COMPARATIVE EVALUATIONS OF VARIOUS FACTORED DEEP CONVOLUTIONAL RNN ARCHITECTURES FOR NOISE ROBUST SPEECH RECOGNITION

Masakiyo Fujimoto and Hisashi Kawai

National Institute of Information and Communications Technology, Japan

{masakiyo.fujimoto, hisashi.kawai}@nict.go.jp

ABSTRACT

In this paper, we present a factored network-based acoustic modeling framework with various deep convolutional recurrent neural network (RNN) architectures for noise-robust automatic speech recognition (ASR). As the factored network-based acoustic model, we have already proposed a deep convolutional neural network (CNN)-based framework. Deep CNNs can emphasize the spatial locality of input speech features, but have no ability to analyze the properties of long-term speech feature sequences. Therefore, we introduce various deep convolutional RNN architectures that achieve both spatial locality and long-term analysis into our proposed factored network-based acoustic modeling framework. Through various comparative evaluations, we reveal that the proposed method successfully improves the accuracy of ASR in noisy environments.

Index Terms— noise robust speech recognition, factored network, deep convolutional RNN architecture, multi-channel input

1. INTRODUCTION

Ensuring robustness to noise in our daily environment is an increasingly crucial problem for the practical use of automatic speech recognition (ASR). As speech applications on mobile and home devices continue to proliferate, the noise robustness of ASR should be improved as far as possible. Although the simplest way to ensure noise robustness is the front-end processing of ASR, including speech or feature enhancement, this can lead to serious performance degradation because of the signal distortion in recent deep neural network (DNN)-based ASR frameworks. This problem is especially prominent in single-channel processing, which includes traditional techniques [1, 2], a denoising autoencoder (DAE) [3], and binary masking [4]. In contrast, techniques with multi-channel and distortionless processing, e.g., minimum variance distortionless response (MVDR) beamforming, are known to provide a positive ASR improvement [5]. Therefore, multi-channel and distortionless processing are key components of noise robust ASR.

Instead of front-end processing, the framework of neural network-based acoustic modeling represents another crucial approach to noise robust ASR. A number of training procedures for acoustic models have been proposed, such as noise adaptive training [6], noise aware training [7], and DNN adaptation [8], and the effectiveness of these techniques has been demonstrated through various comparative evaluations. In addition, acoustic modeling with different network architectures has also attracted attention; in particular, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are well-known tools for accurate acoustic modeling.

CNNs emphasize the spatial locality of input speech features using small-dimensional convolution filters. In many cases, CNN-based acoustic models outperform DNN-based methods with fully connected layers [9]. The advanced CNN-based acoustic model architectures, which include a network-in-network (NIN)-based cross-feature mapping [10, 11], deep CNN [12], and a very deep CNN with residual network (ResNet) [13], have achieved excellent ASR performance. RNNs have a recurrent connection from past time steps, and are suitable for analyzing sequential data such as speech. Representative RNN architectures include long short-term memory (LSTM) [14, 15] and gated recurrent unit (GRU) [16, 17], which are also known to be effective. In addition, LSTM has been applied to end-to-end ASR systems, e.g., connectionist temporal classification (CTC) [18] and attention encoder–decoder [19]. A convolutional LSTM (CLSTM) [20], which combines CNN and LSTM, has also been developed to solve a spatiotemporal sequence forecasting problem. In the research field of ASR, acoustic modeling using bi-directional convolutional LSTM (BCLSTM) [21] and bi-directional convolutional GRU (BCGRU) [22] has already been proposed. Another noteworthy study [23] proposed a novel end-to-end ASR system using residual CLSTM.

CNNs, RNNs, and convolutional RNNs are powerful and useful tools. However, in our opinion, merely using these network architectures is insufficient to ensure the noise robustness of ASR. Thus, we must address some considerations of building acoustic models with a suitable architecture. For this purpose, we previously proposed an acoustic modeling framework based on a factored network with deep CNN architecture [24]. Here, the factored network, which factors out some function blocks into separate layers in the network [25], is able to increase the noise robustness of ASR by building a neural network with specific roles for each block. Our proposed factored network-based acoustic model has a multi-channel feature enhancement block, and significantly improves the ASR accuracy in noisy environments. This acoustic model architecture is similar to a joint training approach [26, 27, 28], which concatenates the networks of speech/feature enhancement and acoustic models. The parameters of the concatenated network are jointly optimized. In the joint training approach, each network is individually trained in advance based on each optimization criterion. However, individual network training sometimes yields a mismatch between each network, resulting in relatively poor ASR performance. In particular, signal distortion caused by the speech/feature enhancement network is a serious factor in this mismatch. In contrast, our proposed framework avoids this problem because it does not employ individual training.

As mentioned above, our proposed factored network-based acoustic model has a deep CNN architecture. Although CNNs are effective neural network architectures for emphasizing spatial locality, they have no ability to analyze the properties of long-term speech feature sequences. Therefore, in this paper, we introduce various deep convolutional RNN architectures into our proposed factored network-based acoustic modeling framework. Within this

framework, comparative evaluations show that the proposed method successfully improves the ASR accuracy in noisy environments by achieving both spatial locality and long-term analysis.

2. CONVOLUTIONAL RNN ARCHITECTURES

In this section, we briefly review the architectures of RNNs and convolutional RNNs such as LSTM, GRU, CLSTM, and CGRU.

2.1. LSTM and GRU

As mentioned in various studies, LSTM and GRU are formulated to avoid the vanishing gradient problem through a gating mechanism in their recurrent architectures. In particular, LSTM is a well-known RNN architecture with a gating mechanism consisting of an input gate, forget gate, and output gate. LSTM feeds information in the memory cells back to each gate using peephole connections. With this architecture, LSTM is able to handle sequential data, i.e., speech data, efficiently.

In the t-th frame, when the input vector $\boldsymbol{x}_t$ and recurrent input vector $\boldsymbol{h}_{t-1}$ (output vector from the previous frame) are given, LSTM (with recurrent projection) is formulated as:

$$\boldsymbol{i}_t = \sigma\left(\boldsymbol{W}_{xi}\boldsymbol{x}_t + \boldsymbol{W}_{hi}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{ci}\boldsymbol{c}_{t-1} + \boldsymbol{b}_i\right) \quad (1)$$

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_{xf}\boldsymbol{x}_t + \boldsymbol{W}_{hf}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{cf}\boldsymbol{c}_{t-1} + \boldsymbol{b}_f\right) \quad (2)$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot g\left(\boldsymbol{W}_{xc}\boldsymbol{x}_t + \boldsymbol{W}_{hc}\boldsymbol{h}_{t-1} + \boldsymbol{b}_c\right) \quad (3)$$

$$\boldsymbol{o}_t = \sigma\left(\boldsymbol{W}_{xo}\boldsymbol{x}_t + \boldsymbol{W}_{ho}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{co}\boldsymbol{c}_t + \boldsymbol{b}_o\right) \quad (4)$$

$$\boldsymbol{h}_t = \boldsymbol{W}_p\left(\boldsymbol{o}_t \odot g\left(\boldsymbol{c}_t\right)\right) \quad (5)$$

where $\boldsymbol{i}_t$, $\boldsymbol{f}_t$, $\boldsymbol{o}_t$, and $\boldsymbol{c}_t$ denote the input gate, forget gate, output gate, and memory cell, respectively. The $\boldsymbol{W}$ terms and $\boldsymbol{b}$ terms denote weight matrices and bias vectors, e.g., $\boldsymbol{W}_{xi}$ denotes the weight matrix for input vector $\boldsymbol{x}_t$ at input gate $\boldsymbol{i}_t$. $\boldsymbol{W}_{ci}$, $\boldsymbol{W}_{cf}$, and $\boldsymbol{W}_{co}$ denote diagonal weight matrices for the peephole connections. $\boldsymbol{W}_p$ denotes the weight matrix for recurrent projection. $\sigma(\cdot)$, $g(\cdot)$, and $\odot$ denote the sigmoid function, hyperbolic tangent function, and Hadamard product, respectively.

Similar to LSTM, GRU has a recurrent architecture, but contains fewer parameters because of its smaller gating mechanism and lack of memory cells and peepholes. In GRU, instead of memory cells, two gates (update gate and reset gate) control the preservation and output of long-term memory. With the input vector $\boldsymbol{x}_t$ and recurrent input vector $\boldsymbol{h}_{t-1}$, GRU is derived as:

$$\boldsymbol{z}_t = \sigma\left(\boldsymbol{W}_{xz}\boldsymbol{x}_t + \boldsymbol{W}_{hz}\boldsymbol{h}_{t-1} + \boldsymbol{b}_z\right) \quad (6)$$

$$\boldsymbol{r}_t = \sigma\left(\boldsymbol{W}_{xr}\boldsymbol{x}_t + \boldsymbol{W}_{hr}\boldsymbol{h}_{t-1} + \boldsymbol{b}_r\right) \quad (7)$$

$$\tilde{\boldsymbol{h}}_t = g\left(\boldsymbol{W}_{xh}\boldsymbol{x}_t + \boldsymbol{W}_{rh}\left(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}\right) + \boldsymbol{b}_h\right) \quad (8)$$

$$\boldsymbol{h}_t = \boldsymbol{z}_t \odot \boldsymbol{h}_{t-1} + (1 - \boldsymbol{z}_t) \odot \tilde{\boldsymbol{h}}_t \,, \quad (9)$$

where $\boldsymbol{z}_t$ and $\boldsymbol{r}_t$ denote the update gate and reset gate, respectively.

2.2. Convolutional LSTM and convolutional GRU

The basic components of CLSTM and CGRU are almost the same as in LSTM and GRU, respectively. The difference is that the fully connected operations in the input and recurrent input vectors are replaced by the convolution operation. Thus, CLSTM is derived as:

$$\boldsymbol{i}_t = \sigma\left(\boldsymbol{W}_{xi} * \boldsymbol{x}_t + \boldsymbol{W}_{hi} * \boldsymbol{h}_{t-1} + \boldsymbol{W}_{ci}\boldsymbol{c}_{t-1} + \boldsymbol{b}_i\right) \quad (10)$$

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_{xf} * \boldsymbol{x}_t + \boldsymbol{W}_{hf} * \boldsymbol{h}_{t-1} + \boldsymbol{W}_{cf}\boldsymbol{c}_{t-1} + \boldsymbol{b}_f\right) \quad (11)$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot g\left(\boldsymbol{W}_{xc} * \boldsymbol{x}_t + \boldsymbol{W}_{hc} * \boldsymbol{h}_{t-1} + \boldsymbol{b}_c\right) \quad (12)$$

$$\boldsymbol{o}_t = \sigma\left(\boldsymbol{W}_{xo} * \boldsymbol{x}_t + \boldsymbol{W}_{ho} * \boldsymbol{h}_{t-1} + \boldsymbol{W}_{co}\boldsymbol{c}_t + \boldsymbol{b}_o\right) \quad (13)$$

$$\boldsymbol{h}_t = \boldsymbol{W}_p * \left(\boldsymbol{o}_t \odot g\left(\boldsymbol{c}_t\right)\right) \,, \quad (14)$$

where $*$ denotes the convolution operation. Note that, in the above formulation, peephole connections are not given by convolution operations.

On the other hand, CGRU is derived as:

$$\boldsymbol{z}_t = \sigma\left(\boldsymbol{W}_{xz} * \boldsymbol{x}_t + \boldsymbol{W}_{hz} * \boldsymbol{h}_{t-1} + \boldsymbol{b}_z\right) \quad (15)$$

$$\boldsymbol{r}_t = \sigma\left(\boldsymbol{W}_{xr} * \boldsymbol{x}_t + \boldsymbol{W}_{hr} * \boldsymbol{h}_{t-1} + \boldsymbol{b}_r\right) \quad (16)$$

$$\tilde{\boldsymbol{h}}_t = g\left(\boldsymbol{W}_{xh} * \boldsymbol{x}_t + \boldsymbol{W}_{rh} * \left(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}\right) + \boldsymbol{b}_h\right) \quad (17)$$

$$\boldsymbol{h}_t = \boldsymbol{z}_t \odot \boldsymbol{h}_{t-1} + (1 - \boldsymbol{z}_t) \odot \tilde{\boldsymbol{h}}_t \,. \quad (18)$$

In CLSTM and CGRU, the output vector $\boldsymbol{h}_t$ and memory cell $\boldsymbol{c}_t$ are propagated in each time step, and so the dimension of their feature maps should be retained. Therefore, zero padding is applied to each feature map in all convolution operations. In addition, a stride size of one should be applied in all convolution directions. With these restrictions, the computational complexity of CLSTM and CGRU becomes much greater than that of LSTM and GRU.

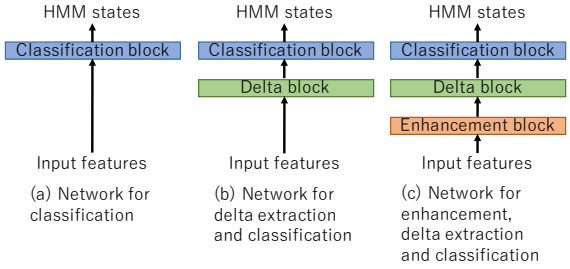

Fig. 1. Architecture of factored networks

3. FACTORED NETWORKS

3.1. Basic idea of factored network

Our proposed factored network-based acoustic modeling framework [24] factors out feature enhancement, delta parameter learning, and hidden Markov model (HMM) state classification into three specific network blocks, as shown in Fig. 1. This framework is similar to the conventional joint training [26, 27, 28]. Here, the joint training individually trains each block with corresponding supervised signals, then performs the joint optimization of all parameters. However, these individual supervised signals are not always optimal for the whole network. For example, when DAE (which learns a forced mapping from a noisy feature to a clean feature) is used for the feature enhancement block, the resulting distortion propagates and adversely affects the whole joint network. It is often difficult to reduce the influence of this distortion, even when joint optimization is used. To avoid this problem, our proposed framework does not use any individual training and requires no additional supervised signals for each block. Namely, the proposed framework defines only the individual network architectures of each block, and performs overall optimization using only the context-dependent (CD) HMM state

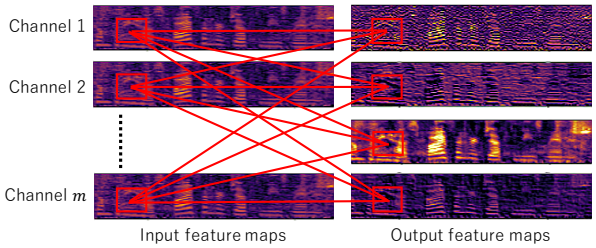

Fig. 2. Feature enhancement with multi-channel 2-dimensional time-frequency filtering

labels, which are the final targets of the acoustic model. With this framework, we believe that noise robust acoustic models can be improved by building a network with specific roles for each block.

3.2. Acoustic modeling with factored network

As a common setup for acoustic modeling, the input feature parameters are m channels of 40 log mel-filter bank (FBank) features, which are extracted using a Hamming window with a 25-ms frame length and 10-ms frame shift. Utterance-wise mean and variance normalization and a context window with 19 (± 9) frames are applied to each utterance.

The classification block is equivalent to the standard DNN-HMM-based acoustic model, which classifies the HMM state in each frame of the input feature parameters. This block is taught to output the posterior probabilities (softmax outputs) of 1,967 CD HMM states under the frame-wise cross-entropy criterion.

Next, the delta block learns dynamic feature extraction with time-domain filtering. Usually, the delta parameters $\Delta \boldsymbol{x}_t$ are extracted using Eq. (19), which is also represented by the convolutional formulation of Eq. (20). Therefore, the filter parameters d_θ of delta parameter extraction can be learnt by the time-domain CNN [29].

$$\Delta \boldsymbol{x}_t = \frac{\sum_{\theta=1}^{\Theta} \theta \left(\boldsymbol{x}_{t+\theta} - \boldsymbol{x}_{t-\theta} \right)}{2 \sum_{\theta=1}^{\Theta} \theta^2} \tag{19}$$

$$= \sum_{\theta=-\Theta}^{\Theta} d_\theta \cdot \boldsymbol{x}_{t+\theta} \ , \tag{20}$$

where Θ denotes the delta window length and $d_\theta = \theta \Big/ 2 \sum_{\tilde{\theta}=1}^{\Theta} \tilde{\theta}^2$.

Finally, the enhancement block learns multi-channel feature enhancement with 2-dimensional time-frequency filtering. As seen in Fig. 2, the input feature parameters are given as m channels of noisy FBank feature maps, and then 2-dimensional CNN-based time-frequency filtering is applied to each input channel. This method sums the output of each convolution filter obtained by all channels [24], and is therefore expected to work in a way that resembles a beamformer fixed to the front direction.

3.3. Factored network with convolutional RNN architectures

In this paper, we report the results of comparative evaluations between various factored network-based acoustic models with convolutional RNN architectures, as shown in Fig. 3.

The model shown in Fig. 3(a) is obtained by conventional joint training with DAE and a clean acoustic model. DAE, i.e., the enhancement layer of this model, is trained in advance as a mapping function from m channels of noisy FBank feature maps to a one-channel clean FBank feature map, and consists of one 2-dimensional time-frequency domain CNN layer and one fully connected layer. The clean acoustic model consists of a delta block and a classification block, and is trained using clean FBank features. Afterwards, DAE and the clean acoustic model are concatenated and all parameters are jointly optimized using m channels of noisy FBank features.

Figure 3(b) illustrates the structure of our previous factored network-based acoustic model [24]. In this model, the enhancement, delta, and classification blocks consist of a 2-dimensional time-frequency domain CNN layer, time-domain CNN layers, and frequency-domain CNN-NIN layers, and fully connected layers. As already mentioned, unlike the joint training of Fig. 3(a), this model concatenates all blocks without individual training and with no additional supervised signals. All parameters are optimized using only the CD HMM state labels.

The models shown in Figs. 3(c)–3(f) are the main evaluation targets of this paper. In these models, the CNN layers in Fig. 3(b) are replaced with convolutional RNNs, i.e., CLSTM, BCLSTM, CGRU, or BCGRU. In the enhancement block, efficient feature enhancement is realized by convolutional RNN architectures consisting of local characteristics analysis based on 2-dimensional time-frequency filtering and long-term sequential analysis based on recurrent connections. Here, the CNN layers of the delta block are not replaced, because the dynamic feature extraction learning with time-domain CNN filtering is essential to the delta block. The classification block stacks one or two convolutional RNN units consisting of a frequency-domain filtering layer, NIN-based 1×1 cross-mapping layer, and max pooling layer. A CNN-based dimension-reduction layer is inserted between the last convolutional RNN unit and the first fully connected layer, because convolutional RNN units output very-high-dimensional tensors.

4. EXPERIMENTS

4.1. Experimental setup

We conducted ASR evaluations using the CHiME-3 corpus [30], which was recorded using a tablet device equipped with six microphones. Various noise environments are included in the CHiME-3 corpus: a public transportation platform (BUS), cafeteria (CAF), pedestrian area (PED), and street junction (STR). The training set consists of 1,600 real and 7,138 simulated (simu) utterances spoken by 4 and 83 different people, respectively. The development (dev) and evaluation (eval) sets consist of 3,280 and 2,640 utterances, respectively, each containing equal quantities of real and simulated data. Both the real and simulated sets were spoken by four speakers. We did not use any of the speech data recorded by a second microphone, because it was located behind the tablet device. Therefore, the number of input channels was $m = 5$.

All acoustic models were trained using TensorFlow [31], and the evaluations (ASR decoding) with trained neural networks were conducted using the Kaldi toolkit [32]. The target CD HMM state labels of the training and development sets were obtained using the latest Kaldi CHiME3 recipe [33]. All networks with recurrent architectures used a truncated back-propagation through time (BPTT) [34] in the training procedure. The step size of the truncated BPTT was set to 20. The parameters of each network were randomly initialized and optimized using momentum stochastic gradient descent with a mini-batch of 320 frames and an initial learning rate of 0.01. Language modeling also followed the latest Kaldi CHiME3 recipe. The ASR experiments were performed using fully composed trigram weighted finite state transducers with the already-mentioned acoustic models. The evaluation criterion was the word error rate (WER).

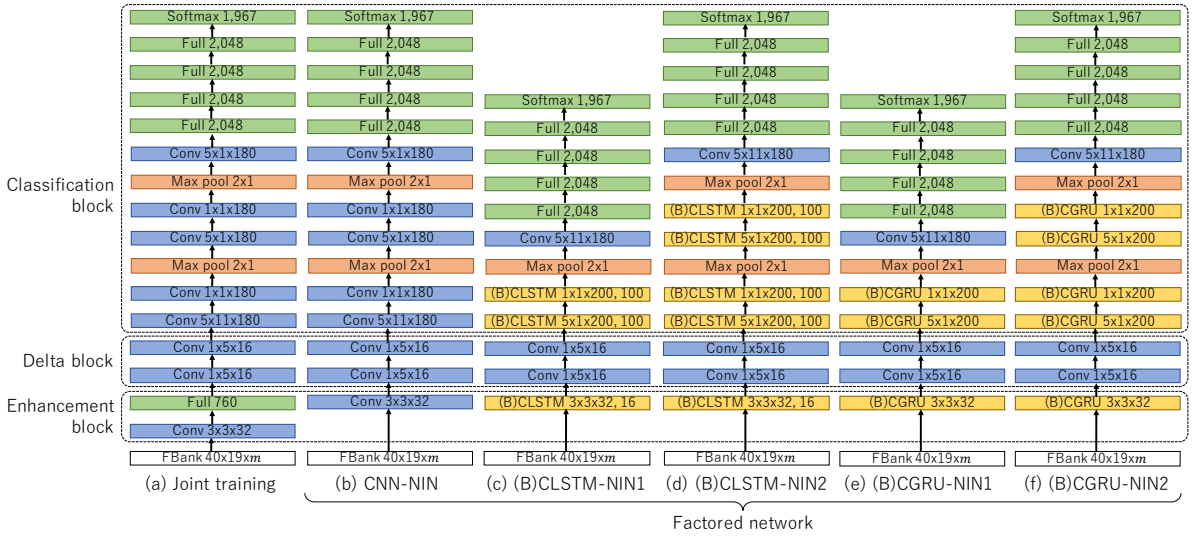

Fig. 3. Network architectures of acoustic models. "Full n", "Conv $f \times t \times ch$", "(B)CLSTM $f \times t \times cell, prj$", "(B)CGRU $f \times t \times gt$", "Max pool $f \times t$" denote a fully connected layer with n nodes, a convolutional layer with ch channels filter of f frequency bands and t time frames, a (B)CLSTM layer with $cell$ channels $f \times t$ memory cell filter and prj channels 1×1 recurrent projection filter, a (B)CGRU layer with gt channels $f \times t$ gate filter, and a max pooling layer with $f \times t$ sub-sampling, respectively.

Table 1. ASR results with various acoustic models in terms of WER (%)

Network type	dev			eval		
	simu	real	avg.	simu	real	avg.
Standard AM	10.63	11.41	11.02	12.52	19.36	15.94
+ Delta block	10.82	11.30	11.06	12.19	19.06	15.63
Joint training	9.52	10.80	10.16	12.20	18.64	15.42
CNN-NIN	9.23	9.84	9.53	11.74	16.85	14.23
CLSTM-NIN1	8.56	9.73	9.14	10.00	16.92	13.46
CLSTM-NIN2	7.58	8.75	8.17	10.21	15.18	12.70
BCLSTM-NIN1	7.93	9.02	8.48	**9.53**	15.68	12.61
BCLSTM-NIN2	**7.51**	**8.63**	**8.07**	10.43	**14.69**	**12.56**
CGRU-NIN1	8.31	9.16	8.73	9.73	16.10	12.92
CGRU-NIN2	7.94	8.93	8.43	9.99	15.51	12.75
BCGRU-NIN1	8.75	9.71	9.23	10.30	18.20	14.24
BCGRU-NIN2	8.45	9.50	8.97	11.26	18.29	14.78

4.2. Experimental results

Table 1 summarizes the ASR results obtained with all acoustic models, as shown in Fig. 3. In the table, "Standard AM" indicates results obtained using acoustic model with only the classification block of Fig. 3(b). We define these results as the baseline, and denote additional baseline with the delta block as "Standard AM + Delta block."

The improvements made by "Joint training" are small, whereas the results obtained with "CNN-NIN" exhibit noticeable improvements. These results reveal that our proposed factored network-based acoustic modeling is effective for ASR in noisy environments.

As seen in the table, the results obtained with CLSTM or BCLSTM indicate further significant improvements from "CNN-NIN." With these results, we can confirm that spatial locality and long-term analysis with convolutional RNN architectures is important for ASR, and also functions effectively in the factored network-based acoustic modeling framework. Here, "BCLSTM-NIN2," which has a bi-directional network and two stacked convolutional RNN units, provided the best results. This model has the most complicated network architecture of the models evaluated in this paper. Hence, it is difficult to use this model for speech applications that require real-time processing. The results obtained with "CLSTM-NIN2" are slightly inferior to those with "BCLSTM-NIN2," but "CLSTM-NIN2" has a unidirectional network architecture. As it also has fewer parameters than "BCLSTM-NIN2," this model is suitable for real-time processing.

The results obtained with CGRU exhibit similar trends to those given by CLSTM. These results are slightly inferior to those from CLSTM, but exhibit sufficient improvements when compared with "CNN-NIN." On the other hand, the results obtained with BCGRU was insufficient in some conditions. Since CGRU has no recurrent projection connection in principle, their output feature dimension becomes much higher than that of CLSTM. In addition, the number of parameters of BCGRU is twice that of CGRU due to their bi-directional architecture. Therefore, BCGRU has the huge number of parameters, it can be considered that this problem influenced the ASR performance.

5. CONCLUSIONS

We have described a factored network-based acoustic modeling framework with various deep convolutional RNN architectures for noise robust ASR. In the framework, we factored out feature enhancement, delta parameter learning, and HMM state classification into three network blocks. By formulating each block with convolutional RNNs, the proposed method can achieve both spatial locality and long-term analysis. We conducted various comparative evaluations on the deep convolutional RNN architectures. From these evaluations, we confirmed that our proposed framework successfully improves the ASR performance by carefully choosing a suitable convolutional RNN network architecture. In future, we plan to build factored networks with very deep convolutional RNN architectures using the ResNet framework.

6. REFERENCES

[1] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. on ASSP*, vol. 27, no. 2, pp. 113–120, April 1979.

[2] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator," *IEEE Trans. on ASSP*, vol. 32, pp. 1109–1121, December 1984.

[3] M. Fujimoto and T. Nakatani, "Feature enhancement based on generative-discriminative hybrid approach with GMMs and DNNs for noise robust speech recognition," in *Proc. of ICASSP '15*, April 2015, pp. 5019–5023.

[4] B. Li and K. C. Sim, "Improving robustness of deep neural networks via spectral masking for automatic speech recognition," in *Proc. of ASRU '13*, December 2013, pp. 279–284.

[5] T. Higuchi, N. Ito, T. Yoshioka, and T. Nakatani, "Robust MVDR beamforming using time-frequency masks for online/offline ASR in noise," in *Proc. of ICASSP '16*, March 2015, pp. 5210–5214.

[6] A. Narayanan and D. Wang, "Joint noise adaptive training for robust automatic speech recognition," in *Proc. of ICASSP '14*, May 2014, pp. 2523–2527.

[7] M. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. of ICASSP '13*, May 2013, pp. 7398–7402.

[8] J. Li, J. T. Huang, and Y. Gong, "Factorized adaptation for deep neural network," in *Proc. of ICASSP '14*, May 2014, pp. 5537–5541.

[9] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE Trans. on ASLP*, vol. 22, no. 10, pp. 1533–1545, October 2014.

[10] M. Lin, Q. Chen, and S. Yan, "Network in network," arXiv:1312.4400v3, 2014.

[11] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi, S. Araki, and T. Nakatani, "The NTT CHiME-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices," in *Proc. of ASRU '15*, December 2015, pp. 436–443.

[12] T. N. Sainath, A. Mohamed, B. Kingsbury, and B Ramabhadran, "Deep convolutional neural networks for LVCSR," *Neural Networks*, vol. 64, pp. 39–48, 2015.

[13] W. Xiong, J. Droppo, X. Huang, F. Seide, M. L. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "The Microsoft 2016 conversational speech recognition system," in *Proc. of ICASSP '17*, March 2017, pp. 5255–5259.

[14] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. of Interspeech '14*, September 2014.

[15] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. of ICASSP '15*, April 2015, pp. 4580–4584.

[16] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. of EMNLP '14*, October 2014, pp. 1724–1734.

[17] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proc. of ICML '15*, July 2015, pp. 2342–2350.

[18] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *Proc. of ASRU '15*, December 2015, pp. 167–174.

[19] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proc. of ICASSP '16*, March 2016, pp. 4945–4949.

[20] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. of NIPS '15*, December 2015, pp. 802–810.

[21] S. Karita, A. Ogawa, M. Delcroix, and T. Nakatan, "Forward-backward convolutional LSTM for acoustic modeling," in *Proc. of Interspeech '17*, August 2017, pp. 1601–1605.

[22] M. Nussbaum-Thom, J. Cui, B. Ramabhadran, and V. Goel, "Acoustic modeling using bidirectional gated recurrent convolutional units," in *Proc. of Interspeech '16*, September 2016, pp. 390–394.

[23] Y. Zhang, W. Chan, and N. Jaitly, "Very deep convolutional networks for end-to-end speech recognition," in *Proc. of ICASSP '17*, March 2017, pp. 4845–4849.

[24] M. Fujimoto, "Factored deep convolutional neural networks for noise robust speech recognition," in *Proc. of Interspeech '17*, August 2017, pp. 3837–3841.

[25] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, and M Bacchiani, "Factored spatial and spectral multichannel raw waveform CLDNNs," in *Proc. of ICASSP '16*, March 2016, pp. 5075–5079.

[26] M. Mimura, S. Sakai, and T. Kawahara, "Joint optimization of denoising autoencoder and DNN acoustic model based on multi-target learning for noisy speech recognition," in *Proc. of Interspeech '16*, September 2016, pp. 3803–3807.

[27] K. H. Lee, T. G. Kang, W. H. Kang, and N. S. Kim, "DNN-based feature enhancement using joint training framework for robust multichannel speech recognition," in *Proc. of Interspeech '16*, September 2016, pp. 3027–3031.

[28] T. Fukuda, O. Ichikawa, G. Kurata, R. Tachibana, S. Thomas, and B Ramabhadran, "Effective joint training of denoising feature space transforms and neural network based acoustic models," in *Proc. of ICASSP '17*, March 2017, pp. 5190–5194.

[29] T. Yoshioka, K. Ohnishi, F. Fang, and T. Nakatani, "Noise robust speech recognition using recent developments in neural networks for computer vision," in *Proc. of ICASSP '16*, March 2016, pp. 5730–5734.

[30] "The 3rd CHiME speech separation and recognition challenge," http://spandh.dcs.shef.ac.uk/chime_challenge/.

[31] "TensorFlow," https://www.tensorflow.org/.

[32] "Kaldi ASR tool-kit," http://kaldi-asr.org/.

[33] "Kaldi CHiME3 recipe with beamforming," https://github.com/kaldi-asr/kaldi/tree/master/egs/chime3.

[34] R. J. Williams and J. Peng, "An efficient gradient-based algorithm for on-line training of recurrent network trajectories," *Neural Computation*, vol. 2, no. 4, pp. 490–501, December 1990.