Layer Trajectory BLSTM

Eric Sun, Jinyu Li, Yifan Gong

Microsoft Speech and Language Group
{sun.eric,jinyli,Yifan.Gong}@microsoft.com

Abstract

Recently, we proposed layer trajectory (LT) LSTM (ltLSTM) which significantly outperforms LSTM by decoupling the functions of senone classification and temporal modeling with separate depth and time LSTMs. We further improved ltLSTM with contextual layer trajectory LSTM (cltLSTM) which uses the future context frames to predict target labels. Given bi-directional LSTM (BLSTM) also uses future context frames to improve its modeling power, in this study we first compare the performance between these two models. Then we apply the layer trajectory idea to further improve BLSTM models, in which BLSTM is in charge of modeling the temporal information while depth-LSTM takes care of senone classification. In addition, we also investigate the model performance among different LT component designs on BLSTM models. Trained with 30 thousand hours of EN-US Microsoft internal data, the proposed layer trajectory BLSTM (ltBLSTM) model improved the baseline BLSTM with up to 14.5% relative word error rate (WER) reduction across different tasks.

Index Terms: BLSTM, layer trajectory, future context, senone classification

1. Introduction

Automatic speech recognition (ASR) has achieved significant progress recently since the deep feedforward neural networks (DNNs) [1] was replaced by long short-term memory (LSTM) [2] based recurrent neural networks (RNNs). Standard RNNs usually have the gradient vanishing or exploding issues which can be alleviated by LSTM. Therefore, LSTM-RNNs [3,4,5,6,7,8] show better capacity of capturing long span temporal information and have been shown to outperform DNNs on a variety of ASR tasks [9].

Recently, we proposed a layer trajectory LSTM (ltLSTM) model [10], in which depth-LSTMs are used to scan the outputs of multi-layer time-LSTMs to extract information for label classification. By decoupling the temporal modeling and senone classification tasks, ltLSTM allows time and depth LSTMs focusing on their individual tasks. In addition, the depth-LSTM can make deeper LSTM training easier since it creates auxiliary connections for gradient flow to further alleviate the gradient vanishing and exploding issues. Compared to traditional LSTMs or residual LSTMs [11, 12], ltLSTM models achieved significant accuracy improvement [10]. In [13], the generated ltLSTM structure with the depth processing block was proposed, in which gated and max-pooling [14, 15, 16] feedforward units were integrated for the depth processing component to reduce the computational cost.

To further improve the performance of ltLSTM models, we recently proposed contextual layer trajectory LSTM (cltLSTM) which uses context frames to capture future information [17]. As shown in [17], the lookahead embeddings from either time-LSTM or depth-LSTM helped to improve recognition performance significantly.

The extreme case of using lookahead frames is the bi-directional LSTM (BLSTM) modeling [18] since it has the backward LSTM running from the end of utterances to the beginning. Since BLSTM models can capture very long future information in speech recognition, they often got superior performance than uni-directional LSTM models [19].

Given the success of ltLSTM and cltLSTM which apply the concept of layer trajectory (LT) processing to LSTM modeling, a natural question is whether we can improve BLSTM with the layer trajectory concept. In this paper, we propose layer trajectory BLSTM (ltBLSTM) by decoupling the tasks of senone classification using depth LSTM and temporal modeling using time BLSTM. Our experiments were performed using around 30 thousand hours of anonymized EN-US data. Results show that the proposed ltBLSTM can outperform BLSTM with up to 14.5% relative word error rate (WER) reduction. It also improves cltLSTM significantly because ltBLSTM observes longer future context than cltLSTM which is still a uni-directional model.

The rest of the paper is organized as following. In Section 2, we describe cltLSTM and how the context future information is integrated. In Section 3, we briefly introduce the standard multi-layer BLSTM models. Then we propose ltBLSTM with different designs of LT components in Section 4. We evaluate the proposed models and compare their performance in Section 5. Finally, we conclude our study in Section 6.

2. Contextual Layer Trajectory LSTM

Standard multi-layer LSTM captures temporal information by using the hidden output of the previous time step at the same layer and the hidden output of the previous layer at current time step as the input of the current time step. The l-th layer LSTM units are calculated at time step t as following.

$$\vec{i}_t^l = \sigma(\vec{W}_{xi}^l \vec{x}_t^l + \vec{W}_{hi}^l \vec{h}_{t-1}^l + \vec{p}_i^l \odot \vec{c}_{t-1}^l + \vec{b}_i^l), \quad (1)$$

$$\vec{f}_t^l = \sigma(\vec{W}_{xf}^l \vec{x}_t^l + \vec{W}_{hf}^l \vec{h}_{t-1}^l + \vec{p}_f^l \odot \vec{c}_{t-1}^l + \vec{b}_f^l), \quad (2)$$

$$\vec{c}_t^l = \vec{f}_t^l \odot \vec{c}_{t-1}^l + \vec{i}_t^l \odot \phi(\vec{W}_{xc}^l \vec{x}_t^l + \vec{W}_{hc}^l \vec{h}_{t-1}^l + \vec{b}_c^l), \quad (3)$$

$$\vec{o}_t^l = \sigma(\vec{W}_{xo}^l \vec{x}_t^l + \vec{W}_{ho}^l \vec{h}_{t-1}^l + \vec{p}_o^l \odot \vec{c}_t^l + \vec{b}_o^l), \quad (4)$$

$$\vec{h}_t^l = \vec{o}_t^l \odot \phi(\vec{c}_t^l), \quad (5)$$

where $\vec{x}_t^l$ is the input vector for the l-th layer with

$$\vec{x}_t^l = \begin{cases} \vec{h}_t^{l-1}, & if\ l > 1 \\ s_t, & if\ l = 1 \end{cases} \quad (6)$$

$l = 1 \dots L$, where L is the total number of hidden layers; s_t is the speech spectrum input at the time step t. $\vec{h}_t^l$ is the output of the time-LSTM. $\vec{W}_{x\cdot}^l$ and $\vec{W}_{h\cdot}^l$ are the weight matrices for the inputs $\vec{x}_t^l$ and the recurrent inputs $\vec{h}_{t-1}^l$, respectively. $\vec{b}^l$ are bias vectors. The vectors $\vec{i}_t^l, \vec{o}_t^l, \vec{f}_t^l, \vec{c}_t^l$ are the activations of the input, output, forget gates, and memory cells, respectively. $\vec{p}_t^l, \vec{p}_o^l, \vec{p}_f^l$ are parameter vectors associated with peephole connections. The functions σ and $\emptyset$ are the logistic sigmoid and hyperbolic tangent nonlinearity, respectively. The operation $\odot$ represents element-wise multiplication of vectors.

We define the standard forward LSTM function $LSTM(\quad)$ using Equations (1) – (5). Then the time-LSTM at time t and layer l can be simply defined as

$$\vec{h}_t^l = LSTM(\vec{h}_{t-1}^l, \vec{h}_t^{l-1}) \tag{7}$$

The layer trajectory LSTM (ltLSTM) model [10] uses time-LSTM in Equation (7) to capture temporal information and depth-LSTM to perform senone classification, respectively. It was shown to achieve better accuracy than standard time-LSTM models. The depth-LSTM is similarly defined as

$$g_t^l = LSTM(g_t^{l-1}, \vec{h}_t^l) \tag{8}$$

where g_t^{l-1} is the output of depth-LSTM at time t and layer $l-1$, and h_t^l is the output of time-LSTM at time t and layer l. The last hidden layer output from depth-LSTM, g_t^L, is then used to predict output labels.

The contextual layer trajectory LSTM (cltLSTM) model further improves ltLSTM by incorporating future context frames into the modeling [17]. The idea is to learn a fixed size vector representation of variable future frames as lookahead embedding to provide additional feature input information to the network. One simple way to incorporate future frame information is to generate an embedding vector of a context window from the depth-LSTM through a linear transform as

$$\zeta_t^{l-1} = \sum_{\delta=0}^{\tau} G_\delta^{l-1} g_{t+\delta}^{l-1} \tag{9}$$

where G_δ^{l-1} denotes the weight matrix for δth future frame, $g_{t+\delta}^{l-1}$ is the output of the depth-LSTM at time $t+\delta$ and layer $l-1$, and τ indicates how many future frames we look ahead at a layer in depth-LSTM. Then, ζ_t^{l-1} replaces g_t^{l-1} in Equation (8) to compute the depth-LSTM output g_t^l at the current layer. Equation (9) is applied to all hidden layers in our experiments. For a L layer cltLSTM which accesses τ future context frames of hidden vectors from depth-LSTM, the total number of lookahead frames is $L\tau$.

3. BSLTM

In speech recognition, it is necessary to exploit future context to obtain better recognition accuracy. Bidirectional RNNs (BRNNs) [22] can do this since they process data in both directions with two separate hidden layers, which are then concatenated and fed to the upper layer. The BLSTM model [23] combines BRNNs with LSTM to access long-range context. Its forward LSTM scans an utterance from the beginning while its backward LSTM scans from the end of the utterance. Equations (1) – (7) shows how a standard forward time-LSTM works. For the backward time-LSTM, it works similarly except that it starts the computation from the end of an utterance. The backward LSTM unit can be simply defined as

$$\overleftarrow{h}_t^l = LSTM(\overleftarrow{h}_{t+1}^l, \overleftarrow{h}_t^{l-1}) \tag{10}$$

In order to make the forward and backward time LSTMs work jointly, the input to the current layer should be modified as

$$\overleftrightarrow{h}_t^l = [\vec{h}_t^{l-1}, \overleftarrow{h}_t^{l-1}] \tag{11}$$

$$\vec{x}_t^l = \begin{cases} \overleftrightarrow{h}_t^l, & if\ l > 1 \\ s_t, & if\ l = 1 \end{cases} \tag{12}$$

where $\vec{h}_t^{l-1}$ is calculated by forward LSTM and $\overleftarrow{h}_t^{l-1}$ is by backward LSTM. In Figure 1, we show a multi-layer BLSTM network topology, where outputs from the forward LSTM and the backward LSTM are concatenated as the inputs to the upper layer.

Since both cltLSTM and BLSTM use the future context as input to improve the recognition accuracy, it would be very interesting compare them. We will provide more experimental results in Section 5.

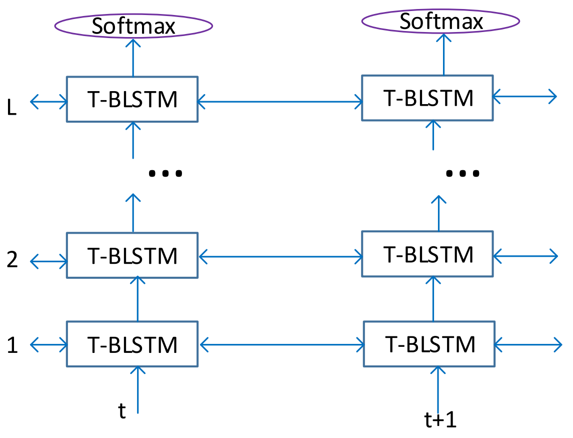

Figure 1*: Flowchart of multi-layer time-BLSTM (T-BLSTM). The outputs from the forward LSTM and the backward LSTM are concatenated as the inputs for upper layers.*

4. Layer Trajectory Designs on BLSTM

4.1. Layer trajectory BLSTM

The concept of layer trajectory (LT) is decoupling the tasks of temporal modeling and senone classification in ASR with time-LSTM and depth-LSTM respectively. Given its success on cltLSTM, we want to see how LT could help to the more powerful BLSTM models.

Here, we propose layer trajectory BLSTM (ltBLSTM) by using depth-LSTM to scan the multi-layer outputs of time-BLSTMs. The output of depth-LSTM at time t and layer l is defined as

$$g_t^l = LSTM(g_t^{l-1}, \overleftrightarrow{h}_t^l) \tag{13}$$

where $\overleftrightarrow{h}_t^l$ is the time-BLSTM's output at time t and layer l as defined in Equation (11). The depth-LSTM units at different time steps do not have any time dependency while the time recurrence is only modelled in time-BLSTM across the time axis. Hence, ltBLSTM uses time-BLSTM to capture the temporal information while doing senone classification with depth-LSTM. Its structure is shown in Figure 2.

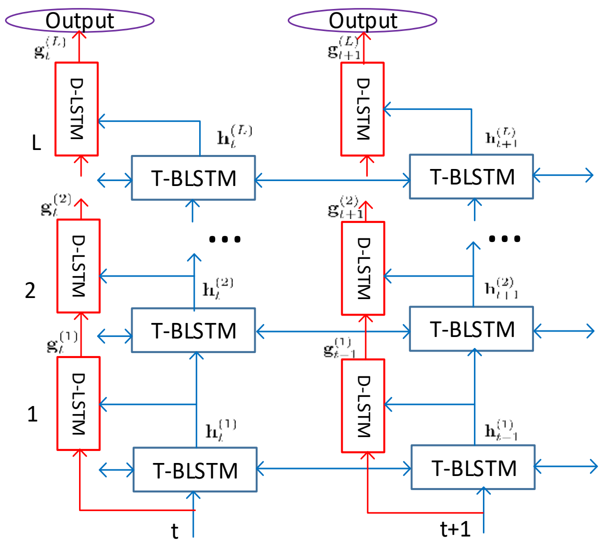

Figure 2: *Flowchart of ltBLSTM. Depth-LSTM (D-LSTM) is used to scan the outputs of time-BLSTM (T-BLSTM) across all layers at the current time step to get summarized layer trajectory information for senone classification.*

4.2. ltBLSTM with two LT components

Since BLSTM includes 2 LSTM components for both forward and backward temporal modeling, it is possible to consider applying separate LTs on the forward and backward LSTMs. We then investigate using two depth-LSTMs in ltBLSTM and each of them is coupled with one directional LSTM component in BLSTM model as shown in Figure 3. Compared to the ltBLSTM which only uses a single depth LSTM at each layer in Section 4.1, the only change is that now we use two depth-LSTMs at each layer.

Figure 3 shows two design options. Instead of using BLSTM outputs as inputs for a single depth-LSTM model as defined in Equation (13), in the first option (Figure 3(a)) the outputs from forward and backward LSTM are fed as inputs to two separate depth-LSTMs as

$$\vec{g}_t^l = LSTM(\vec{g}_t^{l-1}, \vec{h}_t^l) \tag{14}$$

$$\overleftarrow{g}_t^l = LSTM(\overleftarrow{g}_t^{l-1}, \overleftarrow{h}_t^l) \tag{15}$$

where $\vec{g}_t^l$ and $\overleftarrow{g}_t^l$ are the outputs of forward and backward depth-LSTMs at time t and layer l, and $\vec{h}_t^l$ and $\overleftarrow{h}_t^l$ are the outputs of forward and backward time-LSTM at time t and layer l, respectively. Finally the outputs of layer L from depth-LSTMs are $[\vec{g}_t^L, \overleftarrow{g}_t^L]$ are concatenated to predict the output senone labels.

The second option is shown in Figure 3(b), in which the outputs of depth-LSTMs at each layer are concatenated as the input to both depth-LSTMs at the next layer.

$$\vec{g}_t^l = LSTM(\vec{\bar{g}}_t^{l-1}, \vec{h}_t^l) \tag{16}$$

$$\overleftarrow{g}_t^l = LSTM(\vec{\bar{g}}_t^{l-1}, \overleftarrow{h}_t^l) \tag{17}$$

$$\vec{\bar{g}}_t^{l-1} = \begin{cases} [\vec{g}_t^{l-1}, \overleftarrow{g}_t^{l-1}], & if\ l > 1 \\ s_t\,, & if\ l = 1 \end{cases} \tag{18}$$

where $\vec{g}_t^{l-1}$ and $\overleftarrow{g}_t^{l-1}$ are the outputs of depth-LSTMs at time t and layer $l-1$, and s_t is the speech spectrum input at the time step t as defined in Equation (6).

We will compare the model performance of different LT designs in Section 5.

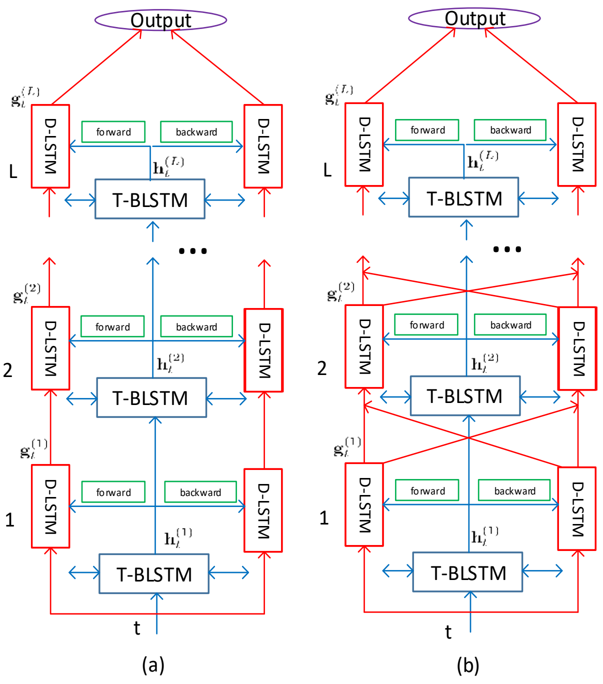

Figure 3: *Two LT design for BLSTM models; (a): depth-LSTM combination only occurs at the output layer; (b): depth-LSTM combination occurs at each hidden layer.*

5. Experiments

In this section, we evaluate the performance of LSTM, cltLSTM, BLSTM, and ltBLSTM. All models were trained with 30 thousand hours of anonymized and transcribed Microsoft production data, including Cortana and Conversational data which is a mixture of close-talk and far-field speech recorded from a variety of devices. All time-LSTMs and depth-LSTMs in cltLSTM use 1024 hidden units and the output of each LSTM layer is reduced to 512 using a linear projection layer. The time forward and backward LSTMs in BLSTM models use 800 hidden units and their outputs at each layer are reduced to 400 by a linear projection layer. The depth-LSTMs in ltBLSTM models use 800 hidden units that are projected to 400 dimensional outputs in the standard one LT design and use 400 hidden units if two LT components are used. All models have 6 hidden layers, which experimentally we found is the best compromise between computational cost and performance. In addition, more hidden layers may lead to model divergence issue [10]. The cltLSTM has 4 frames lookahead at each layer, hence 24 frames lookahead in total. We reduce the latency of BLSTM and ltBLSTM models by using the concept of latency control with 40 frames lookahead [24]. The SoftMax layer has 9404 nodes to model the senone labels. The target senone label is delayed by 5 frames as in [25]. The backpropagation through time (BPTT) [26] with truncation size 40 is used to train all cltLSTM models. The input feature is 80-dimension log Mel filter bank for every 10 milliseconds (ms) speech. We applied frame skipping by a factor of 2 [8] to reduce the runtime cost. A 5-gram language model with around 1 million (M) vocabularies and 100 M n-grams was applied in all our experiment evaluations.

All cross entropy (CE) trained models were evaluated in our experiments with Microsoft Cortana and Conversation test sets, including both close-talk and far-field recordings with 439k and 111k words, respectively. Later we also trained sequence based discriminative models and obtained the similar performance improvement pattern as in [13], but that is not the focus of this study. Compared to the Conversation test set that has longer utterances from conversations, the Cortana test set has shorter utterances of voice search and commands. In addition, the models were also evaluated on a third test set named as DMA with 29k words, which is a Microsoft internal meeting test set from the different domain than Cortana or Conversation data and is not covered well in our training data. Therefore, the DMA test set measures the generalization capability of our models.

Table 1: *WERs of LSTM, cltLSTM, BLSTM and ltBLSTM on Cortana, Conversation, and DMA test sets*

Model	Cortana	Conversation	DMA
LSTM	9.85	19.20	20.19
cltLSTM chunk size =16	8.78	16.51	15.58
cltLSTM chunk size =40	8.01	15.62	14.68
BLSTM	8.18	15.64	14.98
ltBLSTM	7.73	14.39	12.81

Table 1 shows WERs of LSTM, cltLSTM, BLSTM and ltBLSTM models. In our previous study [17], we used truncated backpropagation through time to train cltLSTM models. The truncation size is 16 in order to speed up training process. That means, in a chunk window, the left most frame was able to access at most 15 future frames even though we designed the model structure to have 24 frames lookahead. This undermines the performance of cltLSTM models significantly. In this paper, we increased the training chunk window size from 16 to 40. This solution is still not perfect in theory, it represents however a better tradeoff between training efficiency and accuracy. Actually, the chunk window size 80 was also tried in our experiments, and we didn't get significantly better results. The cltLSTM model with chunk window size 40 achieved 8.01%, 15.62%, and 14.68% WERs on Cortana, Conversation, and DMA test sets, respectively, which corresponds to relative 8.8%, 5.4%, and 5.8% WER reductions from the baseline cltLSTM with the training chunk window size 16 in our previous research. This is much better than the performance of baseline LSTM.

Table 2: *WERs of different LT designs on Cortana, Conversation, and DMA test sets*

Model	Cortana	Conversation	DMA
ltBLSTM (1LT)	7.73	14.39	12.81
ltBLSTM (2LT)	7.74	14.62	13.38
ltBLSTM (2LT-concat)	7.58	14.44	12.75

Since both cltLSTM and BLSTM use future context frames for modeling, we also compare their recognition performance in Table 1. Overall, cltLSTM obtained slightly lower WERs than BLSTM. On Cortana, Conversation, and DMA test sets, cltLSTM model got relative 2.1%, 0.1%, and 2.0% WER reductions over BLSTM model. This is because cltLSTM benefits both LT modeling and access to future information while BLSTM only has the access of future information. After integrating LT modeling into BLSTM using the method in Section 4.1 which only has one depth-LSTM at each layer, the ltBLSTM model clearly outperformed cltLSTM in Table 1. The ltBLSTM model obtained WERs 7.73%, 14.39%, and 12.81% on Cortana, Conversation, and DMA test sets, representing relative 5.5%, 8.0%, and 14.5% WER reductions over the BLSTM baseline across these three test sets respectively. It also improved cltLSTM with relative 3.5%, 7.9%, and 12.7% WER reductions on these three test sets, respectively. Finally, ltBLSTM model size is around 1/3 larger than the baseline BLSTM model without increasing any recognition latency at runtime. However, there were extra computational cost in both training and test due to enlarged model size. We didn't compare the performance between equal sized ltBLSTM and BLSTM models. However, our past observations suggest that simply increasing the BLSTM model size would not yield such significant performance improvement.

In Table 2, we compare the ltBLSTM model performance among different LT designs discussed in Section 4. The ltBLSTM in Section 4.1 is denoted as ltBLSTM (1LT). Using two depth-LSTM components and combining their outputs only at the last hidden layer as shown in Figure 3(a), the ltBLSTM (2LT) model got WERs 7.74%, 14.62%, and 13.38% on Cortana, Conversation, and DMA test sets, which is slight worse than the ltBLSTM (1LT) model . However, if two depth-LSTMs' outputs are concatenated at each hidden layer as shown in Figure 3(b), the ltBLSTM (2LT-concat) model obtained WERs 7.58%, 14.44%, and 12.75% across these three test sets, which is relative 1.9%, -0.3%, and 0.5% WER reductions over one depth-LSTM baseline. Actually, since the lower layer outputs of forward and backward LSTMs are concatenated as inputs to upper layer forward and backward LSTMs in BLSTM, the two separated depth-LSTMs get redundant input information from lower BLSTM layers as shown in Figure 3. Therefore, the options of using two depth-LSTM has no obvious advantage over the baseline using one depth-LSTM. In other words, the design of using one depth-LSTM at each layer is very effective and robust for both uni-directional LSTM and BLSTM models.

6. Conclusions

In this paper, we proposed a novel model called ltBLSTM which scans the outputs of the multi-layer time-BLSTM with a depth-LSTM to learn layer trajectory information for senone classification. By decoupling the tasks of temporal modeling and target classification with time-BLSTM and depth-LSTM, this model allows individual LSTM units to focus on their own tasks. Trained with 30k hours of Microsoft internal speech data, the proposed ltBLSTM achieved relative 5.5%, 8.0%, and 14.5% WER reduction from the BLSTM baseline. The larger improvement on DMA test set also indicates LT component significantly improved model generalization capacity on unseen test sets. Finally, we investigated different LT designs of ltBLSTM models and found that the design with one depth-LSTM at each layer is effective enough for ltBLSTM models to achieve good recognition performance.

7. References

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in Proc. ICASSP, 2013, pp. 6645–6649.

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

[5] H. Sak, O. Vinyals, G. Heigold, A. Senior, E. McDermott, R. Monga, and M. Mao, "Sequence discriminative distributed training of long short-term memory recurrent neural networks," in Proc. Interspeech, 2014.

[6] X. Li and X. Wu, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition," in Proc. ICASSP, 2015, pp. 4520–4524.

[7] Y. Miao and F. Metze, "On speaker adaptation of long short-term memory recurrent neural networks.," in Proc. Interspeech, 2015, pp. 1101–1105.

[8] Y. Miao, J. Li, Y. Wang, S. Zhang, and Y. Gong, "Simplifying long short-term memory acoustic models for fast training and decoding," in Proc. ICASSP, 2016.

[9] D. Yu and J. Li, "Recent progresses in deep learning based acoustic models," IEEE/CAA J. of Autom. Sinica., vol. 4, no. 3, pp. 399–412, July 2017.

[10] J. Li, C. Liu, and Y. Gong, "Layer trajectory LSTM," in Proc. Interspeech, 2018.

[11] Y. Zhao, S. Xu, and B. Xu, "Multidimensional residual learning based on recurrent neural networks for acoustic modeling," in Proc. Interspeech, 2016, pp. 3419–3423.

[12] J. Kim, M. El-Khamy, and J. Lee, "Residual LSTM: Design of a deep recurrent architecture for distant speech recognition," arXiv preprint arXiv:1701.03360, 2017.

[13] J. Li, L. Lu, C. Liu, and Y. Gong, "Exploring layer trajectory LSTM with depth processing units and attention," in Proc. IEEE SLT, 2018.

[14] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," arXiv preprint arXiv:1302.4389, 2013.

[15] Y. Miao, F. Metze, and S. Rawat, "Deep maxout networks for low-resource speech recognition," in Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on. IEEE, 2013, pp. 398–403.

[16] P. Swietojanski, J. Li, and J. Huang, "Investigation of maxout networks for speech recognition," in Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE, 2014, pp. 7649–7653.

[17] J. Li, Liang Lu, C. Liu, and Y. Gong, "Improving layer trajectory LSTM with future context frames," in Proc. ICASSP, 2018.

[18] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673–2681, 1997.

[19] H. Sak, A. Senior, K. Rao, O. Irsoy, A. Graves, F. Beaufays, J. Schalkwyk, "Learning acoustic frame labeling for speech recognition with recurrent neural networks," ICASSP, 2015.

[20] H. Jaeger, *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach*, vol. 5, GMD-Forschungszentrum Informationstechnik Bonn, 2002.

[21] M. Schuster and K. K. Paliwal, "Bidirectional Recurrent Neural Networks," IEEE Transactions on Signal Processing, vol. 45, pp. 2673–2681, 1997.

[22] M. Schuster and K. K. Paliwal, "Bidirectional Recurrent Neural Networks," IEEE Transactions on Signal Processing, vol. 45, pp. 2673–2681, 1997.

[23] A. Graves and J. Schmidhuber, "Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures," Neural Networks, vol. 18, no. 5-6, pp. 602–610, June/July 2005.

[24] Y. Zhang, G. Chen, D. Yu, K. Yao, S. Khudanpur, and J. Glass, "Highway long short-term memory RNNs for distant speech recognition," ICASSP, 2016.

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

[26] H. Jaeger, *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach*, vol. 5, GMD-Forschungszentrum Informationstechnik Bonn, 2002.