EXPLORING DEEP NEURAL NETWORKS AND DEEP AUTOENCODERS IN 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

We propose an approach to reverberant speech recognition adopting deep learning in front end as well as back end of the system. At the front end, we adopt a deep autoencoder (DAE) for enhancing the speech feature parameters, and speech recognition is performed using a DNN-HMM acoustic models at the back end. The system was evaluated on simulated and real reverberant speech data sets. On average, the DNN-HMM system trained on the multi-condition training data outperformed the MLLR-adapted GMM-HMM system trained on the same data. The feature enhancement with the DAE contributed to the improvement of recognition accuracy especially in more adverse conditions. We also performed an unsupervised adaptation of the DNN-HMM models to the test data enhanced by the DAE and achieved improvements in word accuracies in all reverberation conditions of the test data.

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 in adverse conditions is still a difficult task and the recognition accuracies in adverse environments such as those with reverberation and background noise are still staying at low levels.

A key breakthrough for speech recognition technology to be accepted widely in the society will be the establishment of the methodology for easier speech interface with hands-free input. Speech reverberation adversely influences the speech recognition accuracy in such conditions and various efforts have been made to improve the recognition performance for the reverberant speech.

Reverberant speech recognition has so far been tackled by applying feature enhancement at the front end, and by attempting model adaptation and the use of more sophisticated recognition techniques. Speech enhancement techniques include deconvolution approaches that try to reconstruct clean speech by inverse-filtering the reverberant speech [1][2][3] and spectral enhancement approaches that estimate and remove the influences of the late reflection [4][5]. 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 [6]. One of the simplest approach to feature enhancement is the cepstral mean normalization (CMN) [7]. However, since reverberation time is usually larger 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) [8] that tries to adapt the acoustic model parameters to the corrupted speech.

In this paper, we take an approach to reverberant speech recognition based on deep learning, which has been drawing much attention in the speech research community. Recognition of reverberant speech is performed combining "standard" DNN-HMM [9] decoding and a feature enhancement through deep autoencoder (DAE) [10][11]. The combination of the DNN classifier and the DAE can be regarded as a single DNN classifier with a very deep structure. However, we can expect a mutually complementary effects from the combination of two networks that are optimized toward different targets. We have so far seen few practices of applying deep neural network technology to LVCSR in the adverse conditions such as reverberant and noisy speech, and this paper presents some interesting results on the effect of DNNs combined with DAEs.

2. ASR TASK IN REVERB CHALLENGE

The proposed system was evaluated following the instructions for the ASR task of the Reverb Challenge 2014 [12].

For training, we used the standard multi-condition data that is built by convolving clean WSJCAM0 data with room impulse responses (RIRs) and subsequently adding noise signals. Evaluation data consists of "SimData" and "RealData". SimData is a set of reverberant speech simulated 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. These rooms are different from those for measuring RIRs used in generating multi-condition training data. 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 with RealData, which are near ($\approx$100cm) and far ($\approx$250cm). Utterance texts for both SimData and RealData were chosen from WSJCAM0 prompts. All the reverberant speech recordings were made with eight microphones. In the experiments in this paper, however, we only use a single channel both for training and testing. The speech recognition performance is measured by word error rate in a 5k vocabluray speech recognition task.

3. DNN-HMM

Pattern recognition by neural networks has a long history [13]. In recent years, deep neural networks (DNN) has been drawing much attention again in the pattern recognition field due to the establishment of an effective pre-training methodology [14] and the dramatic im-

provement of computing power and the increase of available training data. It has also been applied to speech recognition combined with hidden Markov models (HMM) and reported to achieve significantly higher accuracy than conventional GMM (Gaussian Mixture Model)-HMM technology in various task domains [15][16][9][17].

There has so far been two typical ways to combine DNNs and HMMs. In one approach, the state emission probabilities are computed using DNNs instead of the conventional Gaussian mixture models (GMMs). In the other approach, the output from DNNs are utilized as input to conventional GMM-HMMs. The former is called the hybrid approach [16][9][17] and the latter is called the TANDEM approach [18][19][20]. In this paper, we build acoustic models adopting the hybrid approach, which has a simple structure and therefore easy to handle and has been shown to be effective in many task domains. We call these acoustic models built with hybrid approach as *DNN-HMM* hereafter in this paper.

In the training of DNNs, the standard error backpropagation training from randomly initialized states often does not yield the expected results due to the very little changes especially in the lower layer parameters caused by the repeated multiplications of the values smaller than one. Therefore, we opt to initialize the network weights in a better way by unsupervised generative training before the supervised discriminative training [14].

In the first place, each layer of the network is trained as a restricted Boltzmann machine (RBM) independently. Next, these RBMs are stacked together to constitute a deep belief network (DBN). An initial DNN is then established by adding a randomly initialized softmax layer. This DNN is trained in a supervised way through error backpropagation using HMM state IDs as labels.

It has been a standard practice to train the neural networks for DNN-HMMs independently of other components of the HMM models. The model parameters other than the DNN components are usually copied from well-trained GMM-HMMs, for example, those trained according to the minimum phone error criterion. The state labels are also usually generated by the forced alignments using those GMM-HMMs.

4. SPEECH FEATURE ENHANCEMENT BY DEEP AUTOENCODERS

The DNN structure described in the last section can be utilized as a deep autoencoder (DAE) when trained for a different target [21]. In this case, the lower layers are regarded as an encoder to obtain an efficient code and the upper layers are regarded as a decoder that "reverses" the encoder. As a whole, a DAE has a vertically symmetric network structure.

Initialization by RBM training is very important with DAEs as well. However, each of the networks in the decoder layers are initialized with the same RBM in the encoder-layer counterpart. In decoder layers, network weights are initialized as the transpose of those used for the correspondent encoder layer network and biases are initialized using visible biases from RBMs rather than hidden biases that are used for the encoder layers.

This DNN with a symmetric structure can be used as a *denoising autoencoder* when input is a corrupted data and the target is the clean data [22]. It is trained to recover the clean data from the corrupted data. Other than the input and the target, the training algorithm is the same as the ordinary DAEs [23].

5. EXPERIMENTAL EVALUATIONS

Experimental evaluations were performed for DNN-HMMs and DAEs described in the previous sections using evaluation data for Reverb Challenge [12].

In all of the experiments presented below, only single channel data was used for training and testing. For training, we used the 7,861 utterances of multi-condition data, which was also the training data for multi-condition baseline GMM-HMM models. For decoding, we used the HVite command from HTK-3.4 with a small modification to handle DNN output. The language model we used is the baseline language model supplied in the Reverb Challenge. Decoding parameters such as beam widths are set to be the same for GMM-HMM system and DNN-HMM system. Since the "likelihood" scores have different ranges, the language model weights and insertion penalties are independently optimized for each system.

The evaluation results obtained with the baseline GMM-HMM system are shown in Table 1, rows 1 through 3.

5.1. DNN-HMM

Here we describe the details of the DNN-HMM system we used for the evaluation experiments.

A 1320-dimensional feature vector consisting of eleven frames of 40-channel log Mel-scale filter bank outputs and their delta and acceleration coefficients is used as the input to the network. The targets are chosen to be the 3,113 shared states of the baseline GMM-HMMs. The six-layer network consists of five hidden layers and a softmax output layer. Each of the hidden layers consists of 2,048 nodes. The network is initialized using the RBMs trained with reverberant speech.

The fine-tuning of the DNN is performed using cross entropy as the loss function by error backpropagation supervised by state IDs for frames. The mini-batch size for the stochastic gradient descent algorithm was set to be 256. The learning rate was set to be 0.08 initially and exponentially decayed over the sequence of mini-batches. The momentum was set to be 0.9. The training was stopped after 20 epochs. The state labels for the frames were generated by the forced alignment of clean data with HVite command of HTK3.4 using the baseline GMM-HMM acoustic models trained on MFCC feature parameters of clean data. The HMM model parameters other than emission probabilities such as transition probabilities were copied from the baseline GMM-HMM models.

The word error rates for the evaluation data set obtained with the DNN-HMM system trained using multi-condition data are shown in the seventh row of Table 1. For all subsets of the "SimData" part of the evaluation set, the DNN-HMM system achieved drastically higher accuracies than the adapted GMM-HMM system. In the most adverse condition (Room 3, Far), word error rate was reduced by 15.9 points (from 39.28% to 23.34%). With the "RealData" subsets, the DNN-HMM system achieved higher accuracies than the non-adapted GMM-HMMs, and comparable accuracies with the adapted GMM-HMMs.

The DNN-HMM system was trained on the clean training set as well as the multi-condition training set. The word accuracies obtained with this clean DNN-HMM system are shown in the fourth row of Table 1. As seen in the table, the accuracies by the clean DNN-HMMs are drastically lower than the multi-condition DNN-HMMs. We see that the multi-condition training is effective for DNN-HMMs as well as GMM-HMMs from these results.

We also performed evaluation experiments on clean speech ("ClnData"). The word error rates for the clean versions of the evaluation set obtained with the baseline GMM-HMM systems are shown in rows 1 through 3 of Table 2. The results with the multi-condition DNN-HMM system is shown in the fifth row. We see that the accuracies for clean speech deteriorate significantly with the GMM-HMMs trained using multi-condition data. Meanwhile, the results obtained by DNN-HMMs trained using multi-condition data were as good as those with

Table 1. System performances on the test data (word error rate (%))

		SimData							RealData		
		Room 1		Room 2		Room 3		Ave.	Room 1		Ave.
	Proc. Scheme	Near	Far	Near	Far	Near	Far		Near	Far	
(1)	Baseline (clean, w/o CMLLR)	18.26	25.60	41.87	82.20	53.59	87.99	51.73	89.91	87.58	88.74
(2)	Baseline (multicond, w/o CMLLR)	21.28	21.18	23.12	38.83	28.24	44.77	29.56	58.96	55.60	57.28
(3)	Baseline (multicond, w CMLLR)	16.57	18.21	20.31	32.43	24.86	39.28	25.27	50.37	48.01	49.19
(4)	DNN-HMM (clean)	12.37	18.49	26.05	57.13	35.37	72.05	36.95	77.26	76.2	76.74
(5)	DAE + DNN-HMM (clean)	9.73	10.79	12.06	22.95	13.60	25.85	15.85	53.98	51.38	52.72
(6)	DAE(120) + DNN-HMM (clean)	9.32	10.71	12.46	23.04	13.24	26.09	15.84	52.89	51.38	52.16
(7)	DNN-HMM (multicond)	10.25	10.59	12.91	21.37	14.14	23.34	15.46	49.25	48.08	48.68
(8)	DAE + DNN-HMM (multicond)	14.22	14.20	13.30	19.46	14.01	18.75	15.67	45.48	45.21	45.35
(9)	DAE(120) + DNN-HMM (multicond)	14.33	14.18	13.09	19.63	15.10	19.57	15.99	45.61	44.63	45.13
(10)	DAE + DNN-HMM (multicond) + adap.	11.11	11.79	11.80	16.59	12.49	17.13	13.50	39.67	41.09	40.36
(11)	DAE + DNN-HMM (retrain)	9.74	9.98	11.80	20.69	13.50	22.81	14.77	48.42	48.85	48.63

Table 2. System performances on clean data (word error rate (%))

		ClnData			
		Room 1	Room 2	Room 3	Ave.
	Proc. Scheme				
(1)	Baseline (clean, w/o CMLLR)	13.01	12.69	12.23	12.64
(2)	Baseline (multicond, w/o CMLLR)	30.92	30.28	30.17	30.46
(3)	Baseline (multicond, w CMLLR)	16.25	15.28	15.37	15.63
(4)	DNN-HMM (clean)	7.51	7.67	7.25	7.48
(7)	DNN-HMM (multicond)	10.69	10.27	10.65	10.53

the GMM-HMMs trained using clean data.

The accuracies by the clean DNN-HMMs (Table 2, row 4) were better than the multi-condition DNN-HMMs, although the difference between them were not as large as the difference between the clean and multi-condition GMM-HMM systems.

5.1.1. Best-matched condition training

In general, multi-condition training is an effective strategy, since the run-time reverberation condition is unknown in the system development time. However, the part of the training data with mismatched reverberation conditions from the run time may cause an adverse effect on the accuracy. Therefore, we could expect a better performance if we can prepare multiple models trained with single reverberation conditions and choose a best-matched one at the run time in some way, although the choice of the best-matched model at run time itself is a non-trivial research issue.

To understand the possible effectiveness of this approach, we trained DNNs with simulated training data that would match the "RealData" part of the evaluation data. We generated two simulated training data sets using the RIR for "Near" and "Far" microphone distances in "Large" room. Each of the resulting training sets has the same size as the whole multi-condition training set. The experimental results are shown in Table 3.

As seen in the table, the word accuracy for "RealData"-"Near" as well as "RealData"-"Far" is improved with the "Large"-"Far" model. However, the accuracy for "RealData"-"Near" is degraded with the "Large"-"Near" model. Although both labeled "Near", the microphone distances of 50cm in "Large"- "Near" (training) and 100cm in "RealData"-"Near" (test) seem to have made a big difference in the reverberated speech.

From these preliminary experimental results, we see that the recognition performance may, at times, improve with "the collection of single condition models" approach, when there happens to be a single-condition model that matches the run-time condition very well. On the other hand, the selected single condition model can yield accuracies much worse than multi-condition models when the match of the conditions is not good enough.

Table 3. Performances of single condition DNN-HMMs on RealData (word error rate (%))

		RealData	
		Room 1	
	Proc. Scheme	Near	Far
(7)	DNN-HMM (multicond)	49.25	48.08
(12)	DNN-HMM (Large Near)	55.06	51.82
(13)	DNN-HMM (Large Far)	46.60	44.77

5.2. Denoising deep autoencoder

The input and the target for the denoising autoencoder (DAE) were set to be the eleven-frame sequence of 40-channel log Mel-scale filterbank features with their delta and acceleration parameters. The DAE is fine-tuned using reverberant speech as the input and clean speech as the target. The input frames and the output frames for the training were adjusted to be time aligned in the multi-condition training data generation process. The last portions of reverberant speech utterance files exceeding the length of the clean speech were trimmed to equalize the lengths of input and output.

The autoencoder network has six layers in total consisting of three encoding layers and three decoding layers. The number of nodes in each layer is set to be 2,048 except for input and output layers. The network is initialized using the same RBMs as used for initializing the DNNs described in the last subsection which were trained using reverberant speech. The encoding layers were initialized using the weights of first three RBMs and the hidden unit biases. The decoding layers were initialized using the transpose of the weights mentioned above and the visible unit biases.

The fine tuning of the DAE was performed by error backpropagation with squared error as the loss function. The parameters such as the mini-batch size and the momentum are set to be the same as those for DNN training. However, initial learning rate was set to be 0.001, which is smaller than the one for DNN training.

The evaluation results with the combination of the DAE and the clean DNN-HMM is shown in Table 1, row 5. The accuracies are drastically improved in all conditions from the clean DNN-HMM without DAE (row 4 of the same table) and we understand that the DAE has done an effective feature enhancement as expected. Interestingly, these results are comparable to those from the multi-condition DNN-HMM without DAE (Table 1, row 7) and slightly better in some conditions.

The results with the combination of the DAE and the multi-condition DNN-HMM are shown in Table 1, row 8. Although we see degradations in the word accuracy with the conditions "SimData"-"Room 1" (a small room) and "Room 2" (a middle-sized room), "Near" from the multi-condition DNN-HMM without the DAE, the accuracies are improved with all other conditions, especially drastically in more adverse conditions such as "Far" microphone conditions of "Room 2" (middle-sized) and "Room 3" (large) as well as "RealData" conditions. The accuracies by the combined DAE and multi-condition DNN-HMM system turned out to be higher than the MLLR-adapted GMM-HMM system on average including "RealData" condition.

The speech feature parameters "enhanced" by the DAE may have different characteristics from the original reverberant speech.

Therefore, we retrained the DNN using the DAE output and performed speech recognition experiments. This time, the RBMs for initializing the network were trained using the DAE output as training data.

The word error rates obtained using this retrained network are shown in Table 1, row 11. We see that the deterioration of the word accuracies for "SimData", "Room 1" is ameliorated but improvements of the accuracies are not seen mostly in other conditions.

Overall, retraining of the DNN using the DAE-enhanced data was not effective and the combination of the DAE and the DNN trained using multi-condition data was more robust for severely reverberant speech.

5.2.1. Autoencoder target options

In the experiment above, the DAE was trained with the 11 frames (the center frame and the five frames before and after it) of filterbank-based feature parameters as target. However, there may not be enough information in the input to enhance the left frames, especially in the long reverberation time conditions. Therefore, we also trained the DAE with only the center frame as target. In this experiment, we trained the DAE with the 11 frames of feature parameters (1320 dimensions in all) as input and the center frame feature parameters (120 dimensions) as target. The enhanced frames are concatenated to constitute 11-frame, 1320-dimensional feature parameters to be input to the DNN-HMM. The evaluation results of this version of DAE combined with the clean DNN-HMM and the multi-condition DNN-HMM are shown in rows 6 and 9 in Table 1. When combined with the clean DNN-HMM, the accuracies got a slightly better on average compared with the DAE that outputs 11 frames of feature parameters (Table 1, row 5), but the differences were small. On the other hand, when combined with the multi-condition DNN-HMM that can handle reverberated speech, no particular improvements were seen (row 9 vs. row 8).

5.2.2. Unsupervised adaptation of DNN-HMM using enhanced test data

In order to alleviate the mismatch between the test data enhanced by the DAE and the DNN-HMM, we explored a way of "adapting" the DNN-HMM to the test condition. Adaptation of DNN-HMM models is a topic of ongoing research efforts and statistical techniques such as MLLR and MAP adaptations [8, 24] for GMM-HMM models are not established yet. However, it has been empirically known that the effect similar to model adaptation can be obtained simply by additional backpropagation training using the test data [25].

We attempted an "unsupervised adaptation" of the DNN-HMM by ten epochs of additional backpropagation training using test data. For the purpose of fair comparison with the baseline MLLR-adapted GMM-HMM system that used the all utternaces within one test condition, which is a combination of room size and microphone distance [12], we also used all the utterances within a common test condition. The labels for supervision in backpropagation training were generated from the recognition results using non-adapted DAE + DNN-HMM system. The learning rate was set to be rather small value of 0.001.

The results of these adaptation experiments are shown in Table 1, row 10. It is seen that the word accuracies are improved in all conditions. Looking at the "RealData" part, we see that the errors are reduced by 10.7 points with "Near" and 6.9 points with "Far" compared with the MLLR-adapted GMM-HMM system (row 3).

6. CONCLUSION

In this paper, we proposed an approach to reverberant speech recognition adopting deep learning in front end as well as back end of the system and evaluated it through the ASR task (one channel) of Reverb Challenge 2014.

The DNN-HMM system trained on the multi-condition training set achieved a conspicuously higher word accuracy on average compared with the MLLR-adapted GMM-HMM system trained on the same data. Furthermore, feature enhancement with the DAE contributed to the improvement of recognition accuracy especially in the more adverse conditions. When the DNN-HMM was used without the DAE front end on "RealData", it resulted in a comparable performance with the adapted GMM-HMM system. However, it clearly outperformed the adapted GMM-HMM system when combined with the DAE. We also performed an unsupervised adaptation of the DNN-HMM models to the test data enhanced by the DAE and achieved further improvements in word accuracies in all reverberation conditions of the test data.

In this work, the DAE was initialized using the same set of RBMs as used for the DNN-HMM initialization. The input and output of the DAE was also defined to be the same set of feature parameters as the input for DNN-HMM. However, the network structure and the feature parameters for DAE may be optimized in some criterion to yield better results and we are looking at these issues as future work.

7. REFERENCES

[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

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

[12] 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.

[13] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[14] G.E.Hinton, S.Osindero, and Y.Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, pp. 1527–1554, 2006.

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

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

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

[22] 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.

[23] 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.

[24] J.Gauvain and C-H.Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. Speech & Audio Process.*, vol. 2, pp. 291–298, 1994.

[25] Y.Xiao, Z.Zhang, S.Cai, J.Pan, and Y.Yan, "A initial attempt on task-specific adaptation for deep neural network besed large vocabulary continuous speech recognition," in *Proc. INTERSPEECH*, 2012.