN and DAE consist of multiple frames of filterbank output. Note that the DNN-HMM model is trained using not the DAE-enhanced data but the original multi-condition data.

3. PROPOSED METHOD

3.1. pDAE using phone-class information

Recently, extension of the DNN-HMMs is ivestigated by augmenting additional information as input, for example speaker adaptation using I-Vectors [24][25] and noise-aware training [26]. Saon et al. [24] proposed a method to train a single network that conducts speaker adaptation and phone classification simultaneously by feeding I-Vectors (speaker identity features) to the network. Speaker identity features are helpful, considering that different speakers often use different pronunciations for the same phone.

In this work, we propose to augment a *phone-class feature* as an additional input of the DAE to enhance the dereverberation performance. Since the acoustic features in clean speech vary depending on phones, the phone-class information is expected to be helpful for the DAE to recover the clean speech from corrupted speech. We refer to this proposed DAE as *pDAE*. The training procedure of pDAE is same as the standard DAE, except that the input is augmented with an additional phone-class feature of the center frame of the input.

The concept of the proposed method is similar to the stochastic matching proposed by Sankar et al. [27], where the feature normalization is conducted by a function that depends on the phone information. In this work, the mapping is done by the more general deep learning schema (DAE).

3.2. Phone-class features

Two types of the phone-class feature are investigated in this study: soft and hard features.

The soft phone-class feature PC_{soft} is a soft representation of phone classification. It is derived with phone state posteriors calculated with a DNN trained for phone state classification. Note that we use monophone state posteriors (135-dimensional) instead of triphone state posteriors (3,113-dimensional) which are used in the acoustic model, in order to keep the dimensionarity of the input vector not much larger than the original vector (440-dimensional) [1]. The monophone DNN was trained using the same multi-condition data used for the triphone DNN training. pDAE using the PC_{soft} feature is illustrated in Fig. 2. This is similar to the MLP-derived features used in the TANDEM approach [28][29][30].

The hard phone-class feature is encoded using a 1-of-K scheme. The element corresponding to the phone-class which has the largest posterior probability is 1, and all other classes are 0. We can simply use the hard version of the DNN-derived PC_{soft} feature, which is referred to as PC_{hard}.

In the training data for which manual transcription is available, we can derive the oracle PC_{hard} feature. We refer to this specific type of the hard feature as PC_{hard}^{oracle}. We can generate a phone HMM state label for each frame in the training data by performing forced alignment using the manual transcription. In the PC_{hard}^{oracle} feature vector, the element corresponding to the correct state is 1.

However, we cannot use the PC_{hard}^{oracle} feature in the recognition stage. Instead, we use the speech recognition result of the test data

[1] Actually, the performance of the pDAE was degraded slightly when using triphone DNN posteriors as the phone-class feature in a preliminary experiment.

Table 1. Speech recognition performance on Reverb Challenge 2014 test set (word error rate (%))

		SimData							RealData		
		Room 1		Room 2		Room 3		Ave.	Room 1		Ave.
		Near	Far	Near	Far	Near	Far		Near	Far	
(1)	GMM-HMM (mc, w MLLR)	12.39	12.71	14.23	26.23	17.11	33.92	19.43	42.89	42.27	42.59
(2)	DNN-HMM (cln)	6.85	10.22	16.18	45.52	23.12	60.25	27.05	65.25	66.78	65.99
(3)	DNN-HMM (cln) + DAE	6.25	6.78	7.65	13.67	9.04	16.75	**10.03**	30.66	31.87	**31.25**
(4)	DNN-HMM (cln) + pDAE (PC_{soft})	5.51	6.44	7.06	12.74	8.17	14.26	**9.04**	27.37	26.60	**27.00**
(5)	DNN-HMM (cln) + pDAE (PC_{hard}^{decode})	5.18	6.12	7.14	12.57	7.66	12.42	**8.54**	27.75	26.60	**27.20**
(6)	DNN-HMM (mc)	5.42	6.37	7.27	12.56	7.85	12.90	8.74	28.59	30.87	29.67
(7)	DNN-HMM (mc) + DAE	9.30	9.69	8.36	11.92	9.30	15.25	**10.62**	24.37	25.52	**24.93**
(8)	DNN-HMM (mc) + pDAE (PC_{soft})	8.59	9.13	7.77	11.53	8.74	13.53	**9.87**	23.47	23.09	**23.29**
(9)	DNN-HMM (mc) + pDAE (PC_{hard}^{decode})	7.29	7.86	7.48	10.87	8.09	11.06	**8.78**	22.74	22.96	**22.85**

to derive a hard phone-class feature, referred to as PC_{hard}^{decode}. The phone HMM state labels are generated by performing forced alignment using the initial recognition result. The PC_{hard}^{decode} feature is expected to be more reliable than the PC_{hard} feature, because the initial recognition result is generated using a triphone model as well as a language model. But computation of the PC_{hard}^{decode} feature requires an extra decoding pass and is not suitable for on-line real-time processing. Note that it is not straightforward to derive a soft feature on the monophone states from the recognition result.

In our implementation, the dimension of these features is set to be same as the number of the states in the baseline monophone GMM-HMM. Therefore, we can use different types of the phone-class feature in the training and recognition stage.

4. EXPERIMENTAL EVALUATIONS

4.1. Task and data set

The proposed system was evaluated following the instructions for the task of the Reverb Challenge 2014 [13]. For training, we used the standard multi-condition data that is generated by convolving clean WSJCAM0 data with room impulse responses (RIRs) and subsequently adding noise signals. The amount of the training data is 15.5 hours (7,861 utterances). Evaluation data consists of "SimData" and "RealData". SimData is a set of reverberant speech generated by convolving clean speech with various RIRs and adding measured noise signals to make the resulting SNR to be 20dB. RIRs were recorded in three different-sized rooms (small, medium, and large) and with two microphone distances (near=50cm and far=200cm). The reverberation time (T60) of the small, medium, and large rooms are about 0.25s, 0.5s, and 0.7s, respectively. RealData was recorded in a different room from those used for measuring RIRs for SimData. It has a reverberation time of 0.7s. There are two microphone distances in RealData, which are near (≈100cm) and far (≈250cm).

In the experiments in this paper, we only use a single channel both for training and testing. For decoding, we used the HDecode command from HTK-3.4.1 with a small modification to handle DNN output. The language model we used is the standard WSJ 5K trigram model. The training tools for the DNN-HMM and DAE were implemented using Python.

4.2. Evaluation of DAE coupled with DNN-HMM

Here we describe the details of the baseline system illustrated in Fig. 1.

A 1320-dimensional feature vector consisting of eleven frames of 40-channel log Mel-scale filterbank outputs and their delta and acceleration coefficients is used as input to the DNN-HMM. We performed utterance-based mean normalization as well as global mean and variance normalization to these feature vectors. The targets are chosen to be the 3,113 shared states of the baseline GMM-HMM. The six-layer network consists of five sigmoidal hidden layers and a softmax output layer. Each of the hidden layers consists of 2,048 nodes. The network is initialized using RBMs trained with multi-condition data. The fine-tuning of the DNN is performed by error backpropagation supervised by state labels using cross entropy as the loss function. The training parameters such as the learning rate are same as those in [12]. We trained two DNN-HMM systems ("DNN-HMM (mc)", "DNN-HMM (cln)") using the multi-condition data and the clean version of the training data.

The input for the DAE was an eleven-frame sequence of 40-channel log Mel-scale filterbank outputs (440-dimensional). The target for the DAE was one frame (40-dimensional) of the clean speech which corresponds to the center frame of the input. The DAE is trained using reverberant speech as the input and clean speech as the target. The network is also initialized using RBMs trained with multi-condition data. The autoencoder network has seven layers in total including five sigmoidal hidden layers. The number of nodes in each layer is 2,048 except for input and output layers (Fig. 1). The fine-tuning of the DAE was performed by error backpropagation with mean squared error as the loss function. The training parameters are same as those in [12]. The delta and acceleration parameters are added to the DAE output after global normalization. We splice the output of the eleven consecutive frames before feeding to the DNN-HMM.

The evaluation results with the baseline systems are shown in row (2), (3), (6) and (7) in Table 1. The DNN-HMM trained using the multi-condition data outperformed the MLLR-adapted GMM-HMM drastically in all conditions (from row (1) to row (6)), concluding that the multi-condition training of DNN-HMM is very effective for reverberant speech recognition. The speech feature enhancement using the DAE improved the performance of the DNN-HMM trained with the clean data drastically (row (3)), which has a very high WER when used alone (row (2)). The combination of the DAE front-end and the multi-condition DNN-HMM significantly improved the WER for very adverse "RealData" conditions (row (7)), while it was not effective for "SimData" conditions, which have similar RIRs to the training data. [2]

[2] Although these results have the same tendency as those reported in [12],

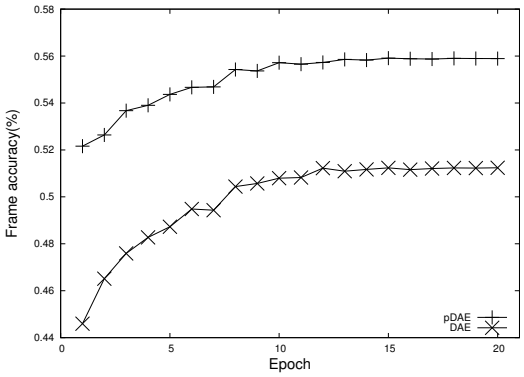

Fig. 3. Frame accuracy by DAE and pDAE on held-out set

4.3. Evaluation of the proposed method (pDAE)

We evaluated the dereverberation performance of the proposed pDAE. The training procedure for pDAE is same as that for the baseline DAE, except that the input vector is augmented with the 135-dimensional phone-class feature. The input layer of pDAE has 575 (135 + 440) nodes. The mean and variance of the phone-class feature vector are normalized in the same manner as the filterbank feature. The frame accuracy obtained on the held-out set during the fine-tuning of the pDAE and the baseline DAE is shown in Fig. 3. Here we used PC_{soft} as the phone-class feature. The frame accuracy was calculated using the clean DNN-HMM back-end and outputs of the DAE and the pDAE at the end of each epoch. We observe that the DAE augmented with the DNN state posteriors (pDAE) is consistently better than the baseline DAE, suggesting that feeding the phone-class information to the DAE is effective.

The evaluation results with the combination of the pDAE (PC_{soft}) and the clean DNN-HMM back-end are shown in Table 1, row (4). In all of "SimData" conditions, the WER was reduced from the baseline DAE (Table 1, row (3)). In more adverse "Far" conditions in "Room 2" and "Room 3", the improvements were larger. In "RealData" conditions, the WER was reduced by 4.25 points from the baseline DAE, and the improvement was significantly higher than in "SimData". The phone-class information is more effective when the mismatch between the training data and the test data is larger.

The WER with the combination of the pDAE (PC_{soft}) and the multi-condition DNN-HMM back-end is shown in Table 1, row (8). In all of "SimData" conditions, the performance degradation observed in the combination of the multi-condition DNN-HMM and the standard DAE front-end (Table 1, row (7)) was mitigated by using the pDAE. In "RealData", the average WER was reduced by 1.64 points from the baseline DAE, confirming that the phone information is effective even when using the multi-condition DNN-HMM back-end, which is more robust for reverberant speech. The improvement from the standard DAE in both "SimData" and "RealData" conditions is statistically significant at the 1% level.

the WERs here are much lower than in [12] mainly because we used the trigram language model in this paper and the conclusions are more statistically reliable.

Table 2. Comparison of phone-class features (word error rate (%))

recognition \ training	PC_{soft}	PC_{hard}^{oracle}
PC_{soft}	23.29	24.10
PC_{hard}	23.27	24.34
PC_{hard}^{decode}	22.85	23.29
(cf.) PC_{hard}^{oracle}	13.74	14.25

4.4. Comparison of soft and hard phone-class features

Next we compared the two types of the phone-class feature described in Sec. 3.2. We evaluated six different combinations of the features in the training and recognition stage through speech recognition experiments on "RealData" using the multi-condition DNN-HMM back-end.

Comparison of PC_{soft} and PC_{hard}^{oracle} in the training stage is shown in the two columns in Table 2. The dereverberation performance of the pDAE is degraded by using the PC_{hard}^{oracle} feature in the training stage, whichever type of the phone-class feature is used in the recognition stage, although the PC_{hard}^{oracle} feature is more accurate than the PC_{soft} feature. One of the reason for this may be that the PC_{soft} feature has richer information.

We also conducted an oracle experiment where we used the PC_{hard}^{oracle} feature derived from the manual transcription of the test data. As shown in the last row of Table 2, the WER was surprisingly reduced, which clearly confirms our hypothesis that phone-class information is useful for DAE-based dereverberation. However, the hard version of the PC_{soft} feature (PC_{hard}) did not yield any improvement.

On the other hand, the results with the PC_{hard}^{decode} feature derived from the initial recognition result is better than those with the PC_{soft} feature as expected, though it requires another recognition pass. The WER in all reverberant conditions including "SimData" obtained with the PC_{hard}^{decode} feature is shown in row (5) and (9) in Table 1. When combined with the multi-condition DNN, the WER was further reduced in all conditions from the PC_{soft} feature (row (4) and (8)).

5. CONCLUSION

We have proposed a novel approach to reverberant speech recognition with front-end preprocessing using deep autoencoders (DAE) augmented with the phone-class information. The proposed method significantly and consistently improved the recognition accuracy in all reverberant conditions. We compared two types of the phone-class feature and concluded that the PC_{soft} feature which does not require an extra decoding step is enough for significant improvement, while using the PC_{hard}^{decode} feature in the recognition stage can yield further improvement. It is also shown that using the PC_{soft} feature is more effective than the PC_{hard}^{oracle} feature in the training phase.

The average WER on "RealData" obtained with the proposed pDAE using the PC_{hard}^{decode} feature (Table 1, row (9)) was 1.0 points better than the best result in the same condition ("1ch", "no own data", "no full batch") of Reverb Challenge 2014.

Acknowledgements: This work was supported by JST CREST and ERATO programs.

6. REFERENCES

[1] A.E.Rosenberg, C.H.Lee, and F.K.Soong, "Cepstral channel normalization techniques for HMM-based speaker verification," in *ICSLP*, 1994, pp. 1835–1838.

[2] C.J.Leggetter and P.C.Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," in *Computer Speech and Language*, 1995, vol. 9, pp. 171–185.

[3] M.Gurelli and C.Nikias, "Evam: An eigenvector-based algorithm for multichannel blind deconvolution of input colored signals," *IEEE Trans. Audio, Speech & Language Process.*, vol. 43, no. 1, pp. 134–149, 1995.

[4] M.Delcroix, T.Hikichi, and M.Miyoshi, "On the use of lime dereverberation algorithm in an acoustic environment with a noise source," in *ICASSP*, 2006, vol. 1.

[5] S.Gannot and M.Moonen, "Subspace methods for multimicrophone speech dereverberation," in *EURASIP J.Appl.Signal Process.*, 2003, vol. 11, pp. 1074–1090.

[6] M.Wu and D.Wang, "A two-stage algorithm for one-microphone reverberant speech enhancement," *IEEE Trans. Audio, Speech & Language Process.*, vol. 14, no. 3, pp. 774–784, 2006.

[7] K.Kinoshita, M.Delcroix, T.Nakatani, and M.Miyoshi, "Suppression of late reverberation effect on speech signal using long-term multiplestep linear prediction," *IEEE Trans. Audio, Speech & Language Process.*, vol. 17, no. 4, pp. 534–545, 2009.

[8] R.Gomez and T.Kawahara, "Robust speech recognition based on dereverberation parameter optimization using acoustic model likelihood," *IEEE Trans. Audio, Speech & Language Process.*, vol. 18, no. 7, pp. 1708–1716, 2010.

[9] T.Ishii, H.Komiyama, T.Shinozaki, Y.Horiuchi, and S.Kuroiwa, "Reverberant speech recognition based on denoising autoencoder," in *INTERSPEECH*, 2013, pp. 3512–3516.

[10] Xue Feng, Yaodong Zhang, and James Glass, "Speech Feature Denoising and Dereverberation via Deep Autoencoders for Noisy Reverberant Speech Recognition," in *Proc. ICASSP*, 2014, pp. 1778–1782.

[11] Felix Weninger, Shinji Watanabe, Yuuki Tachioka, and Björn Schuller, "Deep Reccurent De-noising Auto-encoder and Blind De-reverberation for Reverberated Speech Recognition," in *Proc. ICASSP*, 2014, pp. 4656–4660.

[12] M.Mimura, S.Sakai, and T.Kawahara, "Exploiting Deep NeuralL Networks and Deep Autoencoders in Reverberant Speech Recognition," in *HSCMA*, 2014.

[13] K.Kinoshita, M.Delcroix, T.Yoshioka, T.Nakatani, E.Habets, R.Haeb-Umbach, V.Leutnant, A.Sehr, W.Kellermann, R.Maas, S.Gannot, and B.Raj, "The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA-13)*, 2013.

[14] G.E.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.

[15] A.Mohamed, G.Dahl, and G.Hinton, "Acoustic modelling using deep belief networks," *IEEE Trans. Audio, Speech, & Language Proc.*, vol. 20, no. 1, pp. 14–22, 2012.

[16] G.E.Dahl, D.Yu, L.Deng, and A.Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Trans. Audio, Speech, & Language Proc.*, vol. 20, no. 1, pp. 30–42, 2012.

[17] F.Seide, G.Li, and D.Yu, "Conversational speech transcription using context-dependent deep neural networks," in *INTERSPEECH*, 2011, pp. 437–440.

[18] Li Deng, Mike Seltzer, Dong Yu, Alex Acero, Abdel rahman Mohamed, and Geoffrey Hinton, "Binary coding of speech spectrograms using a deep auto-encoder," in *INTERSPEECH*, 2010, pp. 1692–1695.

[19] X.Lu, Y.Tsao, S.Matsuda, and C.Hori, "Speech enhancement based on deep denoising autoencoder," in *INTERSPEECH*, 2013, pp. 436–440.

[20] Jun Du, Qing Wang, Tian Gao, Yong Xu, Lirong Dai, and Chin-Hui Lee, "Robust speech recognition with speech enhanced deep neural networks," in *INTERSPEECH*, 2014, pp. 616–620.

[21] G.E.Hinton and R.R.Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, pp. 504–507, 2006.

[22] Y.Bengio, P.Lamblin, D.Popovici, and H.Larochelle, "Greedy layer-wise training of deep networks," in *in Advances in Neural Information Processing Systems 19 (NIPS06)*, 2007, pp. 153–160.

[23] P.Vincent, H.Larochelle, Y.Bengio, and P.A.Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML'08)*, 2008, pp. 1096–1103.

[24] G.Saon, H.Soltau, D.Nahamoo, and M.Picheny, "Speaker Adaptation of Neural Network Acoustic Models Using I-Vevtors," in *Proc. ASRU*, 2013, pp. 55–59.

[25] Penny Karanasou, Yongqiang Wang, Mark J.F. Gales, and Philip C. Woodland, "Adaptation of deep neural network acoustic models using factorised i-vectors," in *INTERSPEECH*, 2014, pp. 616–620.

[26] M.Seltzer, D.Yu, and Y.Wang, "An Investigation of deep neural networks for noise robust speech recognition," in *Proc. ICASSP*, 2013, pp. 7398–7402.

[27] Ananth Sankar and Chin-Hui Lee, "A Maximum-Likelihood Approach to Stochastic Matching for Robust Speech Recognition," *IEEE Trans. Speech & Audio Process.*, vol. 4, no. 3, pp. 190–202, 1996.

[28] N.Morgan, "Deep and wide: Multiple layers in automatic speech recognition," *IEEE Trans. Audio, Speech, & Language Proc.*, vol. 20, no. 1, pp. 7–13, 2012.

[29] G.S.V.S.Sivaram and H.Hermansky, "Sparse multilayer perceptron for phoneme recognition," *IEEE Trans. Audio, Speech, & Language Proc.*, vol. 20, no. 1, pp. 23–29, 2012.

[30] P.J.Bell, M.J.F.Gales, P.Lanchantin, X.Liu, Y.Long, S.Renals, P.Swietojanski, and P.C.Woodland, "Transcriptions of multigenre media archives using out-of-domain data," in *Proc. SLT*, 2012, pp. 324–329.