INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Joint Optimization of Denoising Autoencoder and DNN Acoustic Model Based on Multi-target Learning for Noisy Speech Recognition

Masato Mimura, *Shinsuke Sakai*, *Tatsuya Kawahara*

Kyoto University, School of Informatics,
Sakyo-ku, Kyoto 606-8501, Japan

Abstract

Denoising autoencoders (DAEs) have been investigated for enhancing noisy speech before feeding it to the back-end deep neural network (DNN) acoustic model, but there may be a mismatch between the DAE output and the expected input of the back-end DNN, and also inconsistency between the training objective functions of the two networks. In this paper, a joint optimization method of the front-end DAE and the back-end DNN is proposed based on a multi-target learning scheme. In the first step, the front-end DAE is trained with an additional target of minimizing the errors propagated by the back-end DNN. Then, the unified network of DAE and DNN is fine-tuned for the phone state classification target, with an extra target of input speech enhancement imposed to the DAE part. The proposed method has been evaluated with the CHiME3 ASR task, and demonstrated to improve the baseline DNN as well as the simple coupling of DAE with DNN. The method is also effective as a post-filter of a beamformer.

Index Terms: Speech Recognition, Speech Enhancement, Deep Neural Network (DNN), Denoising Autoencoder (DAE)

1. Introduction

Speech reverberation and additive noise adversely influence the speech recognition accuracy when the microphone is distant, and there is increasingly a great need for robust ASR systems. To this end, a number of efforts have been made on front-end signal enhancement, as well as robust modeling for back-end recognizers. Major approaches to the front-end enhancement include traditional optimal filtering techniques such as Wiener filtering [1] and spectral subtraction [2], parametric feature-based methods such as SPLICE [3], and more recently exemplar-based methods [4][5]. Multi-channel enhancement such as beamforming has also been explored.

Following the great success of deep neural networks (DNN) in acoustic modeling [6], speech feature enhancement using a class of DNNs, often referred to as denoising autoencoders (DAEs), has been investigated as well [7][8][9][10][11][12][13][14]. In these works, DAEs are trained to map a corrupted speech observation to a clean one and have achieved significant ASR performance improvements. A remarkable advantage of DAEs is their ease of deployment. The DAE-based speech enhancement is typically conducted at the feature level, and the enhanced features can be directly used in the back-end DNN-HMM acoustic model [15][16] without much latency.

However, the conventional DAE approach has a fundamental problem. While the goal of the front-end speech enhancement is to retain useful information for speech recognition while minimizing distortion caused by the corrupting noise, the objective function used for the DAE training considers only the latter criterion, which is the mean squared error (MSE) between the enhanced features and the clean features. Therefore, there can be a mismatch between the DAE output and the acoustic models, which can limit or even seriously degrade recognition accuracy [10][14]. To address this problem, we propose joint optimization of DAE and DNN acoustic model based on a multi-target learning scheme. In the first training stage for the DAE front-end, we incorporate the back-end classification error to the DAE training objective so that the enhanced features retain useful information for the acoustic model. In the following unified training stage, we add an MSE objective in order for the front-end part to retain the enhancement characteristics during the discriminative fine-tuning of the entire network for senone classification.

After we introduce two baseline methods for DNN-based speech feature enhancement in Section 2, the detail of the proposed method is explained in Section 3. Experimental evaluations are presented in Section 4 before the summary of related works in Section 5 and the conclusion in Section 6.

2. DNN-based speech enhancement for ASR

2.1. Denoising autoencoder (DAE)

A straightforward use of DNN for speech enhancement is to train a network for regression to map corrupted speech features to clean speech features [17]. This type of DNNs for regression tasks are often called deep autoencoders [18], and we refer to a particular class of deep autoencoders for speech enhancement as *denoising autoencoders* (DAEs). Unlike DNNs for classification, DAEs are typically trained to reconstruct signals by using the MSE as the objective function [19]:

$$E_{\mathbf{enh}} = \frac{1}{2}||\boldsymbol{z}_f^{L_f} - \boldsymbol{y}||^2 \quad (1)$$

where $\boldsymbol{z}_f^{L_f}$ designates the DAE output for a corrupted speech observation, i.e. the final layer output of a DAE with L_f layers and $\boldsymbol{y}$ is the corresponding clean speech observation.

We need to be aware that the DAE is not always guaranteed to improve speech recognition performance, because it only tries to minimize the MSE and takes no account of information on the back-end acoustic model. The mismatch between the DAE output and the back-end model becomes serious [10][14] especially when the back-end is a multi-condition model trained using a variety of noisy data, which is known to be effective for robust ASR. Some of discriminant information contained in noisy features can be lost through the highly non-linear "denoising" process.

http://dx.doi.org/10.21437/Interspeech.2016-388

2.2. Feature-space adaptation with DNN

An alternative DNN-based enhancer is a network which conducts feature-space adaptation so that the mapped features match the back-end acoustic model. This approach has been conventionally investigated in the speaker adaptation area [20][21][22][23][24][25]. The training procedure for this *adaptation network* is as follows. After a back-end DNN is trained, typically using clean data, a new DNN is attached to its input layer. It is trained so that the senone classification performance of the back-end DNN is improved typically using the cross-entropy criterion:

$$E_{\mathbf{ce}} = -\sum_i s_i \log z_{bi}^{L_b} \tag{2}$$

where s_i is the i-th element of the 1-0 encoded vector representation of the ground-truth senone label for the input observation and $z_{bi}^{L_b}$ is the i-th element of the output from the L_b-layer back-end DNN for senone classification. Note that the parameters of the back-end DNN are fixed during this front-end training stage. The adaptation network is less susceptible to the mismatch problem that the DAEs for enhancement suffers from, since it is trained to improve the performance of the acoustic model.

3. Joint optimization with multi-target learning

3.1. Front-end DAE training

Based on the discussion in Section 2, we considered an integrated approach. The mismatch problem inherent in the DAE may be mitigated by incorporating the errors from the back-end DNN for senone classification. Therefore, we propose a new front-end DAE training method which uses both enhancement and classification criteria. The objective function for the proposed *multi-target DNN training* is defined as:

$$E_{\mathbf{multi}} = \lambda E_{\mathbf{ce}} + (1-\lambda)\gamma E_{\mathbf{enh}} \tag{3}$$

Here, λ is a weight for the senone classification at the back-end, and γ is a parameter to calibrate the large difference in order of magnitude between the errors from the senone classification and the MSE of enhancement. Practically, we can set γ to be the ratio of the optimized learning rate for the DAE to that for the back-end DNN. The partial derivative of this unified objective with regard to the vector of inputs $\boldsymbol{u}_f^{L_f}$ to the activation in the output layer L_f of the front-end DNN, which we need for the delta rule of backpropagation, is calculated as:

$$\boldsymbol{\delta}_f^{L_f} = \frac{\partial E_{\mathbf{multi}}}{\partial \boldsymbol{u}_f^{L_f}} = \lambda \boldsymbol{W}_b^1 \boldsymbol{\delta}_b^1 + (1-\lambda)\gamma(\boldsymbol{y} - \boldsymbol{z}_f^{L_f}) \tag{4}$$

where $\boldsymbol{W}_b^1$ is the weight matrix for the first layer of the back-end DNN and $\boldsymbol{\delta}_b^1$ is the vector of partial derivatives of $E_{\mathbf{ce}}$ with regard to the input $\boldsymbol{u}_b^1$ to the first-layer activation function of the back-end DNN. $\boldsymbol{z}_f^{L_f}$ is the output vector of the front-end DAE and $\boldsymbol{y}$ is the corresponding clean speech observation. $\boldsymbol{u}_f^{L_f}$ and $\boldsymbol{z}_f^{L_f}$ are identical when the output activation function is the identity function as in the current case.

Figure 1 illustrates the multi-target learning procedure for the front-end DAE. Initially, the back-end DNN acoustic model is trained using clean or noisy training data and senone labels for them. Next, we construct and initialize the front-end DAE.

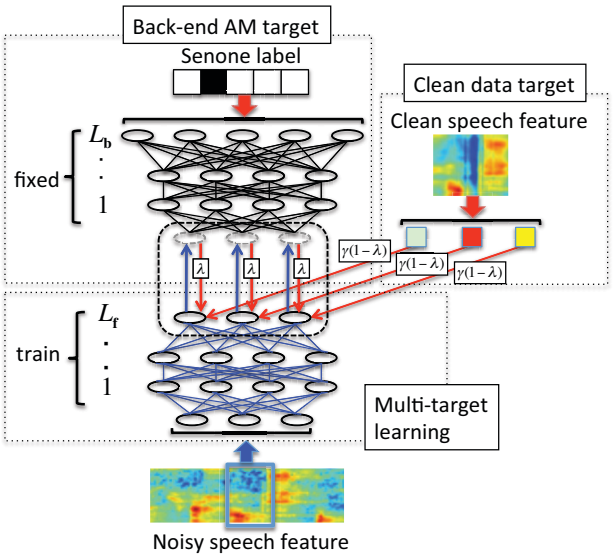

Figure 1: Front-end DAE training with multi-target learning

For each observation vector x, the front-end DAE output $\boldsymbol{z}_f^{L_f}$ is computed. It is then input to the back-end DNN and the frame classification result $\boldsymbol{z}_b^{L_b}$ is obtained. The classification error for the correct senone vector $\boldsymbol{s}$ is calculated as in (2) and propagated back to the first layer derivative $\boldsymbol{\delta}_b^1$, which appears in the first term of equation (4). Together with the second term which is the derivative of the enhancement error defined in (1), the error derivative is propagated to $\boldsymbol{\delta}_f^{L_f}$ for the output layer of the front-end DAE using equation (4). This is propagated down to the first layer of the front-end DAE and only the network parameters of the front-end are updated. Based on the training criterion of this procedure, the resulting front-end is expected to enhance corrupted speech while retaining as much information for senone classification. Note that this (single-task) multi-target learning scheme is different from the multi-task learning where each task has its own output layer and a respective objective function.

3.2. Unified network training

Once we have an improved front-end for speech enhancement, it is natural to consider re-training of the back-end model using training data enhanced by this front-end. However, it is empirically known that this simple approach does not always work as expected [17]. On the other hand, it has been reported with GMM-HMM based ASR that the speech recognition performance can be improved when the feature enhancement front-end and the back-end classifier are optimized jointly [26][27]. Inspired by these works, in this paper, we propose a novel method that re-train the back-end DNN for senone classification jointly with the front-end DNN for enhancement using the clean data as a constraint.

The unified network made by connecting the front-end and the back-end vertically can be regarded as a single very deep network of depth $L_f + L_b$ for senone classification. Therefore, we can train the entire network by back-propagation with the objective of mapping the noisy speech observation to the ground-truth senone [28][13]. Moreover, we add an additional regression objective for the front-end part to minimize the MSE between its output (i.e. layer L_f in Figure 2) and the clean speech observation, expecting that the front-end part is fur-

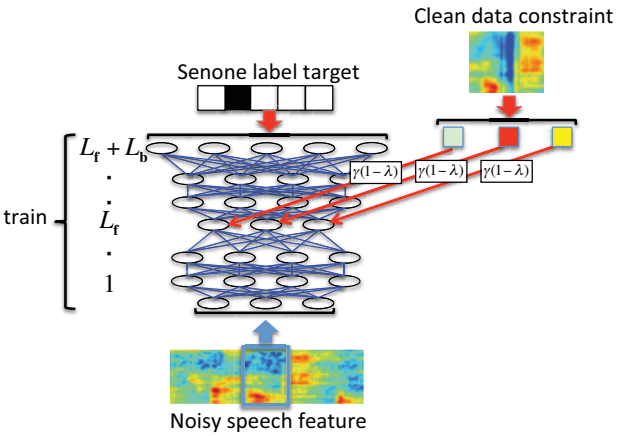

Figure 2: Unified network training with multi-target learning

ther optimized for classification while retaining its enhancement characteristics. The partial derivative of the unified objective function at the interface between the front-end part and the back-end part (i.e. layer L_f) of the unified network with regard to $\boldsymbol{u}_u^{L_f}$ is calculated as:

$$\boldsymbol{\delta}_u^{L_f} = \lambda \boldsymbol{W}_u^{L_f+1} \boldsymbol{\delta}_u^{L_f+1} + (1-\lambda)\gamma(\boldsymbol{y} - \boldsymbol{z}_u^{L_f}), \qquad (5)$$

which is essentially the same as equation (4).

4. Experimental evaluations

4.1. Task and data set

We evaluated the proposed methods through the ASR task of the third CHiME challenge [29]. The noisy training set consists of 1,600 real noisy utterances and 7,138 simulated noisy utterances generated by artificially mixing the clean WSJ0 training set with noisy backgrounds. There are four different types of noisy environments.

We trained three types of DNN-HMM acoustic models using the original noisy data, the enhanced data generated by applying a filter-and-sum beamformer using 5 channels of input, and the clean version of the training data. The beamformer we used is BeamformIt [30]. The training tool for the DNN was implemented in Python using CUDAMat [31]. The back-end DNN of each model has six hidden layers with 2k rectified linear units (ReLUs) and a softmax output layer with 2k nodes. A 1,320-dimensional feature vector consisting of 11 frames of 40-channel log Mel-scale filterbank outputs and their delta and acceleration coefficients is used as input. Dropout [32] is used for training of all hidden layers. The initial learning rate was set to be 0.04. For decoding, we used the Kaldi WFST decoder [33]. The language model is the standard WSJ 5k trigram LM.

We used the real noisy evaluation set ("et05_real") consisting of 1,320 utterances for evaluating the methods, and the real noisy development set ("dt05_real") for tuning the hyper-parameters such as the number of layers. We used the beamformed version of the evaluation set ("beam") to test the effectiveness of our methods when used as a post-filter of a beamformer, as well as the original noisy evaluation data ("noisy").

4.2. Effect of front-end training

The front-end DAE was trained based on the multi-target training scheme described in Section 3.1 for noisy and beamforming enhanced speech, using the entire 8,738-utterance noisy training set and the same set enhanced by beamforming, respectively. The 7,138 clean WSJ0 utterances and 1,600 headset utterances were used for clean regression target. The input is augmented with 137-dimensional phone-class features [34] derived from the posterior output of a monophone DNN. We also trained a conventional DAE front-end for enhancement (FE_{enh} in Section 2.1) and an adaptation network (FE_{adap} in Section 2.2) as baselines using the objective function described in Section 2. The number of the output units of all three types of front-end DNNs is set to be the same as input units so that we can feed them directly to the back-end DNNs. The activation function of the output layer is the identity function for all of them. The number of layers in each front-end DNN has been optimized using the development set and turned out to be 3, 4 and 5 for FE_{adap}, FE_{enh} and the proposed multi-target trained front-end (FE_{multi}, hereafter), respectively. They all have 2k ReLU units in their hidden layers and were trained using the Dropout technique. The optimized learning rates for FE_{adap}, FE_{enh}, and FE_{multi} were 0.02, 0.001 and 0.01, respectively. Accordingly, we set γ in formula (3) to be 0.05 (= 0.001/0.02). The clean data target for FE_{enh} and FE_{multi} is a vector consisting of 11 frames of clean versions of the training set (i.e. WSJ0 or headset).

Figure 3 illustrates frame accuracy on the held-out set during training of three types of front-ends[1]. The clean back-end model and the beamformed features are used here. While FE_{enh} improves the performance of the clean back-end as the training proceeds, FE_{adap} is better and continues to get better, probably because it directly uses the back-end classification as objective. However, the best frame accuracies are achieved with FE_{multi} that uses both targets as the training proceeds.

The recognition results on the real evaluation set in two conditions ("noisy" and "beam") obtained with the two baseline methods coupled with the clean acoustic model are shown in rows (2) and (3) in Table 1. Different DNN front-ends trained using the noisy and beamformed versions of the training set

Table 1: Performance of proposed methods combined with clean acoustic model back-end (WER(%))

Scheme	et05_real	
	noisy	beam
(1) no front-end	51.05	31.72
(2) FE_{adap} (classification target)	31.04	21.55
(3) FE_{enh} (enhancement target)	30.34	20.10
(4) FE_{multi} **(multi-target)**	**28.65**	**19.30**
(5) + back-end re-training	27.25	19.54
(6) **multi-target unified training**	**25.51**	**18.09**

Table 2: Performance of proposed methods combined with matched acoustic model back-end (WER(%))

Scheme	et05_real	
	noisy	beam
(1) no front-end	25.09	17.89
(2) FE_{enh} (enhancement target)	29.32	22.41
(3) FE_{multi} **(multi-target)**	**24.97**	**17.20**
(4) + back-end re-training	26.49	18.95
(5) **multi-target unified training**	**23.95**	**16.47**

[1] All front-end DNNs have the same number of layers (i.e. three) in this experiment for a fair comparison.

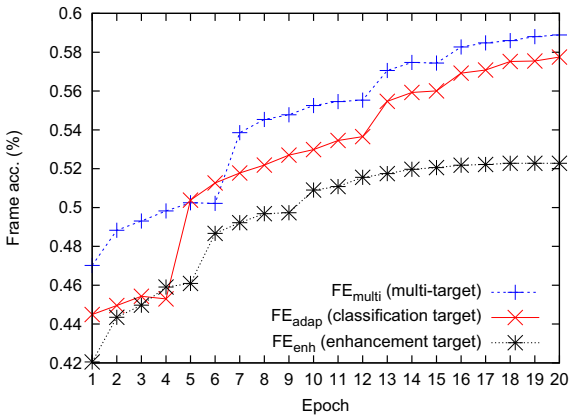

Figure 3: Frame accuracy by front-end networks on held-out set

were used for the evaluation of each method. We see that both of them substantially improve the performance of the clean back-end even after applying the beamformer.

Figure 4 shows the WERs on the development set obtained by FE_{multi} having five layers trained with different values of λ coupled with the clean back-end. Note that the case of $\lambda = 0.0$ is equivalent to FE_{enh} with five layers[2]. We see that the best performance is obtained with $\lambda = 0.5$ and understand that two targets equally contribute to improving the ASR performance. The recognition results for FE_{multi} trained with $\lambda = 0.5$ on the real evaluation set are shown in row (4) in Table 1, which are significantly better than the baseline results.

In Table 2, we show the results obtained with the matched acoustic models (multi-condition noisy model, beamformed model). We see that the matched models without any DNN front-end processing already yields slightly better results than the clean models coupled with FE_{multi}, the multi-target trained front-end DNN, confirming that the matched training data is indeed effective for acoustic modeling and should be used for baseline when available. We do not show here the results with FE_{adap}, because feature-space adaptation to the matched model does not really make sense and did not yield any improvement in fact. Significantly different tendencies from Table 1 are observed for the results with FE_{enh}. The recognition performance with the matched models is drastically degraded when combined with FE_{enh}, which clearly reveals the limitation of the DAE we mentioned in Section 2.1. In contrast, we see from the results with FE_{multi} (row (3) in Table 2) that we can prevent the problem effectively by incorporating the acoustic model target. We obtained an even better result than the matched model alone, particularly for the beamformed test data.

4.3. Effect of unified network training

We initialized a unified network by connecting a multi-target trained front-end and a back-end DNN, and re-trained it using the procedure described in Section 3.2 for both noisy and beamformed training data using clean data as a regression target for the front-end part. In the recognition time, the output of the front-end part is normalized to have zero mean and unit variance. For comparison, we also re-trained only the back-end using training data enhanced by FE_{multi}.

In rows (5) and (6) in Table 1, we show the results for

[2]Similarly, FE_{multi} with $\lambda = 1.0$ is equivalent to FE_{adap}, but the training of the adaptation DNN was not converged when the number of layers was five, and we excluded the result with $\lambda = 1.0$.

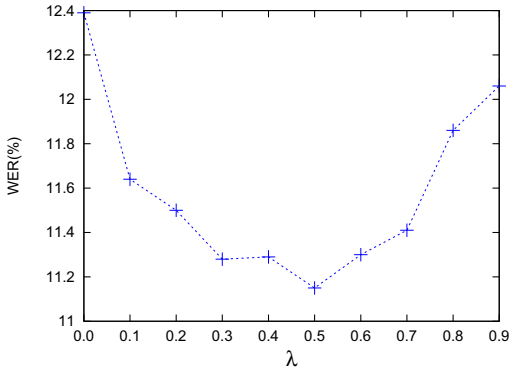

Figure 4: Effect of λ for multi-target front-end DAE training

the back-end re-training and the proposed unified training combined with the clean back-end. We used $\lambda = 0.5$ and $\gamma = 0.05$ here again for consistency. We see that the unified training yields significant improvements in both conditions, while the back-end re-training degraded the accuracy in the "beam" condition. For comparison, we conducted another unified training experiment without feeding the clean speech features to the interface layer, and in this case, the WER was increased to 22.47% from 18.09% in the "beam" condition, which clearly shows that the clean data constraint plays a critical role in the unified training.

We also show the results with the matched back-end acoustic models in Table 2. From rows (4) and (5), we see that the proposed unified training method improves the performance of the matched models, while simple back-end re-training only degrades the performance.

5. Related works

Giri et al. [35] applied multi-task learning of DNN for senone classification with a secondary task of feature enhancement. Huang et al. [36] conducted feature enhancement as the primary task in the multi-task learning. Chen et al. [37] used an LSTM-based front-end in a similar approach. While these multi-task learning framework uses a specific output branch and objective function for each task after the shared hidden layers, in this paper we have demonstrated that a network with single output can be directly trained with multiple targets. The proposed method stacks the front-end DAE and the back-end DNN with a unified objective function which leads to consistent improvement for each DNN.

6. Conclusion

In this paper, we have proposed a multi-target learning method for deep neural networks for speech enhancement and classification, and evaluated its effectiveness through the CHiME3 Challenge ASR task. The contribution of the front-end DAE to the back-end classification was improved by incorporating the secondary classification target into its training objective. Furthermore, the classification performance of the entire network was also improved by the effective use of the secondary enhancement target in the training. We are interested to see how the proposed method contributes to obtain a state-of-the-art performance when combined with 5-gram and RNN language models and fMLLR features for DNNs. We are also interested to incorporate the sequence discriminative criterion for the training of the back-end part for further performance improvement.

7. References

[1] J.S.Lim and A.V.Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proc. of IEEE*, vol. 67, pp. 1586–1604, 1979.

[2] S.Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoustics, Speech & Signal Process.*, vol. 27, no. 2, pp. 113–120, 1979.

[3] J.Droppo, L.Deng, and A.Acero, "Evaluation of the SPLICE algorithm on the aurora 2 database," in *Proc. Eurospeech*, 2001, pp. 217–220.

[4] B.Raj, T.Virtanen, S.Chaudhuri, and R.Singh, "Non-negative matrix factorization based compensation of music for automatic speech recognition," in *Proc. Interspeech*, 2010, pp. 717–720.

[5] J.T.Geiger, J.F.Gemmeke, B.Schuller, and G.Rigoll, "Investigating NMF speech enhancement for neural network based acoustic models," in *Proc. Interspeech*, 2014, pp. 1–5.

[6] 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.

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

[8] X. Feng, Y. Zhang, and J. Glass, "Speech Feature Denoising and Dereverberation via Deep Autoencoders for Noisy Reverberant Speech Recognition," in *Proc. ICASSP*, 2014, pp. 1778–1782.

[9] F. Weninger, S. Watanabe, Y. Tachioka, and B. Schuller, "Deep Reccurent De-noising Auto-encoder and Blind De-reverberation for Reverberated Speech Recognition," in *Proc. ICASSP*, 2014, pp. 4656–4660.

[10] M.Mimura, S.Sakai, and T.Kawahara, "Reverberant speech recognition combining deep neural networks and deep autoencoders augmented with a phone-class feature," *EURASIP journal on Advances in Signal Processing*, 2015.

[11] J.Heymann, R.Haeb-Umbach, P.Golik, and R.Schlueter, "Unsupervised adaptation of a denoising autoencoder by bayesian feature enhancement for reverberant ASR under mismatch conditions," in *Proc. ICASSP*, 2015, pp. 5053–5057.

[12] H.Erdogan, J.R.Hershey, S.Watanabe, and J. Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Proc. ICASSP*, 2015, pp. 708–712.

[13] T.Gao, J.Du, L-R.Dai, and C-H.Lee, "Joint training of front-end and back-end deep neural networks for robust speech recognition," in *Proc. ICASSP*, 2015, pp. 4375–4379.

[14] D.Bagchi, M.I.Mandel, Z.Wang, Y.He, A.Plummer, and E.Fosler-Lussier, "Combining spectral feature mapping and multi-channel model-based source separation for noise-robust automatic speech recognition," in *Proc. ASRU*, 2015, pp. 496–503.

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

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

[19] 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.

[20] J.P.Neto, C.Martins, and L.B.Almeida, "Speaker adaptation in a hybrid HMM-MLP recognizer," in *ICASSP*, 1996, pp. 3382–3385.

[21] V.Abrash, "Mixture input transformations for adaptation of hybrid connectionist speech recognizers," in *Eurospeech*, 1997.

[22] R.Gemello, F.Mana, S.Scanzio, P.Laface, and R.De Mori, "Linear hidden transformations for adaptation of hybrid ANN/HMM models," in *Speech Communication*, vol. 49, 2007, pp. 827–883.

[23] B.Li and K.C.Sim, "Comparison of discriminative input and output transformations for speaker adaptation in the hybrid NN/HMM systems," in *INTERSPEECH*, 2010, pp. 526–529.

[24] O.Abdel-Hamid and H.Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," in *Proc. ICASSP*, vol. 1, 2013, pp. 7942–7946.

[25] Y.Miao, H.Zhang, and F.Metze, "Towards speaker adaptive training of deep neural network acoustic models," in *INTERSPEECH*, 2014, pp. 2189–2193.

[26] J.Droppo and A.Acero, "Maximum mutual information SPLICE transform for seen and unseen conditions," in *Proc. INTERSPEECH*, 2005, pp. 989–992.

[27] D.Povey, B.Kingsbury, L.Mangu, G.Saon, H.Soltau, and G.Zweig, "Fmpe: discriminatively trained features for speech recognition," in *ICASSP*, 2005, pp. 961–964.

[28] A.Narayanan and D.L.Wang, "Joint noise adaptive training for robust automatic speech recognition," in *Proc. ICASSP*, 2014, pp. 2523–2527.

[29] J.Barker, R.Marxer, E.Vincent, and S.Watanabe, "The third chime speech separation and recognition challenge: Dataset, task and baselines," in *Proc. ASRU*, 2015.

[30] X.Anguera, C.Wooters, and J.Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Trans. Audio, Speech & Language Process.*, vol. 15, no. 7, pp. 2011–2023, 2007.

[31] V.Mnih, "Cudamat: a CUDA-based matrix class for python," in *Department of ComputerScience, University of Toronto, Tech. Rep. UTML TR*, 2009.

[32] N.Srivastava, G.Hinton, A.Krizhevsky, I.Sutskever, and R.Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[33] D.Povey, A.Ghoshal, G.Boulianne, L.Burget, O.Glembek, N.Goel, M.Hannemann, P.Motlicek, Y.Qian, P.Schwarz, J.Silovsky, G.Stemmer, and K.Vesely, "The kaldi speech recognition toolkit," in *Proc. ASRU*, 2011, pp. 1–4.

[34] M.Mimura, S.Sakai, and T.Kawahara, "Deep autoencoders augmented with phone-class feature for reverberant speech recognition," in *Proc. ICASSP*, 2015, pp. 4365–4369.

[35] 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.

[36] B. Huang, D. Ke, H. Zheng, B. Xu, Y. Xu, and K. Su, "Multi-Task Learning Deep Neural Networks for Speech Feature Denoising," in *Proc. Interspeech*, 2015, pp. 2464–2468.

[37] Z. Chen, S. Watanabe, H. Erdogan, and J. R. Hershey, "Speech Enhancement and Recognition Using Multi-Task Learning of Long Short-Term Memory Recurrent Neural Networks," in *Proc. Interspeech*, 2015, pp. 3274–3278.