MULTI SCALE FEEDBACK CONNECTION FOR NOISE ROBUST ACOUSTIC MODELING

Dung T. Tran, Ken-ichi Iso, Motoi Omachi, Yuya Fujita

Yahoo Japan Corporation

ABSTRACT

Simply feeding of a last hidden layer of the deep neural network (DNN) back to the input layer recently found to be effective for noise robust acoustic modeling. Such high level feature strengthens the robustness of DNN based acoustic model while paying approximately twice the computational cost. In this paper, we proposed to feed such high level feature iteratively back to lower layers, which is referred as multi-scale feedback connection. With this intention, we firstly extract the high level feature at the last hidden layer of DNN. Second, this high level feature feed back to a lower scale features, they then generates a subsequent prediction as well as a subsequent high level feature. This subsequent high level feature is further feed down to a lower layers. We evaluated the proposed approach on both TIMIT and a large scale internal dataset. The large scale internal dataset includes voice search and far field dataset. Our finding is two aspects. First, at equivalent computational costs, the multi-scale feedback connection outperforms the DNN, the DNN with skip connection and the DNN with feedback connection. The improvement is larger on the far field dataset. Second, pair layers-wise pretraining helps the proposed approach to converge better.

Index Terms— noise robust ASR, feedback connection, robust feature extraction, acoustic modeling.

1. INTRODUCTION

Increasing the robustness of automatic speech recognition in daily environments has attracted a lot of attention recently [1, 2, 3, 4, 5, 6]. While other advanced network architectures including convolutional neural network (CNN) [7, 8] and Long Short Term Memory network (LSTM) [9, 10, 11, 12] have dominated, simple DNN is prefered for acoustic modeling tasks in real products due to its simplicity. In DNNs, the information flows in only one forward direction, from the input layer, through the hidden layers to the output layer. There are no cycles or loops in the network. At the output layer, the final prediction is made whichs based on only the representation at the last hidden layer. Inspite of its simplicity, DNN's performance is still competitive to other models. In the 200 hours clean dataset, [9] reported DNN degraded less than 3% relative WER compared to LSTM and CNN models. Recently, [13, 14] showed that simple DNN with careful tuning can obtain a competitive result compared to other sophisticated deep networks for noisy datasets.

Not only feed-forward network without cycles or loops is fairly competitive to LSTM or CNN but introducing few additional special connections resulting in combining of features at different layers is often very helpful. Skip connection [15] is a type of combining a lower scale feature at a lower layer to a higher scale feature at a higher layer with the original motivation of mitigating the vanishing gradient problem. Such approach is widely used in the computer vision community and can be found in some studies recently [9, 16, 17, 18, 19].

In the contrast with skip connection, feedback connection based approaches [20, 21] have two subnetworks making two corresponding predictions to the same target. The unfolded form of the feedback connection is depicted in Fig. 1. One subnetwork is only used to extract the high level feature and the other one is used to generate the subsequent prediction based on the same input feature and the high level feature. The high level feature often is extracted at the output of the last hidden layer of the first subnetwork. Different to the skip connection, the feedback connection approaches introduce explicitly multi-objective functions. Two subnetworks are jointly trained. Initially, [22, 20] first proposed to use the feedback connection concept for acoustic modeling task. In [20], each subnetwork has two small networks: one is prediction and other one is correction. The feedback connection is formed by feeding the output of the hidden layer of the correction network to the input of the prediction network. In addition, [21] proposed to share the parameters of two subnetworks and show an improvement on CHiME3 dataset [23]. Although sharing parameters can be made to reduce the number of parameters, it still requires a forward pass two times at inference which turns out to be inefficient.

More recently, [24, 25] proposed to feed such high level feature before the input layer in order to control the filter coefficients of beamforming specialized for far-field speech recognition. More precisely, the hidden units in the deep LSTM acoustic model are used to assist in predicting the beamforming filter coefficients. Instead of using the high level feature from the same time step as [20, 21], the authors in [24, 25] extract it from the previous time step.

Our concern is two aspects. Firstly, since the feedback connection depends on the high level feature's quality, therefore we focus on improving the high level feature's quality. We argue that when the high quality of high level feature is obtained, using a large network for the subsequent predictions is unnecessary therefore we might simplify the subsequent networks. Secondly, to fully exploit the feedback connection at all scales, we feed the high level feature not only to the input layer but iteratively back to lower layers. At the first step, the last hidden layer of the DNN which generates the first prediction is considered as the high level feature. This high level feature is then fed back to a lower scale feature which then generates the subsequent prediction as well as the subsequent high level feature. This subsequent high level feature is further feed down to a lower layers. All predictions share the same target. To simplify, the subsequent subnetwork is replaced by a softmax layer only.

The rest of the paper is organized as follows: We describe the simple feedback connection and our proposed approach in Section 2. In Section 3, we evaluate the proposed approach in various experiments with different datasets. Related work is discussed in the Section 4.

2. MULTI-SCALE FEEDBACK CONNECTION

2.1. Simple feedback connection

Although, feedback connection can be applied to CNN or recurrent neural network, in the scope of this work, we only present DNN case. The simple feedback connection has two subnetworks and is described in Fig. 1. While the first subnetwork is used to extract the high level feature given the in-

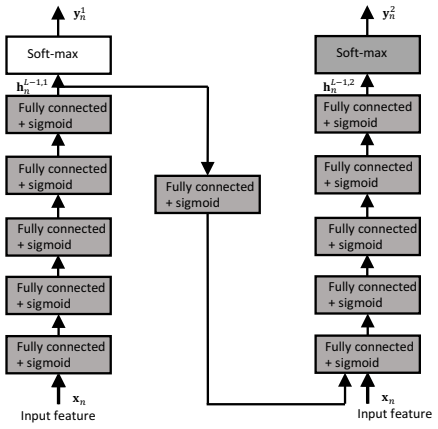

Fig. 1. *Illustration of the simple feedback connection. All layers are updated at training. Only the layers (gray color) are used at inference.*

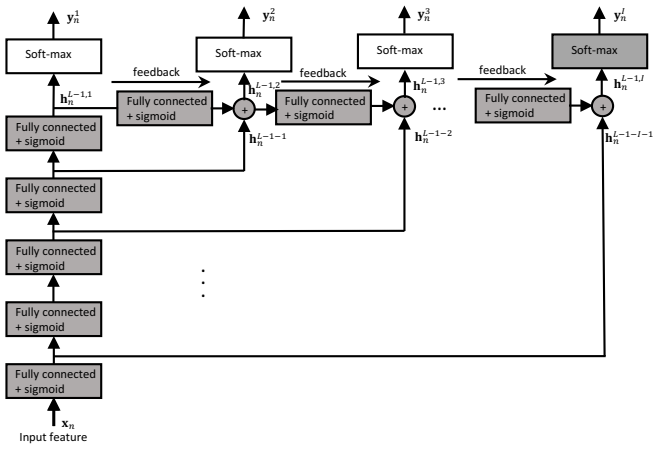

Fig. 2. *Illustration of the multiscale feedback connection. All layers are updated at training. Only the layers (gray color) are use at inference.*

put feature, the subsequent subnetwork simply appends the high level feature to the same input feature to generate the subsequent prediction. As the high level feature has rich information, it does help to guide the input feature to make the final decision. The network computation can be described as follows

$$\mathbf{h}_n^{L-1,1} = \text{DNN}(\mathbf{x}_n, \mathbb{W}_1) \quad (1)$$

$$\mathbf{y}_n^1 = softmax(\mathbf{W}\mathbf{h}_n^{L-1,1} + \mathbf{b}) \quad (2)$$

$$\mathbf{h}_n^{L-1,2} = \text{DNN}(\mathbf{x}_n, \sigma(\mathbf{W}^1\mathbf{h}_n^{L-1,1} + \mathbf{b}^1), \mathbb{W}_2) \quad (3)$$

$$\mathbf{y}_n^2 = softmax(\mathbf{W}\mathbf{h}_n^{L-1,2} + \mathbf{b}) \quad (4)$$

$$J = \sum_n (J_1(\mathbf{y}_n^1, t_n) + J_2(\mathbf{y}_n^2, t_n)) \quad (5)$$

where x_n denotes the input features within the context frames; n is the central frame index; $\mathbb{W}_1, \mathbb{W}_2$ represents the parameter of the first and subsequent subnetwork (excluding the last layers); $\sigma(.)$ denote the nonlinearity activation function. $\mathbf{h}_n^{L-1,1}, \mathbf{h}_n^{L-1,2}$ are the last hidden layers of the first and subsequent subnetwork, respectively; $\mathbf{h}_n^{L-1,1}$ can also be refered to the high level feature; $\mathbf{y}_n^1, \mathbf{y}_n^2$ represent the first and the subsequent prediction, respectively; $\mathbf{W}, \mathbf{b}$ are the weight and bias of the last layer of each subnetwork; $\mathbf{W}^1$,$\mathbf{b}^1$ denotes the weight and bias of the feedback connection. The overall objective function is the sum of individual objective functions. t_n is the state posterior probability; J_1, J_2 denotes the cross-entropy function. Sharing parameters to make $\mathbb{W}_1$ identical to $\mathbb{W}_2$ tends to give better performance on the noisy data [21]. However, at the inference, the computation cost is almost two times compared to the conventional DNN. In addition, the simple feedback connection exploit the high level feature $\mathbf{h}_n^{L-1,1}$ at single scale input feature only.

2.2. Proposed approach

The goal is to efficiently exploit the high level feature $\mathbf{h}_n^{L-1,1}$ not only at single scale feature (input layer) but also at multiscale features(all layers). The network architecture of the proposed approach is shown in Fig. 2. The network consecutively generates predictions to match the same target. After each prediction, the high level feature is extracted from the last hidden layer. The high level feature is then fed back to a lower scale feature forming a new pair of features to generate the subsequent prediction. The network computation can be described as follows

$$\mathbf{h}_n^{L-1,1} = \text{DNN}(\mathbf{x}_n, \mathbb{W}) \quad (6)$$

$$\mathbf{y}_n^1 = softmax(\mathbf{W}\mathbf{h}_n^{L-1,1} + \mathbf{b}) \quad (7)$$

$$\mathbf{h}_n^{L-1,i+1} = \sigma(\mathbf{W}^i\mathbf{h}_n^{L-1,i} + \mathbf{b}^i) + \mathbf{h}_n^{L-1-i} \quad (8)$$

$$\mathbf{y}_n^{i+1} = softmax(\mathbf{W}\mathbf{h}_n^{L-1,i+1} + \mathbf{b}) \quad (9)$$

where Eq. 6 and Eq. 7 describe the DNN network with L layers; $\mathbf{h}_n^{L-1,1}$ denotes the high level feature which is extracted from the first prediction $\mathbf{y}_n^1$. $\mathbf{h}_n^{L-1,i+1}$ represent the high level feature at the step $i+1$. $\mathbf{h}_n^{L-1-i}$ denotes the hidden layer of DNN.$\mathbf{W}^i, \mathbf{b}^i$ denote the weight matrix and bias of the feedback connection. σ is a nonlinearity activation function. $\mathbf{y}_n^{i+1}$ represent the i^{th} subsequent prediction. There are I prediction in total. Note that, while the high level feature requires a fully connected layer, the lower scale feature does not. Different to than simple feedback connection, the large network for generating the subsequent predictions are replaced by a simple softmax layer which reduces dramaticly the computation cost. To make the network's parameter comparable with that of DNN, we reduce the number of neurons of each layer. We did the preliminary experiment by minimizing the total objection function $J = \sum_n \sum_{i=1}^{I} J_i(\mathbf{y}_n^i, t_n)$ but we failed to make it converge to a better local optimum. In fact, minimizing the total objective function gives almost the same performance as a simple DNN. Inherited from the simple feedback connection, we first pre-train the first pair of objective function (ex. $J = \sum_n \sum_{i=1}^{2} J_i(\mathbf{y}_n^i, t_n)$). After this pre-training is finished, we use the updated parameters of the network as the initialization for training of the next pair (eg. $J = \sum_n \sum_{i=2}^{3} (J_i(\mathbf{y}_n^i, t_n))$). This training process is repeated for a few steps until the final prediction. In the inference, only the final prediction $\mathbf{y}_n^I$ is used; the rest are not used.

3. EXPERIMENT

3.1. Dataset

We conduct the experiments on various dataset. They are TIMIT task and internal dataset. The large scale internal dataset includes voice search and far field dataset.

3.1.1. TIMIT

The experiments on TIMIT are based on a phoneme recognition task (aligned with the Kaldi s5 recipe[1]). The features considered in this work are standard 39 dimension Mel-Cepstral Coefficients (MFCCs) computed every 10 ms with a frame length of 25 ms. The context window is 11 frames with left and right context frames of 5 frames. The networks are fed by such 11 frames to predict monophone targets at their output. A bi-gram language model is used. The implementation uses Keras[2]. Optimization uses stochastic gradient descent method with 0.5 momentum. We found 0.5 momentum works best in this experiments. Learning rate of 0.1 is set for 5 iterations at the begining. When the validation loss reduces by less than 0.002 between successive iterations, learning rate is halved. The minibatch size is 256. The code for TIMIT will be available[3].

3.1.2. YJVOICE voice search task

The speech signal was sampled at 16kHz sampling. The 40 dimensional Mel filterbank feature was compute using 25msec window with 10msec frame interval. The input feature has 11 frames context with 5 left frames and 5 right frames. The number of DNN outputs is 4003 corresponding to the number of triphone states obtained by decision tree clustering of 3-state triphones. The training data consists of 177 hours which is approximately 300k utterances sampled from service log data and has 63.8M frames. The validation data consists of 10k utterances and is not included in the above 300k utterances. Language model has a vocabulary size of 1M words and a trigram WFSTs is used. The language model is a tri-gram model trained using text queries of the Yahoo Japan Web search engine and transcriptions of mobile voice search queries. Our decoder is an internally developed single-pass WFST decoder [26]. Evaluation data was sampled from real services (Voice Search and Voice Dialog). Each set of evaluation data was sampled from voice search at 4 different periods. They are denoted as set1, set2, set3 and set4, respectively.

3.1.3. YJVOICE noisy task

The acoustic model was trained on augmented data. For generating noisy data, fan noise, tap noise and microwave noise, were collected in the kitchen and these are added to the original audio. For generating noisy reverberated data, noises of speech and music were convolved with simulated impulse responses and added to the original audio. The total training data is 180 hours. The room impulse response was generated using a room simulator [27]. The rooms were configured with different widths, lengths and heights. The room's width and room's length were sampled from a uniform distribution of between 3 and 15 meters. The room's height was sampled from an uniform distribution between of 2 to 3 meters. The speaker positions were generated randomly in the room but the speaker's heights were chosen randomly in the range between 1 to 2 m. The microphone positions were also sampled randomly in the room but the microphone's heights were constrained between 0.4 meter and 2 meter. The total number of room impulse responses is 19800. The evaluation data consist of 3180 utterances recorded in the kitchen with a microphone array which has 8 microphones. The distance from the speaker to microphone is around 1 meter.

[1]https://github.com/kaldi-asr/kaldi/tree/master/egs/timit/s5

[2]https://keras.io

[3]https://github.com/dzungtran32/MultiscaleFeedback

3.2. Results

3.2.1. TIMIT

The DNN baseline has 7 layers and each has 1024 nodes. A Rectified Linear Units (ReLU) activation function is used. The DNN baseline with skip connection was built based on the 7 layers DNN by adding the output of the input layer (after the non-lineariry) to the input of the last layer (after the non-linearity). Our multiscale feedback connection approach based on DNN baselines which has a 3 steps feedback connection where the high level feature is fed to 5^{th},3^{rd} and 1^{st} layer sequentially. To reduce the parameters used, we use only 848 neurons for each layer. It is denoted as DNN+MulFeedback in Table 1. We also conducted experiment with simple feedback connection (DNN+Feedback) with two options: With small option (S), each fully connected layer has only 848 neurons. With large option (L), each fully connected layer has 1024 neurons. We also conducted experiments with a modification of the simple feedback connection. In this network, all scale features are concatenated to form a big input feature before feeding to the subsequent subnetwork. We used only one layer for the second subnetwork. We refer to this as shallow feedback connection (DNN+shFeedback). We report shallow feedback connection's performance with both small (S) (each layers has 848 neurons) and large network (L) (each layer has 1024 neurons). To make a fair comparision, all models use the same context window of 11 frames. Results on TIMIT are show in Table 1 : Our baseline DNN has 78.4% accuracy. The skip connection (DNN + skip) reduce the performance by 0.5% absolute accuracy. Simple feedback connection (DNN + Feedback(L)) gives 1% absolute improvement while almost doubling the computation cost. Reducing the number of neurons in each layer to 848, the simple feedback connection (DNN + Feedback(S)) does not bring any improvement compared to the simple DNN. With the same computation cost, concatenating all scale features with simple feedback connection (DNN+shFeedback(S)) does not help. If the large network is used, this (DNN + shFeedback(L)) gives 1% WER absolute compared to DNN. Multiscale feedback connection (DNN + MulFeedback) outperforms other baselines and gives 1.6% absolute improvement over simple DNN while keeping the number of parameter unchanged. We concluded that, with the same computation cost, the multiscale feedback connection performs best. Both methods (DNN + MulFeedback) and (DNN + shFeedback(S)) use all scale features extracted from DNN, however, the (DNN + MulFeedback) performs better than (DNN + shFeedback(S)). We argue that combining all scale features with single step high level feature might not fully exploiting feedback connection since this feature might be less informative. In Table 2 we show that after pretraining, the accuracy of the new prediction is increased showing the important of pretraining. The result with only a single feedback connection is already better than simple DNN. Althought the proposed approach is far from reaching the state of the art on TIMIT dataset [28] in which author requires additional modifications, we would like to convince the effectiveness of the multi-scale feedback connection in the combination with DNN and would like to leave the further investigation for LSTM model in future work.

Table 1. *Phone accuracy for TIMIT experiment on the development and evaluation sets*

Model	dev	eval	# param
DNN 7x1024	79.8	78.4	7.3M
DNN+skip	78.6	77.9	7.3M
DNN+Feedback(S)	79.5	78.2	10.0M
DNN+Feedback(L)	80.1	79.4	14.7M
DNN+shFeedback (S)	79.9	78.7	9.3M
DNN+shFeedback (L)	80.0	79.5	13.6M
DNN+MulFeedback	**81.0**	**80.0**	7.2M

Table 2. *Phone accuracy for TIMIT experiment on the development and evaluation sets with different number of feedback connection with DNN*

Model	dev	eval
1	80.2	79.3
2	80.5	79.6
3	**81.0**	**80.0**

3.2.2. YJVOICE voice search

Our acoustic model based neural network is trained using Tensorflow[4]. DNN baseline has 5 layers and each layer has 1024 neurons. The sigmoid activation function is used. All configurations for DNN with skip connection and DNN with feedback connection are similar to Section 3.2.1. The multiscale feedback connection has a 4 step feedback connection where the high level feature is fed to 4^{th},3^{rd},2^{nd},1^{st} layer sequentially .The number of parameter used for DNN, DNN + skip, DNN + Feedback(S),DNN + Feedback(L), DNN + shFeedback(S), DNN + shFeedback(L), DNN + MultiFeedback are 9.3M, 9.3M, 10.5M,14.6M, 17.1M, 21.6M and 9.3M respectively. The Table 3 shows the results for each model. Compared to other baselines, multiscale feedback connection approach performs consistently better than the DNN baseline on all evaluation sets and it brings 3% relative WER reduction compared to DNN baseline while keeping the number of parameter unchanged. Our skip connection+DNN failed to give an improvement compared to the DNN case. Using more parameters, simple feedback connection performs consistently better than DNN but with only 2% relative WER reduction. The shallow feedback connection gives no gain compared to the DNN baseline even though this model use many more parameters. These results suggest that, for the simple feedback connection approaches, second subnetwork should be a deeper network rather than a shallow network.

[4]https://www.tensorflow.org

Table 3. *Accuracy for noisy YJVOICE voice search experiment on different sets*

Model	set 1	set2	set3	set4	Ave
DNN	84.37	83.88	90.40	84.86	85.87
DNN+Skip	80.05	78.42	88.07	79.84	81.59
DNN+Feedback(S)	84.14	83.82	90.41	84.75	85.78
DNN+Feedback(L)	84.64	84.18	90.66	85.19	86.16
DNN+shFeedback(S)	84.34	83.75	90.10	84.85	85.76
DNN+shFeedback(L)	84.53	83.48	90.56	84.91	85.87
DNN+Multi-scale	84.97	84.15	90.87	85.24	**86.30**

Table 4. *Accuracy for YJVOICE noisy data experiment.*

Model	Word Acc	Sentence Acc
DNN	60.78	31.35
DNN+skip	59.60	30.09
DNN+Feedback (S)	61.54	31.30
DNN+Feedback (L)	64.09	33.58
DNN+shFeedback (S)	61.02	31.06
DNN+shFeedback (L)	62.29	32.24
DNN+Multi-scale	**64.12**	**34.18**

3.2.3. YJVOICE far field data

We conducted the experiment on the far field data. A similar trend is observed on Table 4. These results are significant lower than those results in the Table 3. Again, the skip connection did not give an improvement compared to the simple DNN. The multiscale feedback connection gives a significant improvement over the simple DNN. More precisely, the multiscale feedback connection gives 8% relative WER reduction compared to the simple DNN. The improvements is much higher in the clean condition. This suggest the multiscale feedback connection is more suitable for far field data. The multiscale feedback connection gives similar performance compared to simple feedback connection (DNN+Feedback(L)) while significantly reducing number of parameters used, similar to that of the simple DNN. Similar to Section 3.2.1, (DNN + MulFeedback) and (DNN + shFeedback(S)) use all scale features extracted from DNN, however, the (DNN + MulFeedback) performs better than (DNN + shFeedback(S)).

4. RELATION TO PRIOR WORK

Our proposed approach is most related to related to [20, 21]. Compared to other methods, our proposed approach applied feedback connection to all scale features while the rest applied for only the input layer. We confirmed that the proposed method outperform other methods while keeping number of parameters unchanges reavealing multi-scale exploits the high level feature effectively. In addition, our proposed approach is two times faster than [21].

In other words, [24, 25] proposed to extract the high level feature from the previous time step and therefore no need to perform inference two times. While [24, 25] feed the high level feature before the input layer only, our proposed approach introducing feedback connection at all layers.

5. CONCLUSION

We present a multiscale feedback connection for noise robust acoustic modeling. We conducted experiment for DNN based acoustic model and found the proposed approach consitently outperform skip connection and simple feedback connection on different dataset including both clean and far field dataset. We confirmed the proposed approach works better on the far field data reaveal that multiscale feedback connection is more robust to noise. With DNN based acoustic model, on the large scale far field dataset, the proposed approach can obtained 8% relative WER reduction compared to other DNN baseline while keeping the number of parameters unchanged. This contribution could be applied to our speech engine without increasing the system's computational cost. In the future, we will investige multiscale feedback connection for other advance models such as CNN or LSTM [24, 25]. In addition, the number of feedback connection steps could be also further investigated in the future work.

6. REFERENCES

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

[2] V. Mitra, H. Franco, C. Bartels, J. van Hout, M. Graciarena, and D. Vergyri, "Speech recognition in unseen and noisy channel conditions," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 5215–5219.

[3] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, April 2014.

[4] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition." *Computer Speech and Language*, vol. 46, pp. 535–557, Jul. 2017.

[5] J. Li, Y. Huang, and Y. Gong, "Improved cepstra minimum-mean-square-error noise reduction algorithm for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 4865–4869.

[6] Y. Qian, T. Tan, and D. Yu, "An investigation into using parallel data for far-field speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 5725–5729.

[7] T. N. Sainath, B. Kingsbury, G. Saon, H. Soltau, A. Mohamed, G. Dahl, and B. Ramabhadran, "Deep convolutional neural networks for large-scale speech tasks," *Neural Networks*, vol. 64, pp. 39 – 48, 2015, special Issue on Deep Learning of Representations.

[8] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural network for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 1533 – 1545, 2014.

[9] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *ICASSP*, April 2015, pp. 4580–4584.

[10] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *INTERSPEECH*, 2014, pp. 338–342.

[11] Y. Zhang, G. Chen, D. Yu, K. Yao, S. Khudanpur, and J. R. Glass, "Highway long short-term memory RNNS for distant speech recognition," in *ICASSP*, 2016, pp. 5755–5759.

[12] S. Zhang, C. Liu, H. Jiang, S. Wei, L. Dai, and Y. Hu, "Feedforward sequential memory networks: A new structure to learn long term dependency," in *arXiv preprint arXiv:1512.08301v2*, 2016.

[13] Y. Tachioka, S. Watanabe, and T. Hori, "The MELCO/MERL system combination approach for the fourth CHiME challenge," in *The 4th International Workshop on Speech Processing in Everyday Environments*, 2016.

[14] T. H. Dat, N. Terence, S. Sivadas, L. T. Tuan, and T. A. Dung, "The i2r system for chime-4 challenge," in *The 4th International Workshop on Speech Processing in Everyday Environments*, 2016.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *arXiv prepring arXiv:1506.01497*, 2015.

[16] J. Heymann, L. Drude, and R. Haeb-Umbach, "Wide residual blstm network with discriminative speaker adaptation for robust speech recognition," in *The 4th International Workshop on Speech Processing in Everyday Environments , San Francisco, September, 2016*, 2016.

[17] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "A network of deep neural networks for distant speech recognition," in *ICASSP*, Mar 2017, pp. 4880–4884.

[18] Z. Chen, Y. Huang, J. Li, and Y. Gong, "Improving mask learning based speech enhancement system with restoration layers and residual connection," in *INTERSPEECH*, August 2017, pp. 3632–3636.

[19] Y. Zhang, W. Chan, and N. Jaitly, "Very deep convolutional networks for end-to-end speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 4845–4849.

[20] Y. Zhang, D. Yu, M. L. Seltzer, and J. Droppo, "Speech recognition with prediction-adaptation-correction recurrent neural networks," in *ICASSP)*, 2015, pp. 5004–5008.

[21] D. T. Tran, M. Delcroix, A. Ogawa, C. Hummer, and T. Nakatani, "Feedback connection for deep neural network-based acoustic modeling," in *ICASSP*, Mar 2017, pp. 5240–5244.

[22] G. S. V. S. Sivaram, S. K. Nemala, N. Mesgarani, and H. Hermansky, "Data-driven and feedback based spectro-temporal features for speech recognition," *IEEE Signal Processing Letters*, vol. 17, no. 11, pp. 957–960, Nov 2010.

[23] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third CHiME speech separation and recognition challenge: Dataset, task and baselines," in *ASRU*, Dec 2015, pp. 504–511.

[24] B. Li, T. N. Sainath, R. J. Weiss, K. W. Wilson, and M. Bacchiani, "Neural network adaptive beamforming for robust multichannel speech recognition," in *INTERSPEECH*, 2016.

[25] Z. Meng, S. Watanabe, J. R. Hershey, and H. Erdogan, "Deep long short-term memory adaptive beamforming networks for multichannel robust speech recognition," in *ICASSP*, 2017, pp. 271–275.

[26] K. I. Iso, E. Whittaker, T. Emori, and J. Miyake, "Improvements in japanese voice search." in *INTERSPEECH*, 2012, pp. 2109–2112.

[27] J. Allen and D. Berkley, "Image method for efficiently simulating small-room acoustics." *Journal Acoustic Society of America*, vol. 65, pp. 943–946, 1979.

[28] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Improving speech recognition by revising gated recurrent units." in *INTERSPEECH*, 2017.