CONSTRUCTING LONG SHORT-TERM MEMORY BASED DEEP RECURRENT NEURAL NETWORKS FOR LARGE VOCABULARY SPEECH RECOGNITION

Xiangang Li, Xihong Wu

Speech and Hearing Research Center,
Key Laboratory of Machine Perception (Ministry of Education),
Peking University, Beijing, 100871
{lixg, wxh}@cis.pku.edu.cn

ABSTRACT

Long short-term memory (LSTM) based acoustic modeling methods have recently been shown to give state-of-the-art performance on some speech recognition tasks. To achieve a further performance improvement, in this research, deep extensions on LSTM are investigated considering that deep hierarchical model has turned out to be more efficient than a shallow one. Motivated by previous research on constructing deep recurrent neural networks (RNNs), alternative deep LSTM architectures are proposed and empirically evaluated on a large vocabulary conversational telephone speech recognition task. Meanwhile, regarding to multi-GPU devices, the training process for LSTM networks is introduced and discussed. Experimental results demonstrate that the deep LSTM networks benefit from the depth and yield the state-of-the-art performance on this task.

Index Terms— long short-term memory, recurrent neural networks, deep neural networks, acoustic modeling, large vocabulary speech recognition

1. INTRODUCTION

Recently, the context dependent (CD) deep neural network (DNN) hidden Markov model (HMM) (CD-DNN-HMM) has become the dominant framework for acoustic modeling in speech recognition (e.g. [1][2][3][4]). However, given that speech is an inherently dynamic process, some researchers pointed out that recurrent neural networks (RNNs) can be considered as alternative models for acoustic modeling [5]. The cyclic connections in RNNs exploit a self-learnt amount of temporal context, which makes RNNs better suited for sequence modeling tasks. Unfortunately, in practice, conventional RNNs are hard to be trained properly due to the vanishing gradient and exploding gradient problems as described in [6]. To address these problems, literature [7] proposed an elegant RNN architecture, called as long short-term memory (LSTM).

LSTMs and conventional RNNs have been successfully used for many sequence labeling and sequence prediction tasks. In language modeling, RNNs were used as generative models over word sequences, and remarkable improvements were achieved [8] over the standard n-gram models. For handwriting recognition, LSTM networks have been applied for a long time [9], in which, the bidirectional LSTM (BLSTM) networks trained with connectionist temporal classification (CTC)[10] has been demonstrated performing better than the HMM-based system. In speech synthesis, the BLSTM network has also been applied and a notable improvement was obtained [11]. For language identification, LSTM based approach was proposed in [12] to compared with i-vector and DNN systems, and better performance was achieved. Recently, LSTM networks have also been introduced on phoneme recognition task [5], robust speech recognition task [13], and large vocabulary speech recognition task [14][15][16], and shown state-of-the-art performances. Subsequently, the sequence discriminative training of LSTM networks is investigated in [17], and a significant gain was obtained.

In the researches of acoustic modeling, depth for feed-forward neural networks can lead to more expressive models. LSTMs and conventional RNNs are inherently deep in time, for they can be expressed as a composition of multiple nonlinear layers when unfolded in time. This paper explores the depth of LSTMs, which is defined as the depth in space. Based on earlier researches on constructing deep RNNs [18], in this work, possible approaches are explored to extend LSTM networks into deep ones, and various deep LSTM networks are empirically evaluated and compared on a large vocabulary Mandarin Chinese conversational telephone speech recognition task. Although lots of attentions have been attracted to the deep LSTM networks, this paper summaries the approaches of constructing deep LSTM networks from different perspectives, and suggests alternative architectures that can yield comparable performance.

2. CONSTRUCTING LSTM BASED DEEP RNNS

2.1. The conventional LSTM architecture

Given an input sequence $x = (x_1, x_2, \ldots, x_T)$, a conventional RNN computes the hidden vector sequence $h = (h_1, h_2, \ldots, h_T)$ and output vector sequence $y = (y_1, y_2, \ldots, y_T)$ from $t = 1$ to T as follows:

$$h_t = \mathcal{H}(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (1)$$

$$y_t = W_{hy}h_t + b_y \quad (2)$$

where, the W denotes weight matrices, the b denotes bias vectors and $\mathcal{H}(\cdot)$ is the recurrent hidden layer function.

In the LSTM architecture, the recurrent hidden layer consists of a set of recurrently connected subnets known as "memory blocks". Each memory block contains one or more self-connected memory cells and three multiplicative gates to control the flow of information. In each LSTM cell, the flow of information into and out of the cell is guarded by the learned input and output gates. Later, in order to provide a way for the cells to reset themselves, the forget gate was added [19]. In addition, the modern LSTM architecture contains peephole weights connecting the gates to the memory cell, which improve the LSTM's ability to learn tasks that require precise timing and counting of the internal states [20]. As illustrated in Fig. 1, the

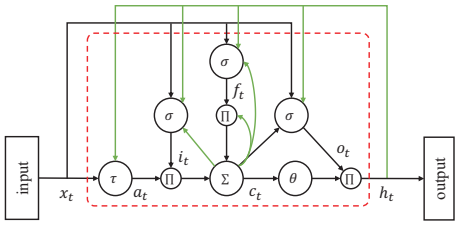

Fig. 1. The architecture of a LSTM network with one memory block, where green lines are time-delayed connections.

recurrent hidden layer function $\mathcal{H}$ for this version of LSTM networks is implemented as following:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (3)$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (4)$$
$$a_t = \tau(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (5)$$
$$c_t = f_t c_{t-1} + i_t a_t \quad (6)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (7)$$
$$h_t = o_t \theta(c_t) \quad (8)$$

Where, σ is the logistic sigmoid function, and i, f, o, a and c are respectively the input gate, forget gate, output gate, cell input activation, and cell state vectors, and all of which are the same size as the hidden vector h. W_{ci}, W_{cf}, W_{co} are diagonal weight matrices for peephole connections. τ and θ are the cell input and cell output non-linear activation functions, generally in this paper $tanh$.

2.2. Deep LSTM networks

A number of theoretical results support that a deep, hierarchical model can be more efficient at representing some functions than a shallow one [21]. This paper is focused on constructing deep LSTM networks.

In [18], the architecture of conventional RNNs is carefully analyzed, and from three points, an RNN can be deepened: (1) input-to-hidden function, (2) hidden-to-hidden transition and (3) hidden-to-output function. In this paper, from these three points and the stacked LSTMs, several novel architectures to extend LSTM networks to deep ones are introduced as follows. For convenience, a simplified illustration of the LSTM is shown in Fig. 2(a) firstly.

2.2.1. Deep hidden-to-hidden transition

In [18], an RNN with deep transition is discussed for increasing the depth of the hidden-to-hidden transition. Thus, two architectures can be obtained, as illustrated in Fig. 2(b) and Fig. 2(c). In details, in the architecture shown in Fig. 2(b), a multiple layer transformation is added before the cell input activation, and which means that the calculation of a_t in equation (5) is changed as:

$$a_{0,t} = \phi_0(W_{0,x}x_t + W_{0,h}h_{t-1} + b_0) \quad (9)$$
$$a_t = \phi_L(W_L\phi_{L-1}(\dots\phi_1(W_1 a_{0,t} + b_1)) + b_L) \quad (10)$$

Where, ϕ is the activation function. In this paper, this architecture is called as the *LSTM with input projection layer* (LSTM-IP for short).

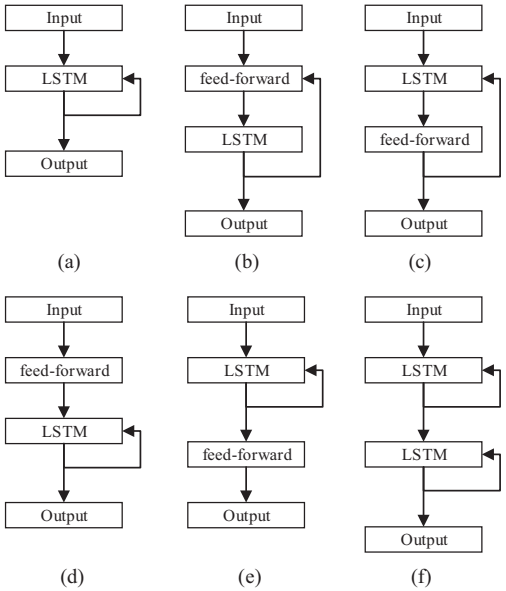

Fig. 2. Illustrations of different strategies for constructing LSTM based deep RNNs. (a) a conventional LSTM; (b) a LSTM with input projection; (c) a LSTM with output projection. (d) a LSTM with deep input-to-hidden function; (e) a LSTM with deep hidden-to-output function; (f) stacked LSTMs

Another architecture, shown in Fig. 2(c), has a separate hidden layer after the LSTM memory blocks, and the output activation h_t is changed as p_t

$$p_t = \phi_L(W_L\phi_{L-1}(\dots\phi_0(W_0 h_t + b_0)) + b_L) \quad (11)$$

We call this architecture as the *LSTM with output projection layer* (LSTM-OP for short). However, this architecture is proposed earlier in literature [15] to address the computation complexity of learning a LSTM network, in which it is called as the LSTM projected. From the perspective of this paper, this architecture is considered as a way to increase the depth of the hidden-to-hidden transition, although it may further be beneficial in tackling computation complexity issue.

It should be noticed that, in the LSTM-OP architecture, linear activation units can be used in projection layer, just like literature [15] suggested. By contrast, there must be a non-linear activation (e.g. $tanh$) units used in the projection layer in the LSTM-IP.

2.2.2. Deep input-to-hidden function

A typical way to make the input-to-hidden function deep is using higher-level representations of DNNs as the input for RNNs. Literature [22] reported that a better phoneme recognition performance could be achieved by applying this strategy for RNNs.All the previous studies are based on conventional RNNs, and in this research, this method is adopted for constructing deep LSTM networks as illustrated in Fig. 2(d), and applied to a large vocabulary speech recognition task.

2.2.3. Deep hidden-to-output function

It was discussed that a deep hidden-to-output function can be useful to disentangle the factors of variations in the hidden state [18]. Based

on this view, we construct a deep LSTM network shown in Fig. 2(e) by adding some intermediate layers between the output of the LSTM and the softmax layer.

2.2.4. *Stack of LSTMs*

Perhaps, the most straight-forward way to construct the deep LSTM network is to stack multiple LSTM layers on top of each other. Specifically, output h_t from the lower LSTM layer, is the input x_t of the upper LSTM layer. This stacked LSTM networks can combine the multiple levels or representations with flexible use of long range context, and was introduced for acoustic modeling in speech recognition in [5], which showed that a significant performance improvement can be obtained compared with the shallow one.

3. GPU IMPLEMENTATION

We implement the LSTM network training on multi-GPU devices. In the training procedure, the truncated back-propagation though time (BPTT) learning algorithm [23] is adopted. Each sentence in the training set is split into subsequences with equal length T_{bptt} (e.g. 15 frames). As illustrated in Fig. 3, two adjacent subsequences have overlapping frames $T_{overlap}$ (e.g. 5 frames). The gradients are computed for each subsequence and back-propagated to its start. For computational efficiency, one GPU operates in parallel on N (e.g. 20) subsequences from different utterances at a time. After the GPU has updated the parameters in the LSTM networks, it continues with the next N subsequences in these utterances.Besides, in order to train these networks on multi-GPU devices, asynchronous stochastic gradient descent (ASGD) [24][25] is adopted.

In our experiments, it took us about two days to train a shallow conventional LSTM network having 750 cells with four GPU devices on a 150-hour speech corpus, where training a LSTM layer took around two to five times as much time as the training for a full-connection feed-forward hidden layer.

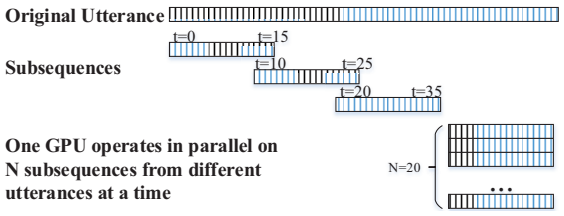

Fig. 3. Illustration of GPU implementation.

4. EXPERIMENTS

We evaluate these LSTM networks on a large vocabulary speech recognition task - the HKUST Mandarin Chinese conversational telephone speech recognition [26]. The corpus (LDC2005S15, LDC2005T32) is collected and transcribed by Hong Kong University of Science and Technology (HKUST), which contains 150-hour speech, and 873 calls in the training set and 24 calls in the development set, respectively. In our experiments, around 10-hour speech was randomly selected from the training set, used as the validate set for network training, and the original development set in the corpus was used as speech recognition test set, which is not used in the training or the hyper-parameters determination procedures.

4.1. Experimental setup

The speech in the dataset is represented with 25ms frames of Mel-scale log-filterbank coefficients (including the energy value), along with their first and second temporal derivatives. In the experiments, the feed-forward DNNs used the concatenated features, which were produced by concatenating the current frame with 5 frames in its left and right context. However, for the inputs of LSTM networks, only current features (no context) were used.

A trigram language model was used in all the experiments, which was estimated using all the transcriptions of the acoustic model training set. We use a hybrid approach [4][15] for acoustic modeling with LSTM networks or DNNs, in which the neural networks' outputs are converted as pseudo likelihood as the state output probability in the HMM framework. All the networks were trained based on the alignments generated by a well-trained GMM-HMM systems with 3304 tied context dependent HMM states (realignments by DNNs were not performed), and only the cross-entropy objective function was used for all networks.

For network training, the learning rate was decreased exponentially. We tried to set the initial and final learning rates specific to a network architecture for stable convergence of each network. In the experiments, the initial learning rates ranged from 0.0005 to 0.002, and each final learning rate was always set as one-tenth of the corresponding initial one. In the training procedure of LSTM networks, the strategy introduced in [27] was applied to scale down the gradients. Besides, since the information from the future frames helps making LSTM networks better decisions for current frame, we also delayed the output HMM state labels by 3 frames.

4.2. Experimental results

Firstly, the baseline performance is summarized in Table 1. For training the Subspace GMM [28], KALDI toolkit [29] was used. All the DNNs in the experiments had 4 hidden layers. Each layer in the "ReLU DNN" model had 2000 ReLU units [30]. Each layer in the "PNorm DNN" model had 800 pnorm units [31], where the hyper-parameter p is set to 2, and the group size is set to 8. The "Conv DNN" model had two convolutional layers (along with max-pooling) and three ReLU layers. It can be found out that, the character error rates (CER) of baseline GMM-HMM and DNN-HMM are comparable with those reported in [32][33][34].

Table 1. Speech recognition results of baseline systems on the HKUST Mandarin Chinese conversational telephone speech recognition task.

Model Descriptions	CER(%)
GMM	48.68
Subspace GMM	44.29
ReLU DNN	38.42
PNorm DNN	38.01
Conv DNN	37.13

Experiments were conducted to evaluate these deep LSTM networks shown in Fig. 2. In the training procedure of these LSTM networks, the T_{bptt} was fixed on 15, $T_{overlap}$ was fixed on 5. Four GPUs were used, and each GPU operated in parallel on 20 subsequences at a time.

The LSTM-IP network in the experiment had 750 LSTM cells and a non-linear activation projection layer with 2000 $tanh$ units.

The LSTM-OP network in the experiment had 2000 LSTM cells and a linear activation projection layer with 750 nodes.

In order to construct a LSTM network with deep input-to-hidden function, we constructed a LSTM network by putting a LSTM network on three feed-forward intermediate layers, and each feed-forward layer had 2000 ReLU units. This network is indicated as "3-layer ReLU + LSTM" in Table 2. Similarly, we trained a model indicated as "2-layer Conv + 2-layer ReLU + LSTM". For the deep hidden-to-output function, a LSTM network, indicated as "LSTM + 3-layer ReLU" in Table 2, was constructed by adding three feed-forward intermediate hidden layers on top of the LSTM layer, and each feed-forward hidden layer had 2000 ReLU units. The stacked LSTMs network was also evaluated, in which, three conventional LSTMs were stacked, and each layer had 750 LSTM cells. These three networks were trained using the discriminative pre-training algorithm [35]. Concretely, in the training procedure of "3-layer ReLU + LSTM", three ReLU hidden layers were firstly pre-trained, and then the original output softmax layer was replaced by a new random initialized LSTM layer along with a new output softmax layer. Finally, the whole network was jointly optimized.

Table 2. Speech recognition results of different strategies of constructing deep LSTM networks.

Model Descriptions	CER(%)
LSTM	40.28
LSTM-IP	39.09
LSTM-OP	35.92
3-layer ReLU + LSTM	37.31
2-layer Conv + 2-layer ReLU + LSTM	36.66
LSTM + 3-layer ReLU	37.16
Stack of LSTM (3-layer)	**35.91**

Comparing these results listed in Table 2 with the baseline, the performance of 1-layer conventional LSTM network is even worse than the feed-forward DNNs. Through making deep hidden-to-hidden transitions, obvious performance improvements can be obtained, especially the LSTM-OP. Besides, the performance can also been improved by making deep input-to-hidden and hidden-to-output functions. It should be noted that, the LSTM-OP can yield comparable performance with the stacked LSTMs, which reached a similar conclusion with that in [15].

It is possible to design and train deeper variant of a LSTM network that combines different methods in Fig 2 together. For instance, a stacked LSTM-OPs network may be constructed by combining the deep hidden-to-hidden transition and the stack of LSTMs. Combining different methods in Fig 2 is a potential way to further improve the performance. Thus, experiments were conducted to evaluate some selected combinations of these methods for constructing deep LSTM networks, where each hidden layer had the same configuration as that in the experiments described above. The results are listed in Table 3, and the best performance can be obtained by combining the LSTM-OP and deep hidden-to-output function.

From these results in Table 3, we can find out that, the performance can be further improved by stacking LSTM-IPs and the LSTM-OPs. However, the network, that had LSTM-OP layer on top of three feed-forward intermediate layers, yielded worse performance than the LSTM-OP network, which needed to be further researched. What is noteworthy is that the network that had three full-connection hidden layers on top of LSTM-OP layer yielded the best performance, and required less computations than the stacked LSTM-OPs network in both training and testing procedures.

Table 3. Speech recognition results of selected combinations for constructing deep LSTM networks.

Model Descriptions	CER(%)
3-layer ReLU + LSTM-OP	36.73
2-layer Conv + 2-layer ReLU + LSTM-OP	36.15
LSTM-OP + 3-layer ReLU	**34.65**
Stack of LSTM-IP (3-layer)	35.00
Stack of LSTM-OP (3-layer)	34.84

These experimental results had revealed that deep LSTM networks benefit from the depth. Compared with the shallow LSTM network, a 13.98% relatively CER reduction can be obtained. Compared with the feed-forward DNNs, the deep LSTM networks can reduce the CER from 38.01% to 34.65%, which is a 8.87% relatively CER reduction.

5. DISCUSSION AND CONCLUSIONS

In this paper, we have explored novel approaches to construct long short-term memory (LSTM) based deep recurrent neural networks (RNNs). A number of theoretical results support that a deep, hierarchical model can be more efficient at representing some functions than a shallow one [21]. This paper is focused on constructing deep LSTM networks, which have been shown to give state-of-the-art performance for acoustic modeling on some speech recognition tasks. Inspired from the discussion about how to construct deep RNNs in [18], several alternative architectures were constructed for deep LSTM networks from three points: (1) input-to-hidden function, (2) hidden-to-hidden transition and (3) hidden-to-output function. Furthermore, in this paper, some deeper variants of LSTMs were also designed by combining different points.

In this work, these LSTM network training were implemented on multi-GPU devices, in which the truncated BPTT learning algorithm was adopted, and the experiments discovered that the LSTM RNNs can also be quickly trained on GPU devices.

We empirically evaluated various deep LSTM networks on a large vocabulary Mandarin Chinese conversational telephone speech recognition task. The experiments revealed that constructing deep LSTM architecture outperformed the standard shallow LSTM networks and DNNs. Besides, the LSTM-OP followed with three feed-forward intermediate layers outperformed the stacked LSTM-OPs.

However, we believe that this work is just a preliminary study on how to construct deep LSTM networks. There are many efforts need to be done about the architectures of LSTM networks. Some other architectures will be explored and evaluated in our future work, such as a LSTM-IP network which has three non-linear activation projection layers, a stacked LSTMs network followed with multiple feed-forward intermediate layers, a LSTM network with both input and output project layers, and deep architectures with the maxout unit improved LSTM networks [36].

6. ACKNOWLEDGEMENTS

The work was supported in part by the National Basic Research Program (2013CB329304), the research special fund for public welfare industry of health(201202001), and National Natural Science Foundation (No.61121002, No.91120001).

7. REFERENCES

[1] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio Speech Lang. Processing*, vol. 20, pp. 14–22, 2012.

[2] G. Hinton, L. Deng, D. Yu, and et al., "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *IEEE Signal Processing Mag*, vol. 29, pp. 82–97, 2012.

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

[4] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pretrained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio Speech Lang. Processing*, vol. 20, pp. 30–42, 2012.

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

[6] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Networks*, vol. 5, pp. 157–166, 1994.

[7] S. Hochreiter and J. Schimidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[8] T. Mikolov, M. Marafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Interspeech*, 2010, pp. 1045–1048.

[9] A. Graves, M. Liwichi, S. Fernández, and et al., "A novel connnectionist system for unconstrained handwriting recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 855–868, 2009.

[10] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural network," in *ICML*, 2006, pp. 369–376.

[11] Y. Fan, Y. Qian, F. Xie, and F.K. Soong, "Tts synthesis with bidirectional lstm based recurrent neural networks," in *Interspeech*, 2014, pp. 1964–1948.

[12] J. Gonzalez-Dominguez, I. Lopez-Moreno, H. Sak, and et al., "Automatic language identification using long short-term memory recurrent neural networks," in *Interspeech*, 2014, pp. 2155–2159.

[13] J. Geiger, X. Zhang, F. Weninger, and et al., "Robust speech recognition using long short-term memory recurrent neural networks for hybrid acoustic modelling," in *Interspeech*, 2014, pp. 631–635.

[14] A. Graves, N. Jaitly, and A. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *ASRU*, 2013, pp. 273–278.

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

[16] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," arXiv:1402.1128, 2014.

[17] H. Sak, O. Vinyals, G. Heigold, and et al., "Sequence discriminative distributed training of long short-term memory recurrent neural networks," in *Interspeech*, 2014, pp. 1209–1213.

[18] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," arXiv:1312.6026, 2013.

[19] F. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural Computation*, vol. 12, pp. 2451–2471, 2000.

[20] F. Gers, N. Schraudolph, and J. Schmidhuber, "Learning precise timing with lstm recurrent networks," *Journal of Machine Learning Research*, vol. 3, pp. 115–143, 2003.

[21] Y. Bengio, "Learning deep architecture for ai," *Found. Trends Mach. Learn.*, vol. 2, pp. 1–127, 2009.

[22] L. Deng and J. Chen, "Sequence classification using the high-level features extracted from deep neural networks," in *ICASSP*, 2014, pp. 6894–6898.

[23] R. Williams and J. Peng, "An efficient gradient-based algorithm for online training of recurrent neural network trajectories," *Neural Computation*, vol. 2, pp. 490–501, 1990.

[24] R. Ormándi, I. Hegedüs, and M. Jelasity, "Asynchronous peer-to-peer data mining with stochastic gradient descent," *Lecture Notes in Computer Science*, pp. 528–540, 2011.

[25] S. Zhang, C. Zhang, Z. You, R. Zheng, and B. Xu, "Asynchronous stochastic gradient descent for dnn training," in *ICASSP*, 2013, pp. 6660–6663.

[26] Y. Liu, P. Fung, Y. Yang, and et al., "Hkust/mts: A very large scale mandarin telephone speech corpus," in *ISCSLP*, 2006, pp. 724–735.

[27] R. Pascanu and Y. Bengio, "On the difficulty of training recurrent neural networks," arXiv:1211.5063, 2012.

[28] D. Povey, L. Burget, M. Agarwal, and et al., "The subspace gaussian mixture model - a structured model for speech recognition," *Computer Speech & Language*, vol. 25, pp. 404–439, 2011.

[29] D. Povey, A. Ghoshal, L. Burget, and et al., "The kaldi speech recognition toolkit," in *ASRU*, 2011, pp. 1–4.

[30] M. Zeiler, M. Ranzato, R. Monga, and et al., "On rectified linear units for speech processing," in *ICASSP*, 2013, pp. 3517–3521.

[31] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *ICASSP*, 2014, pp. 215–219.

[32] C. Weng and B. Juang, "Adaptive boosted non-uniform mce for keyword spotting on spontaneous speech," in *ICASSP*, 2013, pp. 6960–6964.

[33] C. Ni, N. Chen, and B. Ma, "Multiple time-span feature fusion for deep neural network modeling," in *ISCSLP*, 2014, pp. 138–142.

[34] Y. Liu, X. Li, and X. Wu, "Margin-based discriminative pronunciation modeling for large vocabulary mandarin speech recognition," in *SLT*, 2014.

[35] D. Yu, L. Deng, G. Li, and F. Seide, "Discriminative pretraining of deep neural networks," U.S. Patent Filling, 2011.

[36] X. Li and X. Wu, "Improving long short-term memory networks using maxout units for large vocabulary speech recognition," in *ICASSP*, 2015.