Unidirectional Neural Network Architectures for End-to-End Automatic Speech Recognition

Niko Moritz, Takaaki Hori, Jonathan Le Roux

Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA

moritz@merl.com, thori@merl.com, leroux@merl.com

Abstract

In hybrid automatic speech recognition (ASR) systems, neural networks are used as acoustic models (AMs) to recognize the distinctive sounds of speech, i.e., phonemes, that are composed to words and sentences using pronunciation dictionaries, hidden Markov models, and language models, which can be jointly represented by a weighted finite state transducer (WFST). The importance of capturing temporal context by an AM has been well studied and discussed in prior work. In an end-to-end ASR system, however, all components are merged into a single neural network, i.e., the breakdown into an AM and the different parts of the WFST model is no longer possible. This implies that neural network architectures used for end-to-end ASR have even stronger requirements for processing long contextual information. Bidirectional long short-term memory (BLSTM) neural networks have demonstrated state-of-the-art results in end-to-end ASR but are unsuitable for streaming applications. Latency-controlled BLSTMs account for this by limiting the future context seen by the backward directed recurrence using chunk-wise processing. In this paper, we propose a new unidirectional neural network architecture of parallel time-delayed LSTM (PTDLSTM) streams, which limits the processing latency to 250 ms and shows significant improvements compared to prior art on a variety of ASR tasks.

Index Terms: unidirectional encoder architectures, streaming end-to-end ASR, low-latency neural networks, parallel time-delayed LSTM, automatic speech recognition

1. Introduction

The processing of temporal information is of paramount importance in automatic speech recognition (ASR), since most linguistic information for recognizing phones, phonemes, and larger units of speech such as syllables and words are encoded in spectral envelopes such as amplitude modulation frequencies [1–3]. For example, human listening experiments have shown that in noise-free acoustic conditions only four spectral bands of modulated noise are sufficient to achieve high speech recognition performance, while additional spectral bands increase speech intelligibility in the presence of noise, presumably due to masking effects, and modulation frequencies below 12 Hz are indispensable for speech recognition [1,4].

In today's ASR systems, processing of temporal information is accomplished by neural networks whose architectures define how well the system can recognize such cues. We distinguish two major types of neural network architectures in this work: unidirectional and bidirectional. Both types involve recurrent neural networks (RNNs) such as long short-term memory (LSTM) neural networks, whose model performance for end-to-end ASR has not yet been matched by solely using convolutional neural networks (CNNs) [5,6]. We suspect the reason is that RNNs can better compensate for the temporal dynamics of speech signals such as varying speech rates, whereas CNNs are restricted in this ability by their static temporal windowing. Bidirectional RNNs such as bidirectional LSTMs (BLSTMs) have demonstrated state-of-the-art results in ASR but at the expense of large output delays, which makes this type of architecture unsuitable for streaming ASR applications, where the text output must be generated soon after each spoken word. Latency-controlled BLSTMs (LCBLSTMs) account for this by limiting the future context seen by the backward directed LSTM using chunk-wise processing but at the expense of an increased computational cost due to overlapping chunks [7–9]. Therefore, the most widely used neural network architectures for streaming applications with end-to-end ASR systems rely on unidirectional LSTMs [10, 11].

In the present paper, new unidirectional neural network architectures are proposed for streaming ASR that are studied and compared to other common neural network architectures from the literature, such as a deep BLSTM [10], a deep LSTM [11, 12], a deep LCBLSTM [9], and a deep time-delay neural network (TDNN) with interleaving LSTM layers (TDNN-LSTM) [13]. The discussed architectures are applied as an encoder neural network in a hybrid connectionist temporal classification (CTC) and attention-based end-to-end ASR system [14]. Note that the hybrid CTC/attention ASR system of this work is not suitable for streaming recognition due to the full-sequence attention model. We proposed a triggered attention mechanism in earlier work that does enable streaming recognition with attention models [15], but we shall leave its combination with unidirectional encoder architectures to future work. Our proposed neural network architectures are based on a deep time-delay structure, where each layer may be composed of different neural network building blocks. Two new building blocks are proposed: a time-delay LSTM (TDLSTM), which is an LSTM cell with stacked time-delayed inputs followed by a bottleneck layer, and a neural network component of parallel time-delayed LSTM (PTDLSTM) streams that are merged using a bottleneck layer. ASR experiments are conducted on three different tasks of different size (between 80 and 960 hours of training) and of different language (English and Mandarin Chinese). Unless otherwise noted, the number of model parameters is the same for all tested encoder architectures to enable a balanced comparison between different settings.

2. Neural network architectures

In this section, the different components and building blocks of the investigated neural network architectures are presented. For the purpose of comparison and unless otherwise noted, subsampling by a factor of 3 is conducted for all neural network architectures by concatenating three consecutive frames of acoustic features and only forwarding every third stacked feature frame as an input to the encoder neural network, which is inspired

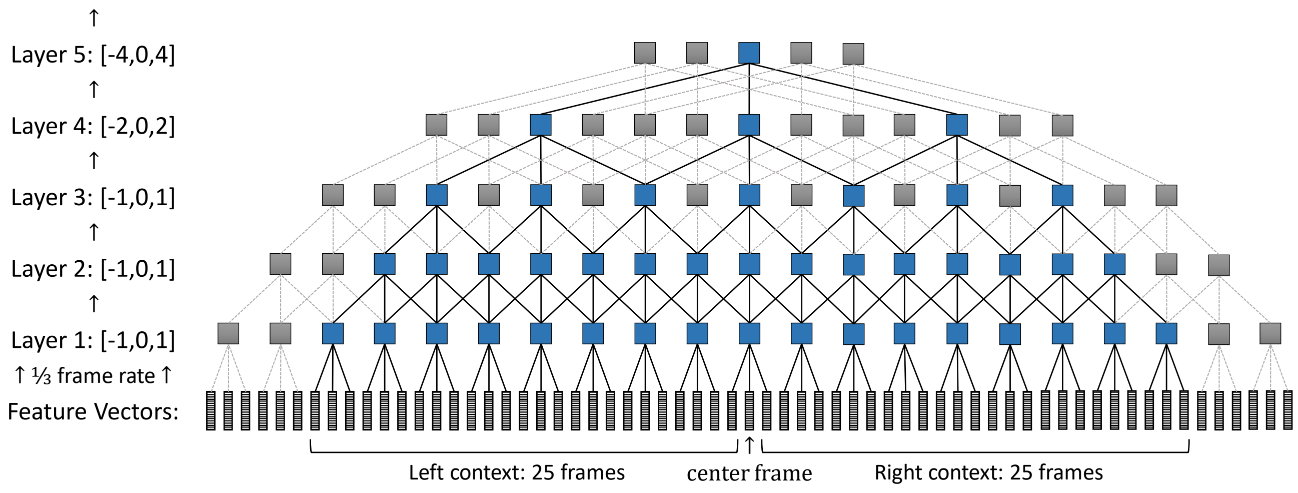

Figure 1: *Deep time-delay neural network architecture. The numbers in square brackets denote the frame delays of the input to each layer. Each square box represents a neural network building block. The solid black lines and the blue squares highlight the path to generate one output frame. The dashed lines and gray squares denote connections and building blocks of past and future output frames.*

by [10,11]. In addition, every neural network architecture is followed by a final linear projection layer, which is a feed-forward neural network that projects the encoder output to a vector of fixed size. No activation function is applied on top of this final projection layer, because in our experiments this has led to slightly better results compared to using a tanh nonlinearity.

2.1. Prior work

Deep BLSTM neural networks achieve state-of-the-art results in end-to-end ASR systems [10, 14, 16–18]. A BLSTM typically requires an entire speech utterance to compute an output, since each output frame is derived by knowing the entire past and future context of the speech utterance. This is a powerful architecture but due to the large future context required to compute the BLSTM output of an input sequence, it is not applicable for streaming ASR. In order to enable streaming ASR with BLSTMs, latency-controlled BLSTMs (LCBLSTMs) have been proposed that use overlapping chunks of frames to compute the output of the backward LSTM for a fixed size of future context [7–9]. One disadvantage of LCBLSTMs is an increased computational cost due to the overlapping backward LSTM output frames from different chunks. Other work focused on using deep CNN-based architectures for end-to-end ASR to reduce computational costs as well as to limit processing delays but yet without achieving significant improvements in terms of word error rates (WERs) compared to RNN-based architectures such as unidirectional LSTMs [5, 6]. A combination of time-delay neural networks (TDNN) [19, 20], which is also known as dilated convolution [21], and LSTM neural networks has been proposed as an acoustic model for hybrid ASR systems [13], but to the best of our knowledge this combination has not been tested yet for end-to-end ASR systems.

2.2. Baseline architectures

In this paper, four different neural network architectures from the literature are used as baseline models, respectively based on a BLSTM, LSTM, TDNN-LSTM, and LCBLSTM architecture. The BLSTM, LSTM, and LCBLSTM architectures are all composed of five layers followed by a final projection layer, which is similar to our proposed deep time-delay architecture, c.f. Section 2.3, and also a common encoder setup for end-to-end ASR systems [10]. The baseline LSTM and BLSTM encoder architectures of this paper are referred to as "Google"-LSTM and "Google"-BLSTM, respectively, because their configuration is similar to architectures used in Google publications [10–12].

The chunk size of our baseline LCBLSTM amounts to 8 frames after subsampling, which is equivalent to 24 feature frames of 10 ms frame rate, with a stride of 75% corresponding to a maximum delay of 250 ms (including the 1 frame delay for stacking 3 feature frames prior to the first layer), which is similar to the delay induced by our proposed deep time-delay architecture, c.f. Section 2.3. While the hidden states and cell states of the backward LSTM are reset after each chunk, the forward LSTM states are maintained.

Our baseline TDNN-LSTM model, which is similar to a setup proposed by [13], is referred to as "Kaldi"-TDNN-LSTM, and has the following structure: *input features* $\rightarrow$ *TDNN: [-1,0,1],[-1,0,1],[-1,0,1]* $\rightarrow$ *subsampling by 3* $\rightarrow$ *LSTM* $\rightarrow$ *TDNN: [-1,0,1],[-1,0,1]* $\rightarrow$ *LSTM* $\rightarrow$ *TDNN: [-1,0,1],[-1,0,1]* $\rightarrow$ *LSTM* $\rightarrow$ *projection layer*. In this notation, each enclosed square bracket denotes one TDNN layer with the associated frame splicing configuration. Note that the splicing configurations following *subsampling by 3* refer to the new (3$\times$ lower) frame rate, i.e., the total left and right context, relative to the center frame, amounts to 15 feature frames on both sides. We also experimented with delayed LSTM outputs similar to [13] but this did not help to improve results, which might be due to the alignment-free training of end-to-end ASR systems, whereby an LSTM-based system could potentially learn to delay outputs on its own.

2.3. Proposed architectures

Our proposed encoder architectures for end-to-end ASR are based on feed-forward and unidirectional neural networks only, which avoids additional computational costs that occur due to processing of overlapping chunks such as in LCBLSTM setups. Figure 1 illustrates our proposed deep time-delay neural network architecture, where the solid black lines and blue squares denote the path for processing one output frame. The time-delay tree structure of Figure 1 is designed to limit the

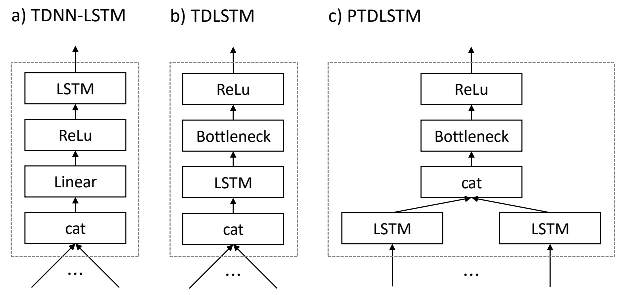

Figure 2: *Three different neural network building blocks are shown in a dotted box from a) to c). Each of the long input arrows represents an input from a different time-delay. Each solid box denotes one of the following processing operations: concatenation of inputs (cat), feed-forward neural network (Linear), rectified linear unit (ReLu) activation function, LSTM layer (LSTM), or bottleneck feed-forward neural network with 37.5% fewer neurons compared to a preceding neural network block (Bottleneck).*

overlap of leaves within each layer similar to [20], while capturing a large temporal context, which here amounts to 25 past and 25 future feature frames of 10 ms frame rate, thus inducing a latency of 250 ms. In this architecture, each of the squares represents a neural network building block, of which we consider three types shown in Figure 2. Building block a) of Figure 2 is similar to a TDNN-LSTM setting [13], while building blocks b) and c) are newly proposed settings. Figure 2 b) shows a time-delay LSTM (TDLSTM) neural network, which differs from the TDNN-LSTM by its reversing of the order of the feed-forward and the LSTM neural network layers, i.e., the LSTM processes the time-delayed and concatenated input prior to the feed-forward neural network and the rectified linear unit (ReLu) activation function. In addition, the feed-forward neural network layer acts as a bottleneck, whose dimension is 62.5% of that of the preceding LSTM layer, which aims at reducing the input size to a following layer. Figure 2 c) shows a parallel time-delayed LSTM (PTDLSTM) neural network layer, where multiple time-delayed input streams are each processed by separate LSTM layers, whose parameters are not shared, prior to concatenating LSTM outputs and further processing using a bottleneck layer and a ReLu activation function. This architecture is inspired by the parallel forward and backward LSTM streams of BLSTM models. Note that if a TDLSTM or PTDLSTM building block is used in our deep time-delay architecture for the 5th layer, i.e., the final layer, we do not use an addition projection layer but instead set the size of the bottleneck layer to be equal to the dimension of encoder states, and no activation function is applied.

3. Experimental Setup

We conducted ASR experiments on three different data sets of different size, ranging from 90 to 960 hours of training data, and of different language, which are English and Mandarin Chinese. We use the Wall Street Journal (WSJ) corpus of read English newspapers [22], the LibriSpeech corpus [24], which is based on an open-source English audio books project featuring various recording qualities, and the Mandarin telephone speech corpus developed by the Hong Kong University of Science and Technology (HKUST) [23]. Basic information about the corpora are shown in Table 1.

Table 1: *ASR corpora information.*

WSJ1 (English) [22]	#Utterances	Size [h]
Training	37,416	80
Development (dev93)	503	1.1
Test (eval92)	333	0.7
HKUST (Mandarin) [23]	**#Utterances**	**Size [h]**
Training	197,391	174
Development	4,000	4.8
Test	5,413	4.9
LibriSpeech (English) [24]	**#Utterances**	**Size [h]**
Training	281,231	960
Development [clean/other]	2,703 / 2,864	5.4 / 5.3
Test [clean/other]	2,620 / 2,939	5.4 / 5.1

All encoder architectures are tested within an end-to-end ASR system based on a hybrid CTC/attention model trained on a multi-objective loss function

$$\mathcal{L} = \lambda \log p_{\text{ctc}} + (1 - \lambda) \log p_{\text{att}}, \tag{1}$$

where p_{ctc} and p_{att} denote the CTC and attention model loss functions, and λ controls their relative weight [14]. As input to the system, we use 80-dimensional log Mel-filterbank features and pitch features plus their first and second order derivatives (80+3=83 feature dimensions).

Specific model and training parameters are summarized in Table 2. The number of trainable encoder parameters is constant for all tested encoder configurations and only depends on the training data size. The number of output targets of the WSJ and HKUST end-to-end systems amount to 50 (number of English characters in WSJ) and 3653 (number of Mandarin characters in HKUST), respectively. The LibriSpeech ASR system uses 5000 sentence-pieces as output targets, which are derived by the sentence-piece tokenizer proposed by [25]. In this work, an RNN-based language model (LM) is applied to the output of the end-to-end ASR system via cold fusion. A word-based LM of 65k words is applied to the WSJ test data [26] and character-based LMs are applied to the HKUST and LibriSpeech data sets [27].

4. Results

Results for our proposed encoder architectures as well as the baseline encoder models are shown in Table 3. Note that the number of parameters of all encoder neural networks is the same for each ASR task. An exception is made for the "Kaldi"-TDNN-LSTM model on the LibriSpeech data set: this model had to be limited to a maximum of approximately 80M parameters in our experiments because of GPU-related out-of-memory errors that occurred at training time. These errors are due to the fact that the initial three TDNN layers of the "Kaldi"-TDNN-LSTM model are running at the higher frame rate, and to the high memory demands of attention model training. Thus, the "Kaldi"-TDNN-LSTM encoder has fewer parameters compared to other settings in the LibriSpeech experiments, since we preferred not to change the model topology or the training batch size for this architecture.

The "Google"-BLSTM model serves as a benchmark to determine the discrepancy between offline and online (streaming) encoder models. By comparing the "Google"-BLSTM to its unidirectional LSTM counterpart, it becomes obvious that

Table 2: *Experimental hyperparameters.*

WSJ model parameters	
# trainable encoder parameters	18M
Size of projection layer	320
# decoder LSTM cells / layers	300 / 1
HKUST model parameters	
# trainable encoder parameters	80M
Size of projection layer	1024
# decoder LSTM cells / layers	1024 / 2
LibriSpeech model parameters	
# trainable encoder parameters	115M
Size of projection layer	1024
# decoder LSTM cells / layers	1024 / 2
Common training parameters	
Optimization	AdaDelta
Adadelta ρ	0.95
Adadelta ϵ / ϵ decaying factor	10^{-8} / 10^{-2}
Maximum # epochs	15 (WSJ, HKUST)
	10 (LibriSpeech)
λ	0.2 (WSJ)
	0.5 (HKUST, LibriSpeech)
Decoding parameters	
Language model / CTC weight	1.0 / 0.3 (WSJ)
	0.3 / 0.6 (HKUST)
	0.5 / 0.5 (LibriSpeech)

the missing future information incorporated by the backward LSTM increases error rates significantly between 1.6% and 5.6% on an absolute scale. The "Kaldi"-TDNN-LSTM model can compensate for this lack of information to some extent by analysing 150 ms of future context, which improves error rates for the WSJ and HKUST tasks by 1.1% on average, while error rates of the LibriSpeech data have slightly increased, which can be explained by the reduced model size as explained in the previous paragraph. The LCBLSTM model further enhances recognition results, especially for the HKUST data set. The TDNN-LSTM encoder model, which is based on our proposed deep time-delay architecture shown in Figure 1, demonstrates improved recognition results for the WSJ and HKUST tasks compared to the "Google"-LSTM and "Kaldi"-TDNN-LSTM model, whereas word error rates (WERs) of LibriSpeech are deteriorated. The reason is not obvious and requires deeper investigation.

The proposed TDLSTM encoder architecture improves error rates for the HKUST and LibriSpeech recognition tasks on average by 1.5% compared to both baseline models. WSJ-based results of the TDLSTM neural network are better compared to the "Google"-LSTM model and marginally worse compared to the "Kaldi"-TDNN-LSTM neural network. This demonstrates that the TDLSTM architecture improves ASR results compared to the TDNN-LSTM architecture, with the essential differences between the two being that in the former case the LSTM receives the time-delayed and stacked input, while in the latter case a feed-forward neural network processes this input prior to the LSTM layer. The PTDLSTM encoder architecture further improves ASR results by using parallel LSTMs to process each input of different frame delay separately, which is inspired by BLSTM neural networks that process the forward and backward sequence using two parallel LSTMs as well. Note that the first layer of our PTDLSTM encoder model is composed of a TDLSTM building block and layer two to five are PTDLSTM neural networks, cf. Figures 1 and 2. The results of Table 3 show that our PTDLSTM architecture outperforms all baseline architectures and reduces the gap towards the BLSTM model. In addition, the PTDLSTM is entirely unidirectional unlike the LCBLSTM, which explains the training and inference speed advantages we could observe with our implementation.

[1]The LibiSpeech experiments of the LCBLSTM model had not finished by the submission deadline. In addition, HKUST results were obtained with an LCBLSTM model of larger latency (chunk size) compared to our proposed architectures. LCBLSTM training is slow and other results are still computing. The missing/correct results will be added in the camera-ready paper, if it is accepted for publication.

Table 3: *Word error rates (WSJ and LibriSpeech) as well as character error rates (HKUST) of the hybrid CTC/attention decoder using different encoder architectures. The upper section presents results of the BLSTM-based baseline model, which cannot be used in a streaming fashion, whereas the middle and lower sections present results of the baseline and our proposed encoder architectures, respectively, which are suitable for streaming recognition. Note that the "Kaldi"-TDNN-LSTM model has fewer model parameters compared to the other encoder neural networks for the LibriSpeech data set, c.f. Section 4 for explanation.*

	WSJ		HKUST		LibriSpeech			
Encoder Architecture	dev	test	dev	test	dev clean	dev other	test clean	test other
"Google"-BLSTM	7.9	4.7	29.9	28.9	4.7	14.1	4.9	15.2
"Google"-LSTM	9.9	6.5	35.5	33.8	6.3	18.2	6.5	19.4
"Kaldi"-TDNN-LSTM	8.8	5.3	34.5	32.7	6.8	18.7	6.9	19.9
LCBLSTM[1]	8.8	5.8	31.6	30.2	-	-	-	-
TDNN-LSTM	8.5	5.3	33.0	31.3	7.4	19.9	7.4	21.1
TDLSTM	9.1	5.7	32.7	31.0	5.9	17.0	6.0	18.2
PTDLSTM	8.0	5.4	31.4	30.1	5.6	16.2	5.7	16.9

5. Conclusions

In this paper, we presented and compared various encoder neural network architectures for end-to-end ASR that are suitable for streaming applications. We proposed two novel unidirectional neural network models, the time-delay LSTM (TDLSTM) and the parallel time-delayed LSTM (PTDLSTM) architectures. Both encoder neural network models demonstrated improved ASR results compared to a deep LSTM and TDNN-LSTM model, which are similar to a "Google" and "Kaldi" implementation, using three ASR tasks of different size (80h to 960h of training data) and language (English and Mandarin Chinese). The average relative word/character error rate improvement of the PTDLSTM model amounts to 13.2% and 11.0% compared to our baseline LSTM and TDNN-LSTM models, respectively. The PTDLSTM has also shown better error rates compared to a latency-controlled BLSTM of similar size and latency, which here amounts to 250 ms, while also improving training and inference speed due to avoiding the backward LSTM computation from overlapping chunks.

6. References

[1] R. V. Shannon, F.-G. Zeng, V. Kamath, J. Wygonski, and M. Ekelid, "Speech recognition with primarily temporal cues," *Science*, vol. 270, no. 5234, pp. 303–304, 1995.

[2] N. Moritz, J. Anemüller, and B. Kollmeier, “An auditory inspired amplitude modulation filter bank for robust feature extraction in automatic speech recognition,” *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 23, no. 11, pp. 1926–1937, 2015.

[3] N. Moritz, B. Kollmeier, and J. Anemüller, “Integration of optimized modulation filter sets into deep neural networks for automatic speech recognition,” *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 24, no. 12, pp. 2439–2452, 2016.

[4] T. M. Elliott and F. E. Theunissen, “The modulation transfer function for speech intelligibility,” *PLoS Computational Biology*, vol. 5, no. 3, 2009.

[5] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. Laurent, Y. Bengio, and A. C. Courville, “Towards end-to-end speech recognition with deep convolutional neural networks,” in *Proc. ISCA Interspeech*, 2016, pp. 410–414.

[6] K. Krishna, L. Lu, K. Gimpel, and K. Livescu, “A study of all-convolutional encoders for connectionist temporal classification,” in *Proc. IEEE ICASSP*, 2018, pp. 5814–5818.

[7] K. Chen and Q. Huo, “Training deep bidirectional LSTM acoustic model for LVCSR by a context-sensitive-chunk BPTT approach,” *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 24, no. 7, pp. 1185–1193, 2016.

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

[9] A. Zeyer, R. Schlüter, and H. Ney, “Towards online-recognition with deep bidirectional LSTM acoustic models,” in *Proc. ISCA Interspeech*, 2016, pp. 3424–3428.

[10] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, “A comparison of sequence-to-sequence models for speech recognition,” in *Proc. ISCA Interspeech*, 2017, pp. 939–943.

[11] K. Rao, H. Sak, and R. Prabhavalkar, “Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer,” *CoRR*, vol. abs/1801.00841, 2018.

[12] T. N. Sainath, C. Chiu, R. Prabhavalkar, A. Kannan, Y. Wu, P. Nguyen, and Z. Chen, “Improving the performance of online neural transducer models,” *CoRR*, vol. abs/1712.01807, 2017.

[13] V. Peddinti, Y. Wang, D. Povey, and S. Khudanpur, “Low latency acoustic modeling using temporal convolution and lstms,” *IEEE Signal Process. Lett.*, vol. 25, no. 3, pp. 373–377, 2018.

[14] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, “Hybrid ctc/attention architecture for end-to-end speech recognition,” *J. Sel. Topics Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[15] N. Moritz, T. Hori, and J. L. Roux, “Triggered attention for end-to-end speech recognition,” in *Proc. IEEE ICASSP*, 2019.

[16] 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, 2005.

[17] A. Graves, S. Fernández, and J. Schmidhuber, “Bidirectional LSTM networks for improved phoneme classification and recognition,” in *Proc. International Conference on Artificial Neural Networks: Formal Models and Their Applications (ICANN)*, 2005, pp. 799–804.

[18] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *Proc. IEEE ICASSP*, 2016, pp. 4960–4964.

[19] A. H. Waibel, T. Hanazawa, G. E. Hinton, K. Shikano, and K. J. Lang, “Phoneme recognition using time-delay neural networks,” *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 37, no. 3, pp. 328–339, 1989.

[20] V. Peddinti, D. Povey, and S. Khudanpur, “A time delay neural network architecture for efficient modeling of long temporal contexts,” in *Proc. ISCA Interspeech*, 2015, pp. 3214–3218.

[21] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” in *ICLR*, 2016.

[22] L. D. Consortium, “CSR-II (wsj1) complete,” *Linguistic Data Consortium, Philadelphia*, vol. LDC94S13A, 1994.

[23] Y. Liu, P. Fung, Y. Yang, C. Cieri, S. Huang, and D. Graff, “HKUST/MTS: A very large scale mandarin telephone speech corpus,” in *Proc. ISCSLP*, vol. 4274, 2006, pp. 724–735.

[24] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” in *Proc. IEEE ICASSP*, 2015, pp. 5206–5210.

[25] T. Kudo and J. Richardson, “Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,” *CoRR*, vol. abs/1808.06226, 2018.

[26] T. Hori, J. Cho, and S. Watanabe, “End-to-end speech recognition with word-based RNN language models,” *CoRR*, vol. abs/1808.02608, 2018.

[27] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, “Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM,” in *Proc. ISCA Interspeech*, 2017, pp. 949–953.