INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Forward-backward Convolutional LSTM for Acoustic Modeling

Shigeki Karita, Atsunori Ogawa, Marc Delcroix, and Tomohiro Nakatani

NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan
{karita.shigeki, ogawa.atsunori, marc.delcroix, nakatani.tomohiro}@lab.ntt.co.jp

Abstract

An automatic speech recognition (ASR) performance has greatly improved with the introduction of convolutional neural network (CNN) or long-short term memory (LSTM) for acoustic modeling. Recently, a convolutional LSTM (CLSTM) has been proposed to directly use convolution operation within the LSTM blocks and combine the advantages of both CNN and LSTM structures into a single architecture. This paper presents the first attempt to use CLSTMs for acoustic modeling. In addition, we propose a new forward-backward architecture to exploit long-term left/right context efficiently. The proposed scheme combines forward and backward LSTMs at different time points of an utterance with the aim of modeling long term frame invariant information such as speaker characteristics, channel etc. Furthermore, the proposed forward-backward architecture can be trained with truncated back-propagation-through-time unlike conventional bidirectional LSTM (BLSTM) architectures. Therefore, we are able to train deeply stacked CLSTM acoustic models, which is practically challenging with conventional BLSTMs. Experimental results show that both CLSTM and forward-backward LSTM improve word error rates significantly compared to standard CNN and LSTM architectures.

Index Terms: speech recognition, convolutional LSTM, acoustic model, AURORA4, CHiME3

1. Introduction

In state-of-the-art automatic speech recognition (ASR) systems, acoustic models are replaced from fully connected neural networks (NNs) to convolutional neural networks (CNNs) and long short term memory networks (LSTMs) [1]. A system combination of CNNs and LSTMs is a popular technique for ASR because these architectures show strong complementarity [1, 2, 3].

However, the system combinations of many models are hard to construct because they require many tuning parameters such as model selection and voting weight balance. To obtain the strong complementarity of model combination even in a single model, we propose a new acoustic model that combines characteristics of CNN and LSTM by a convolutional LSTM (CLSTM) [4].

CNNs are powerful models that have been shown to outperform conventional fully-connected NNs for many tasks [5].

Table 1: *Word error rates (WERs) of CNNs with different lengths of the context window.*

context	AURORA4	CHiME3
15	**11.7**	10.0
19	11.9	**9.6**
23	12.0	10.7

This may be attributed to their translational invariance property, that makes frequency shift due to speaking styles or speaker variations easier to capture than with fully-connected NNs. However, CNN acoustic models require manual tuning of the configuration of the context window that concatenates fixed length consecutive frames. An ASR accuracy is sensitive to the length of the context window [6, 7]. In Table 1, we report word error rates (WERs) of CNNs with different lengths of the context window. These results show that we have to determine the best length of the context window for each task. In contrast, LSTM acoustic model is free from this problem because it does not have such context window [8]. It captures long dynamic contexts by its recurrent connections and gates.

On the other hand, an output of LSTM loses local structures of the input along a frequency axis unlike CNN. Considering that the noise robustness of CNN appears in its feature maps [5], remaining the structure of the input feature is important for ASR in noisy conditions.

We expect CLSTM to compensate these two problems of the context window and feature structure from a CNN and LSTM. CLSTM was first proposed in recent image processing tasks [4, 9] and has been recently used in an end-to-end ASR system [10]. In this paper, we attempt to use CLSTM for hidden Markov model (HMM) hybrid acoustic models because ASR systems using such acoustic models still achieve the best performance in most ASR tasks.

In addition, bidirectional extensions of LSTMs (BLSTM) [11] have been shown to further improve the performance compared to unidirectional LSTMs [1, 2]. In speech, a phoneme has some future and past dependencies because of co-articulation and linguistic tendency of a word [12]. A motivation of the BLSTM is to take advantage of both future and past information to improve predictions.

However, such a BLSTM architecture is difficult to optimize. For instance, when we train a RNN with sequences of length 1000, it costs the equivalent to forward-backward passes in a neural network (NN) of 1000 layers [13]. This problem affects larger models seriously. Hence, it is common to use a windowed recurrent neural network (RNN) approach (similar to the context window of CNN) to train large networks [14, 8].

In this paper, to address this problem without such a window, we investigate a new method "forward-backward LSTM" (FB-LSTM) that takes forward and backward sequences in a source side (i.e., speech features) and target side (i.e. HMM state labels). Unlike the conventional methods [8, 14], we design our method to be optimized with truncated back propagation through time (BPTT) [15]. Hence, we can train larger FB-LSTMs efficiently with larger mini-batch because it requires only limited range of gradients.

The authors in [8] also propose new architecture that unifies the CNN and gated recurrent unit (GRU) [16]. Although they have a lot of similar motivations with our work, we have three different points:

http://dx.doi.org/10.21437/Interspeech.2017-554

- We adopt CLSTM instead of their gated recurrent convolutional unit (GRCU) because CLSTM derives from the state of the art acoustic models based on CNN or LSTM.
- We propose new forward-backward (FB) architecture that is free from the context window, while they adopt windowed approach [14].
- We examine that our FB-CLSTM models offer better WERs than the CNN and LSTM models in multiple noisy conditions.

This paper is organized as follows. Section 4 describes the related work and our novelty from it. Section 2 and 3 provides our framework of new bidirectional architecture and training methods. Section 5 presents our experimental results. And we conclude our remarks in Section 6.

2. Convolutional LSTM

In this section, we describe the details of our implementation of CLSTM. First, we define the convolutional operator of CNN that is a key part of CLSTM. Second, we derive CLSTM from a LSTM definition with the convolutional operator.

2.1. Convolutional operator

CLSTM has four convolutional layers to compute its gates and cell, while LSTM has four fully-connected layers. Here, we consider a 1-D convolutional layer has a filter size of an odd integer I, stride of 1 and zero padding along a frequency axis. It convolves its weight and an input along frequency axis. When we input a feature $\mathbf{x} \in \mathbb{R}^{M \times F \times T}$ with M input channels, F frequency-bins (i.e. filterbank) and T frames into the convolutional layer, we obtain its output

$$z_{n,f,t} = \sum_{m=1}^{M} \left(\sum_{i=1}^{I} w_{n,m,i} x_{m,f',t} \right) \tag{1}$$

where $w_{n,m,i}$ is a convolutional weight, $n = 1, 2, \ldots, N$ are output channel indices and $f' = f + i - (I+1)/2$. For simplicity, we use the operator $*$ to describe the convolution of multi dimensional arrays (bold lower case) in Eq. (1) as

$$\mathbf{z}_t = \mathbf{w} * \mathbf{x}_t, \tag{2}$$

where the arrays are shaped as follows: $\mathbf{z}_t \in \mathbb{R}^{N \times F}, \mathbf{w} \in \mathbb{R}^{N \times M \times I}$.

2.2. LSTM activation

A conventional LSTM layer can be expressed as

$$\mathbf{f}'_t = \sigma(\mathbf{w}'_{xf}\mathbf{x}'_t + \mathbf{w}'_{hf}\mathbf{h}'_{t-1} + \mathbf{b}'_f), \tag{3}$$
$$\mathbf{i}'_t = \sigma(\mathbf{w}'_{xi}\mathbf{x}'_t + \mathbf{w}'_{hi}\mathbf{h}'_{t-1} + \mathbf{b}'_i), \tag{4}$$
$$\mathbf{a}'_t = \tanh(\mathbf{w}'_{xa}\mathbf{x}'_t + \mathbf{w}'_{ha}\mathbf{h}'_{t-1} + \mathbf{b}'_a), \tag{5}$$
$$\mathbf{o}'_t = \sigma(\mathbf{w}'_{xo}\mathbf{x}'_t + \mathbf{w}_{ho}\mathbf{h}'_{t-1} + \mathbf{b}'_o), \tag{6}$$

where $\mathbf{w}'_{x*} \in \mathbb{R}^{N \times (M \times F)}$, $\mathbf{w}'_{h*} \in \mathbb{R}^{N \times N}$ are weights, $\mathbf{b}'_* \in \mathbb{R}^N$ are biases, $\mathbf{x}' \in \mathbb{R}^{(M \times F) \times T}$ is a flatten input and $\mathbf{h}'_{t-1} \in \mathbb{R}^N$ is a recurrent hidden state. Finally, we obtain the output of LSTM layer $\mathbf{h}_t$ as follows

$$\mathbf{c}'_t = \mathbf{f}'_t \odot \mathbf{c}'_{t-1} + \mathbf{i}'_t \odot \mathbf{a}'_t, \tag{7}$$
$$\mathbf{h}'_t = \mathbf{o}'_t \odot \tanh(\mathbf{c}'_t), \tag{8}$$

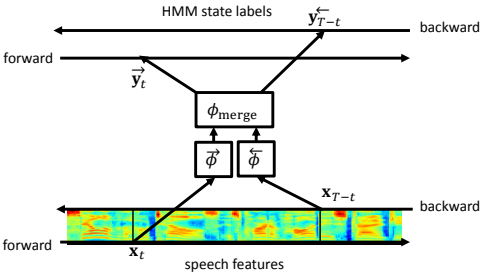

Figure 1: *An overview of our forward-backward (FB) architecture. When we input speech features $\mathbf{x}_t$ to its forward side and $\mathbf{x}_{T-t}$ to its backward side at the same time, FB-LSTM outputs two predictions $\overrightarrow{\mathbf{y}}_t$ and $\overleftarrow{\mathbf{y}}_{T-t}$ of HMM state labels.*

where $\odot$ denotes an element-wise product of vectors. For simplicity, we describe a LSTM layer defined with Eqs. (3)-(8) as

$$\mathbf{h}'_t = \phi(\mathbf{x}'_t). \tag{9}$$

This feature map $\mathbf{h}' \in \mathbb{R}^{N \times T}$ loses local structures along a frequency axis in the input $\mathbf{x} \in \mathbb{R}^{M \times F \times T}$ unlike a CNN.

A CLSTM layer is obtained by replacing matrix-vector products in Eqs. (3)-(6) with the $*$ operator as follows

$$\mathbf{f}_t = \sigma(\mathbf{w}_{xf} * \mathbf{x}_t + \mathbf{w}_{hf} * \mathbf{h}_{t-1} + \mathbf{b}_f), \tag{10}$$
$$\mathbf{i}_t = \sigma(\mathbf{w}_{xi} * \mathbf{x}_t + \mathbf{w}_{hi} * \mathbf{h}_{t-1} + \mathbf{b}_i), \tag{11}$$
$$\mathbf{a}_t = \tanh(\mathbf{w}_{xa} * \mathbf{x}_t + \mathbf{w}_{ha} * \mathbf{h}_{t-1} + \mathbf{b}_a), \tag{12}$$
$$\mathbf{o}_t = \sigma(\mathbf{w}_{xo} * \mathbf{x}_t + \mathbf{w}_{ho} * \mathbf{h}_{t-1} + \mathbf{b}_o). \tag{13}$$

where σ denotes a sigmoid function and $\mathbf{b}_* \in \mathbb{R}^{N \times F}$ are biases for each gate. Note that, the last two element-wise computations to obtain a output feature in Eqs. (7) and (8) are the same to CLSTM.

3. Forward-backward architecture

Truncated BPTT is the most popular technique to train unidirectional RNN without resetting its recurrent hidden states and cells. Because the conventional BPTT loss $L = \sum_{t=1}^{T} L_t/T$ has very long back propagation graph (e.g., $T = 2000$), we adopt truncated BPTT to accumulate losses along time within bounded range as $L = \sum_{t=kT_B+1}^{\min(T,(k+1)T_B)} L_t/T_B$, where L_t is a frame-wise loss at time t, T_B is a step size of truncated BPTT and $k = 0, 1, \ldots, \lfloor T/T_B \rfloor - 1$ are update indices.

For bidirectional RNNs, the windowed approach [8, 14] is proposed but it resets its recurrent hidden states and cells between update steps. In addition to that, it brings the same constraint of the context window like CNN models that we discussed in Section 1. In this section, we explain the detail of our forward-backward architecture that enables both truncated BPTT and bidirectional context modeling.

3.1. Definitions

We define our forward-backward LSTM (FB-LSTM) that predicts HMM state labels $\mathbf{y}_1, \ldots, \mathbf{y}_T$ from a input feature $\mathbf{x}_1, \ldots, \mathbf{x}_T$ as follows

$$\overrightarrow{\mathbf{h}}_t = \overrightarrow{\phi}(\mathbf{x}_t) \tag{14}$$
$$\overleftarrow{\mathbf{h}}_{T-t} = \overleftarrow{\phi}(\mathbf{x}_{T-t}) \tag{15}$$
$$(\overrightarrow{\mathbf{g}}_t, \overleftarrow{\mathbf{g}}_{T-t}) = \phi_{\text{merge}}\left(\overrightarrow{\mathbf{h}}_t, \overleftarrow{\mathbf{h}}_{T-t}\right) \tag{16}$$
$$\overrightarrow{\mathbf{y}}_t = \text{softmax}\left(\overrightarrow{\mathbf{g}}_t\right) \tag{17}$$
$$\overleftarrow{\mathbf{y}}_{T-t} = \text{softmax}\left(\overleftarrow{\mathbf{g}}_{T-t}\right) \tag{18}$$

where $\overrightarrow{\phi}$ and $\overleftarrow{\phi}$ are LSTMs as shown in Eq. (9) that perform forward and backward feature extractions over a source sequence $\mathbf{x}$ to obtain $\overrightarrow{\mathbf{h}}$ and $\overleftarrow{\mathbf{h}}$ respectively. The final layer ϕ_{merge} consists of several LSTM layers and creates two outputs $\overrightarrow{\mathbf{g}}_t, \overleftarrow{\mathbf{g}}_{T-t}$ from the concatenated two feature inputs $\overrightarrow{\mathbf{h}}_t, \overleftarrow{\mathbf{h}}_{T-t}$. We illustrate an overview of this architecture in Fig. 1. During decoding, we use an averaged prediction $(\overrightarrow{\mathbf{y}}_t + \overleftarrow{\mathbf{y}}_t)/2$ as a prediction of the HMM state label at time t. As variants of the layer ϕ_{merge}, we consider three scenarios as follows: (a) additional separated LSTM layers instead of a merging layer, (b) a merging LSTM layer before output LSTM layer, (c) a merging LSTM layer after input LSTM layers. In Fig. 2, we visualize these variants referred to FB-LSTM (a)-(c). We expect that the merging layer encourages LSTM based acoustic models to exploit the future and past contexts or time-invariant information in an utterance (e.g., speaker and channel characteristics).

3.2. Optimization with truncated BPTT

Our forward-backward architecture does not require full sequences of an utterance $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$ and target HMM state labels $\mathbf{y}_1, \ldots, \mathbf{y}_T$ in a training stage with truncated BPTT except for the decoding stage. When truncated BPTT optimizes the FB-LSTM, it requires this limited range loss as

$$L = \frac{1}{T_B} \sum_{t=kT_B+1}^{\min(T,(k+1)T_B)} \left\{ \mathrm{KL}(\hat{\mathbf{y}}_t || \overrightarrow{\mathbf{y}}_t) + \mathrm{KL}(\hat{\mathbf{y}}_{T-t} || \overleftarrow{\mathbf{y}}_{T-t}) \right\} \tag{19}$$

and initial recurrent hidden states $\overrightarrow{\mathbf{h}}_{kT_B}$, $\overleftarrow{\mathbf{h}}_{T-kT_B}$, $\overrightarrow{\mathbf{g}}_{kT_B}$, $\overleftarrow{\mathbf{g}}_{T-kT_B}$ as the last hidden outputs from the previous update, where $\hat{\mathbf{y}}$ is a target sequence of HMM state labels as supervisions. As described in Section 3.1, to output $\overrightarrow{\mathbf{y}}_t$ and $\overleftarrow{\mathbf{y}}_{T-t}$, FB-LSTM only require the previous hidden outputs $\overrightarrow{\mathbf{h}}_{t-1}, \overrightarrow{\mathbf{g}}_{t-1}, \overleftarrow{\mathbf{h}}_{T-t+1}$ and $\overleftarrow{\mathbf{g}}_{T-t+1}$. Because it does not require $\overrightarrow{\mathbf{h}}_{t+1}, \overrightarrow{\mathbf{g}}_{t+1}, \overleftarrow{\mathbf{h}}_{T-t-1}$ and $\overleftarrow{\mathbf{g}}_{T-t-1}$ like BLSTM, truncated gradients are accumulated only within forward and backward ranges $[kT_B+1, (k+1)T_B], [T-(k+1)T_B, T-kT_B+1]$. Note that, during decoding, we cannot create predictions from such limited ranges because we use an averaged prediction of two predictions $\overrightarrow{\mathbf{y}}$ and $\overleftarrow{\mathbf{y}}$ from a single FB model. However, this is not a problem because we do not need to store any gradients for backward paths during decoding.

4. Related work

There have been many attempts to extend BLSTM architectures before our FB-CLSTM. The work published in [11] applies deep BLSTM into acoustic modeling for the Wall Street Journal (WSJ) task. The deep BLSTM is one of the stacked BLSTM architecture that an upper BLSTM layer takes an input as merged outputs from a lower BLSTM layer. Before applying to acoustic modeling, the authors applied it first for an end-to-end ASR system without language models [17]. The authors expected that deep BLSTM has the ability to learn an implicit language model from training targets of HMM alignments. In recent ASR challenge CHiME4, a single acoustic model based on deep BLSTM and CNN achieved the best results without the system combination [18].

In addition, optimization methods are developed to train the recent bidirectional models. On the handwriting recognition task, chunk BPTT is proposed in [19, 20] to train bidirectional models efficiently. For acoustic modeling, chunk BPTT

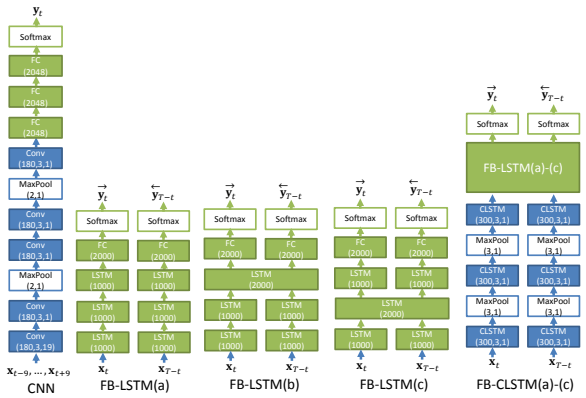

Figure 2: *An illustration of architectures of acoustic models. Note that, architectures of non FB models are the same to single sides of FB(a) models. Our proposed forward or backward architectures predicts two distribution of HMM state lables* $\overrightarrow{\mathbf{y}}_t, \overleftarrow{\mathbf{y}}_{T-t}$ *from two forward-backward inputs* $\mathbf{x}_t$ *and* $\mathbf{x}_{T-t}$*. We examined three variants of FB-LSTM and FB-CLSTM: (a) no merging layer (b) merge before output layers (c) merge after input layers.*

equivalents are adopted to optimize windowed BLSTM [14] and BGRCU [8] with more complex procedures. Like the context window of the CNN, these approaches only exploit limited ranges of input and recurrent hidden state sequences. Hence, the novelty of our forward-backward approach is an ability to capture unlimited ranges of input and recurrent hidden state sequence while it requires only limited ranges of gradients by truncated BPTT.

5. Experiments

In this section, we examine our methods in AURORA4 [21] and CHiME3 [22] noisy speech corpora.

5.1. Conditions

For AURORA4 task, we used the same configurations as in [23]. We used Gaussian mixture model (GMM)-HMM alignments as targets in training of NN-HMM hybrid models. Our GMM models were trained with perceptual linear prediction (PLP) features of 13 coefficients, their first and second temporal derivatives. NN-HMM hybrid acoustic models were trained with 7137 utterances from the multi-condition training data sets. We used the log-Mel filterbank (FBANK) features with 40 coefficients and their first and second temporal derivatives as input features for NN-HMM hybrid models. The ASR results were obtained by one-pass decoding with a HMM of 3042-states and the provided bi-gram language model (LM) with fixed LM scale factor of 14.0. We did not use any speaker adaptation and front-end enhancement.

For CHiME3 task, we followed the configuration of [24] that is quite similar to that of the AURORA4 task. The largest difference is that the input of the CNN was 80 dimensional FBANK without derivatives because it was the best configuration for this task. When we trained NN-HMM acoustic models, we used 8738 utterances that consist of 7138 simulated and 1600 real ones from a single channel "CH5". The dev-set and eval-set consist of 6560 and 5280 utterances of fifty-fifty real and simulated environments. The one-pass decoding is operated with a HMM of 5976-states and tri-gram LM with a fixed LM scale factor of 15.0.

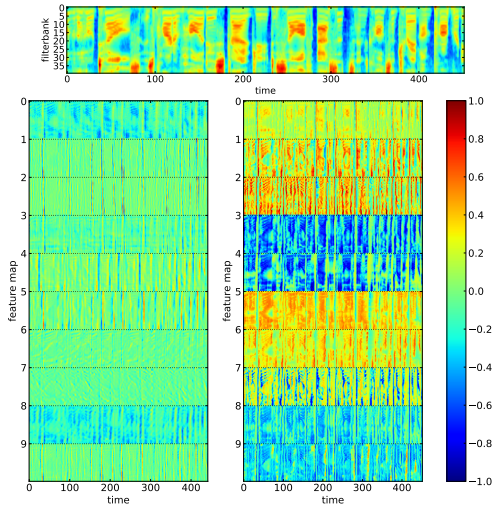

Figure 3: *(top): An example of FBANK input feature. (bottom-left): 10 feature maps corresponding to the input. These are randomly obtained from CNN acoustic model's activation of the first layer. (bottom-right): CLSTM acoustic model's 10 feature maps. Note that, all the values are rescaled into* $[-1, 1]$.

5.2. Structures of acoustic models

Figure 2 summarizes the models that we investigated. First, we configured the baseline as the best of CNN and LSTM acoustic models for our settings. For CNN models in the AURORA4 task, we reused the best result of the officially provided bi-gram decoding that is described as "$\mathrm{B7_Q - PReLU}$" with the context window of 11 frames in [23]. In the CHiME3 task, we created our best CNN model that is illustrated in Fig. 2 that has the context window of 19 frames and performs better than the the previous model in [24]. Our LSTM models have three LSTM layers of 1000 units.

Second, we created our CLSTM and FB-LSTM and FB-CLSTM models from the LSTM baseline. To analyze differences of the merging layer ϕ_{merge}, we explored three ways to perform acoustic modeling with FB-LSTM and FB-CLSTM for each type (a), (b) and (c) as described in Section 3.1. For all the CLSTM-based models, we stack three CLSTM layers of 300 channels and 3×1 filters before (FB-)LSTM layers of (a) – (c) variants as shown in Fig. 2 because a similar architecture was found to be effective in [8]. During training of all the LSTM based models, we delay the target HMM state labels by four frames as described in [25].

We used stochastic gradient descent to train all the NNs with a learning rate of 0.04 that is reduced by a factor of 0.5 until 0.0004, mini-batch size of 128 and momentum of 0.9 in every task and model. For the CNN model, we set a different learning rate from 0.01 to 0.0001 for the AURORA4 and from 0.08 to 0.0008 for the CHiME3 as the best configuration respectively. For LSTM based models, we set the truncated BPTT length T_B to seven. We regularized all the fully connected and LSTM layers with dropout of rate 0.5 during the training. According to [26], we did not apply the dropout into lateral connections in the LSTM. Furthermore, the LSTM-based models have an additional operation of gradient clipping to make convergences faster and stable with a L2 norm threshold of 10.0 [27].

5.3. Comparison on feature maps

An image in Fig. 3 (top) shows an example of FBANK feature for acoustic models. Two images on the bottom-left and bottom-right are several feature maps obtained from CNN and CLSTM acoustic models respectively. Colors in the images are related to the activations, where high-middle-low values correspond to red-green-blue colors. Although the contrasts are different, CLSTM's feature maps also seem to keep the time-frequency structure like CNN's.

Table 2: *WERs of acoustic models for the AURORA4 task. We describe subsets of clean and noise, channel, noise+channel as A to D respectively.*

	WER (%) eval				
	A	B	C	D	average
CNN [23]	4.5	7.6	7.0	16.5	11.1
LSTM	5.8	9.9	11.1	21.8	14.8
CLSTM	5.1	8.5	8.0	18.3	12.4
FB-LSTM(a)	5.3	8.4	8.3	18.4	12.5
FB-LSTM(b)	5.1	8.7	9.0	19.2	13.0
FB-LSTM(c)	4.6	8.0	8.0	18.0	12.0
FB-CLSTM(a)	3.9	6.8	6.2	15.2	10.1
FB-CLSTM(b)	4.0	7.0	6.2	15.2	10.3
FB-CLSTM(c)	**3.7**	**6.8**	**5.7**	**15.0**	**9.9**

Table 3: *WERs of acoustic models for the CHiME3 task.*

	WER (%)					
	dev			eval		
	sim	real	average	sim	real	average
CNN	11.3	11.5	11.4	13.3	19.5	16.4
LSTM	15.3	15.8	15.5	18.3	27.5	22.9
CLSTM	12.5	13.2	12.9	15.4	23.1	19.2
FB-LSTM(a)	16.0	16.7	16.4	20.2	28.4	24.3
FB-LSTM(b)	12.3	12.9	12.6	15.6	21.5	18.6
FB-LSTM(c)	12.7	13.1	12.9	15.5	21.5	18.5
FB-CLSTM(a)	10.0	10.4	10.2	12.1	17.4	14.8
FB-CLSTM(b)	**9.6**	10.2	9.9	11.8	**16.6**	14.2
FB-CLSTM(c)	**9.6**	**9.9**	**9.8**	**11.6**	16.7	**14.1**

5.4. Evaluation on AURORA4

In Table 2, we summarize the result of WERs for the AURORA4 task. For this task, the baseline CNN greatly outperforms the LSTM. The CLSTM and FB-LSTM improve performance compared to the LSTM but still perform worse than the CNN. The FB-CLSTM outperforms all other configurations and achieved 11–33% WER reductions from the CNN [23] and LSTM.

5.5. Evaluation on CHiME3

In Table 3, we summarize our experimental results for the CHiME3 task. The results show the same trend with the AURORA4 task. We can see the significant improvement from CNN and LSTM to FB-CLSTMs with relative WER reductions of 14 – 38%. However, we can see the degradation in FB-LSTM(a) while (b) and (c) improved. This result indicates that the merging layer has a important roll to make the performance stable as seen in multi-task learning.

6. Conclusion

This paper has presented the first attempt to use a CLSTM for acoustic modeling. We also investigate a new forward-backward architecture to efficiently exploit long-term left/right contexts. The experimental results showed the advantage of our FB-CLSTM models from the conventional CNN and LSTM. On AURORA4 and CHiME3, our FB-CLSTM acoustic model provided 11% – 33% and 14% – 38% relative WER reductions from the strong baseline deep CNN or LSTM models respectively.

7. References

[1] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "The microsoft 2016 conversational speech recognition system," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2017.

[2] G. Saon, H.-K. J. Kuo, S. Rennie, and M. Picheny, "The IBM 2016 English Conversational Telephone Speech Recognition System," *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association*, pp. 3–7, 2015.

[3] K. J. Geras, A.-r. Mohamed, R. Caruana, G. Urban, S. Wang, O. Aslan, M. Philipose, M. Richardson, and C. Sutton, "Blending LSTMs into CNNs," *International Conference on Learning Representation (ICLR) Workshop*, pp. 1–11, 2015.

[4] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," *Advances in Neural Information Processing Systems 28*, pp. 802–810, 2015.

[5] Y. Qian, M. Bi, T. Tan, and K. Yu, "Very Deep Convolutional Neural Networks for Noise Robust Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2263–2276, 2016.

[6] J. Pan, C. Liu, Z. Wang, Y. Hu, and H. Jiang, "Investigation of deep neural networks (DNN) for large vocabulary continuous speech recognition: Why DNN surpasses GMMS in acoustic modeling," in *2012 8th International Symposium on Chinese Spoken Language Processing*. IEEE, 2012, pp. 301–305.

[7] A. L. Maas, P. Qi, Z. Xie, A. Y. Hannun, C. T. Lengerich, D. Jurafsky, and A. Y. Ng, "Building DNN acoustic models for large vocabulary speech recognition," *Computer Speech & Language*, vol. 41, pp. 195–213, 2017.

[8] M. Nussbaum-Thom, J. Cui, B. Ramabhadran, and V. Goel, "Acoustic Modeling Using Bidirectional Gated Recurrent Convolutional Units," in *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association*, 2016, pp. 390–394.

[9] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel Recurrent Neural Networks," *Proceedings of The 33rd International Conference on Machine Learning*, pp. 1747–1756, 2016.

[10] Y. Zhang, W. Chan, and N. Jaitly, "Very Deep Convolutional Networks for End-to-End Speech Recognition," pp. 10–14, 2016. [Online]. Available: http://arxiv.org/abs/1610.03022

[11] A. Graves, N. Jaitly, and A. R. Mohamed, "Hybrid speech recognition with Deep Bidirectional LSTM," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2013 - Proceedings*, 2013, pp. 273–278.

[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[13] I. Sutskever, "Training Recurrent neural Networks," *PhD thesis*, p. 101, 2013. [Online]. Available: http://www.cs.utoronto.ca/~ilya/pubs/ilya_sutskever_phd_thesis.pdf

[14] A. Mohamed, F. Seide, D. Yu, J. Droppo, A. Stolcke, G. Zweig, and G. Penn, "Deep Bi-directional Recurrent Networks Over Spectral Windows," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*. IEEE Institute of Electrical and Electronics Engineers, 2015, pp. 78–83.

[15] 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, 1990.

[16] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN EncoderDecoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2014, pp. 1724–1734.

[17] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, no. 3. IEEE, 2013, pp. 6645–6649.

[18] J. Heymann, L. Drude, and R. Haeb-umbach, "Wide Residual BLSTM Network with Discriminative Speaker Adaptation for Robust Speech Recognition," *The 4th International Workshop on Speech Processing in Everyday Environments, CHiME4 Workshop*, no. 1, 2016.

[19] P. Doetsch, M. Kozielski, and H. Ney, "Fast and Robust Training of Recurrent Neural Networks for Offline Handwriting Recognition," in *2014 14th International Conference on Frontiers in Handwriting Recognition*, vol. 2014-Decem. IEEE, 2014, pp. 279–284.

[20] K. Chen, Z.-j. Yan, and Q. Huo, "A Context-Sensitive-Chunk BPTT Approach to Training Deep LSTM / BLSTM Recurrent Neural Networks for Offline Handwriting Recognition," *13th International Confrence on Document Analysis and Recognition - ICDAR'15*, pp. 411–415, 2015.

[21] N. Parihar and J. Picone, *Aurora Working Group: DSR Front End LVCSR Evaluation AU/384/02*, 2002. [Online]. Available: http://aurora.hsnr.de/aurora-4.html

[22] J. Barker *et al.*, "The third chime speech separation and recognition challenge: Dataset, task and baselines," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015, pp. 504–511.

[23] T. Yoshioka, K. Ohnishi, F. Fang, and T. Nakatani, "Noise robust speech recognition using recent developments in neural networks for computer vision," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5730–5734.

[24] 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 *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 436–443.

[25] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *INTERSPEECH, 15th Annual Conference of the International Speech Communication Association*, 2014, pp. 338–342.

[26] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent Neural Network Regularization," 2014. [Online]. Available: http://arxiv.org/abs/1409.2329

[27] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training Recurrent Neural Networks," *Proceedings of The 30th International Conference on Machine Learning*, vol. 28, no. 3, pp. 1310–1318, 2013.