Low latency acoustic modeling using temporal convolution and LSTMs

Vijayaditya Peddinti*, Yiming Wang, Daniel Povey, Sanjeev Khudanpur

***Abstract*—Bidirectional long short term memory (BLSTM) acoustic models provide significant word error rate reduction compared to their unidirectional counterpart, as they model both the past and future temporal contexts. However it is non-trivial to deploy bidirectional acoustic models for online speech recognition due to an increase in latency. In this paper we propose the use of temporal convolution, in the form of time-delay neural network (TDNN) layers, along with unidirectional LSTM layers to limit the latency to 200 ms. This architecture has been shown to outperform the state-of-the-art low frame rate (LFR) BLSTM models. We further improve these LFR BLSTM acoustic models by operating them at higher frame rates at lower layers and show that the proposed model performs similar to these mixed frame rate (MFR) BLSTMs. We present results on the Switchboard 300 Hr LVCSR task and the AMI LVCSR task, in the three microphone conditions.**

***Index Terms*—time delay neural networks, recurrent neural networks, LSTM, acoustic model**

I. Introduction

The use of future context information is typically shown to be helpful for acoustic modeling. This context is provided in feed-forward neural networks (FFNNs) by splicing a fixed set of future frames in the input representation [1] or through temporal convolution over the future context [2]. In unidirectional LSTM acoustic models this is accomplished using a delayed prediction of the output labels [3], while in bidirectional LSTMs this is accomplished by processing the data in the backward direction using a separate LSTM layer [4], [5], [6].

Among the LSTM acoustic models, including their variants like highway LSTM networks [7], the bidirectional versions have been shown to outperform the unidirectional versions by a large margin [8], [7]. However the latency of the bidirectional models is significantly larger, making them unsuitable for online speech recognition. To overcome this limitation chunk based training and decoding schemes [8], [9], [10], [11] have been previously investigated.

In this paper we propose interleaving of temporal convolution, with unidirectional LSTM layers, for modeling the future temporal context. This model is shown to outperform LFR-BLSTMs in two different LVCSR tasks, while enabling online decoding with a maximum latency of 200 ms. We also show that the model performs similar to the improved MFR-BLSTMs, where a higher frame-rate is used at lower layers.

The paper is organized as follows : Section II presents the prior work, Section III presents the motivation for this effort, Section IV describes the proposed model, Section V describes the experimental setup, Section VI presents the results and finally the conclusion is presented in Section VII.

II. Prior Work

The superior performance of BLSTM acoustic models has motivated recent research efforts [10], [7], [12], [8] to make them amenable for online decoding. A common characteristic of these methods is the use of frame chunks in place of the entire utterance, and they differ in the way the recurrent states are initialized when processing these chunks.

Chen *et al*., [10] proposed the use of context-sensitive chunks (CSC), where a fixed context of frames to the left and right of the chunk is used to intialize the recurrent states of the network. Zhang *et al*., [7] carried over the recurrent states for the forward LSTM from previous chunks reducing the computation on the left context. Xue *et al*., [12] proposed the use of a feed-forward DNN to estimate the initial state of the backward LSTMs, for a given chunk. They also proposed the use of a simple RNN in place of an LSTM for the backward direction. Zeyer *et al*., [8] proposed the use of overlapping chunks, without additional chunk context, and combining the posterior estimates from overlapping chunks. In all these *online* variants inference is restricted to chunk-level increments to amortize the computation cost of backward LSTMs, which significantly increases the model latency.

In this paper we propose the use of temporal convolution for modeling the future temporal context, to enable inference with frame-level increments of audio. Combining convolution with recurrent layers has been previously shown to be helpful for acoustic modeling in [13], [14]. However the use of spectro-temporal convolution restricts the placement of convolutional layers to below the recurrent layers in [13]. In this work we focus on temporal convolution which affords the exploration of more combinations, including the interleaving of convolutional and recurrent layers[1]. Temporal convolution was also used in [14] but just above or below the recurrent layer stack.

*corresponding author (p.vijayaditya@gmail.com)

This work was partially supported by DARPA LORELEI Grant No HR0011-15-2-0024, NSF Grant No CRI-1513128 and IARPA Contract No 2012-12050800010.

Vijayaditya Peddinti, Daniel Povey and Sanjeev Khudanpur are with the Center for Language and Speech Processing(CLSP) and Human Language Technology Center of Excellence (HLTCoE), Johns Hopkins University (JHU), USA. Yiming Wang is with CLSP, JHU, USA

[1]Recent experiments by Gaofeng Cheng have shown that combining spectro-temporal convolution with the architectures proposed in this paper could further improve the results [15]. This work is in progress.

TABLE I
COMPARISON OF SUB-SAMPLED TDNN ARCHITECTURES ON THE 300 HR SWITCHBOARD LVCSR TASK

Model	Layer-wise context							Network context	WER (%)		
									SWBD	CHM	Total
TDNN-A	{-2,-1,0,1,2}	{-1,2}	{-3,3}	{-7,2}	{0}	{0}	{0}	[-13, 9]	11.1	21.8	16.5
TDNN-B	{-2,-1,0,1,2}	{-1,2}	{-3,0,3}	{-3,0,3}	{-3,0}	{0}	{0}	[-12, 10]	10.5	21.9	16.3
TDNN-C	{-2,-1,0,1,2}	{-1,0,1}	{-1,0,1}	{-3,0,3}	{-3,0,3}	{-3,0}	{0}	[-13, 10]	10.3	20.7	15.5
TDNN-D	{-1,0,1}	{-1,0,1}	{-1,0,1}	{-3,0,3}	{-3,0,3}	{-3,0,3}	{-3,0,3}	[-15, 15]	9.6	19.9	14.8

$*$ Please note that the overall temporal context of the neural network is kept similar, except in TDNN-D. There are 625 filters in each TDNN layer. However due to the change in temporal convolution kernel context there is a slight increase in parameters as we move from TDNN-A to TDNN-D.

III. MOTIVATION

In this section we present a set of empirical results which motivate the model proposed in this paper. We initially detail the modeling of large temporal contexts using TDNNs and compare their performance with (B)LSTMs. All these models are trained using the LF-MMI cost function computed on 33 Hz outputs [16]. These results correspond to the Switchboard (SWBD) and Call-Home (CHM) subsets of the Hub5'00 set (LDC2002S09) and the 300 hr Switchboard LVCSR task. Please see Section V for the experimental setup and neural network hyper-parameters.

A. Sub-sampled TDNNs

Time-delay neural networks are effective in modeling long-span temporal contexts [2]. However there is a linear increase in parameters with increase in temporal context; and also a linear increase in computation when training with frame randomization. In the sub-sampled time-delay neural networks [17] the issue of linear increase, both in parameters and computation, is alleviated using a non-uniform sub-sampling technique where the layer frame rate decreases with the layer depth. This sub-sampled TDNN (TDNN-A), is the baseline acoustic model in this paper.

Recently training with just sequence level cost functions has been shown to be very effective for acoustic modeling [18], [16]. In this scenario frame-shuffling is no longer applicable. Thus the computation can be amortized over all the outputs in the sequence and sub-sampled TDNNs can match the output frame rate even at deeper layers.

In [18], [16] and [19] the authors have shown that lower output frame rate models outperform conventional frame rate models, while providing great savings in computation. They propose the use of reduced frame rates of 25-33 Hz for the neural network outputs. Hence we change the frame rate at all the layers in the TDNN to match the output frame rate (33 Hz). This configuration is denoted TDNN-B.

Finally, we also explore the use of higher frame rate (100 Hz) at the lower layers of the TDNN. We restrict the higher frame rates to the lower layers as this preserves the computational efficiency; and as the gains were negligible when increasing the frame rate even at the higher layers. This TDNN is denoted TDNN-C. Finally we tuned the temporal contexts of the TDNN layers (TDNN-D).

The configurations of the sub-sampled TDNNs described above and their performance is shown in Table I. We specify the TDNN architectures in terms of the splicing indices which define the temporal convolution kernel input at each layer. e.g. $\{-3,0,3\}$ means that the input to the temporal convolution at a given time step t is a spliced version of previous layer outputs at times $t-3$, t, $t+3$. It can be seen that using the higher frame rates at lower layers and tuning the temporal contexts of the layers (TDNN-D) provides 10.3% relative gain over the sub-sampled TDNNs proposed in [17] (TDNN-A) [2].

B. Comparison with LSTMs

We compare the performance of the best TDNN model (TDNN-D) with stacked (B)LSTM models. These have three[3] layers of (B)LSTM. These models are denoted LFR-LSTM and LFR-BLSTM as all their layers operate at a low frame rate (LFR) of 33 Hz similar to [18] and [19] .

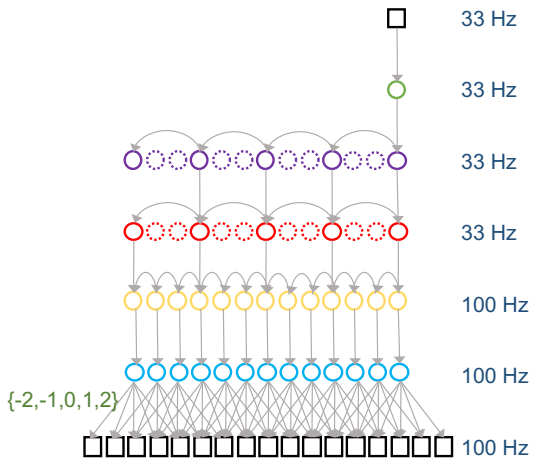

Fig. 1. Dependencies among activations at various layers and time-steps in the stacked LSTM network with the lowest LSTM layer operating at 100 Hz

Based on our observations with TDNNs, we explored the use of higher frame rate (100 Hz) at lower (B)LSTM layers. This architecture is similar to the hierarchical subsampling networks, proposed in [20] and more recently applied in [21] and [22]. We denote these models as MFR-LSTM and MFR-BLSTM as they use a mixed frame rate (MFR) across layers. Figure 1 represents the computation in the MFR-LSTM.

Table II compares the models discussed in this section with the best TDNN model described in Section III-A. Firstly, it can be seen that operating the lower LSTM layers at a higher

[2]Part of this improvement was already realized in [16].

[3]The depth and other hyper-parameters of the LSTM and BLSTM models have been tuned. Further increase in depth leads to negligible improvements.

TABLE II
PERFORMANCE COMPARISON OF TDNN, LSTM AND BLSTM ON THE 300 HR SWITCHBOARD LVCSR TASK

Model	WER (%)		
	SWBD	CHM	Total
TDNN-D	9.6	19.9	14.8
LFR-LSTM	10.1	21.0	15.6
LFR-BLSTM	9.6	19.2	14.5
MFR-LSTM	9.9	19.7	14.8
MFR-BLSTM	9.0	18.1	**13.6**

frame rate is beneficial. It results in a relative improvement of $\sim 6\%$ in BLSTM and $\sim 5\%$ in LSTM. However the overall computational complexity increases by 30% during inference compared to the corresponding LFR models, for our input sizes of interest. Operating even the higher (B)LSTM layers at a higher frame rate did not lead to gains, while further increasing the computational complexity.

Secondly, it can be seen that both the TDNN and LSTM models perform worse than both the BLSTM models. The superior performance of the bidirectional recurrent models compared to their unidirectional counterparts can be attributed to the modeling of the future context which could not be matched even with the use of output delay in LSTMs (see Section V for details of output delay). Thus to model the future context in the LSTMs, we propose the use of TDNN layers.

IV. PROPOSED MODEL

In this section we detail the use of temporal convolution in the recurrent neural networks, to model the future temporal context. We explore three different ways of combining temporal convolution and LSTMs *viz.*,

- Stacking LSTMs over TDNNs (TDNN-LSTM-A)
- Stacking TDNNs over LSTMs (TDNN-LSTM-B)
- Interleaving TDNNs and LSTMs (TDNN-LSTM-C)

Figure 2 represents the computation in the TDNN-LSTM-C network. It can be seen that all the LSTM layers, which can be computationally expensive, operate at the 33 Hz frame rate.

Further, to compare the benefits of performing temporal convolution in BLSTM models, we also interleave temporal convolution with the forward and backward LSTMs (TDNN-BLSTM-A) or with forward and backward LSTM stack *i.e.,* the BLSTM layer (TDNN-BLSTM-B).

As only the lower TDNN layers operate at a 100 Hz frame rate in TDNN-LSTM-C and TDNN-BLSTM-A, we also verify if additional gains can be had by operating even the lowest recurrent layer at 100 Hz frame rate, similar to MFR-(B)LSTM. These models are denoted as TDNN-LSTM-D and TDNN-BLSTM-C, respectively.

V. EXPERIMENTAL SETUP

Experiments were conducted using the Kaldi toolkit [23]. We report the main set of results on 300 hour Switchboard conversational telephone speech task. We perform phone-level sequence training, without frame level pretraining, using the lattice-free MMI objective [16] on outputs of frame rate 33 Hz. The experimental setup is same as the one described

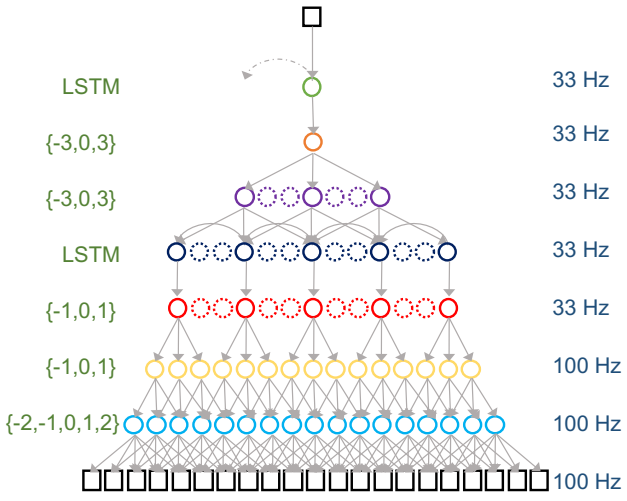

Fig. 2. Dependencies among activations in a stacked TDNN-LSTM network with interleaved temporal convolutions. The convolution kernel input contexts are on left and the layer-wise frame rates are on the right.

in [16]. We also present results on the AMI LVCSR task [24], [25], [26], for the individual headset, single distant and multiple distant microphone conditions (IHM, SDM, MDM). This recipe is same as the one described in [27], where the SDM and MDM LVCSR systems are trained with numerator lattices generated from parallel IHM data. A salient difference in the current setup compared to [16], [27] is the use of a recurrence scaling factor (0.85) in the LSTM layers for generalization to longer sequences than those seen during training. This is of interest in our scenario as we train on fixed length context-sensitive chunks (CSCs, [10]) in randomized order, but decode with state saving across chunks to reduce latency. This exposes the model to longer length sequences during inference than those seen in training.

The cost function is computed using CSCs of width $1000-1500$ ms and left/right context of 400 ms. The chunk contexts are further expanded to satisfy the context requirements of TDNNs. The LSTMs are trained with an output delay of 50 ms[4]. We used chunk based decoding with similar chunk-width, but we increased the chunk contexts to 500 ms[5] as this reduced WER.[6]

Model latency during inference: Latency is affected by input context, chunk-width, chunk contexts and output delay. Further each TDNN layer adds to the latency due its kernel context. As the recurrent state of the forward LSTM can be propagated across chunks, the chunk left context does not add to the latency. Further with forward LSTMs chunk-width does not add to latency, as we can perform inference in frame-level increments. However when backward LSTMs are used inference is performed in chunk-level increments to amortize the backward LSTM cost over the entire chunk. Thus chunk-width and chunk right context add to the latency in BLSTM models.

[4]based on a selective search of values from 0-150 ms
[5]based on a grid search of values from 0-800 ms
[6]Scripts to reproduce experiments in this paper [28]

TABLE III
PERFORMANCE COMPARISON OF VARIOUS MODELS IN THE 300 HR SWBD LVCSR TASK†

Model	Architecture*	Latency (ms)	Matched Inference			State-saving Inference		
			WER(%)		RTF	WER(%)		RTF
			SWBD	Total		SWBD	Total	
TDNN-D	$T^{100}T^{100}T^{100}TTTT$	150	9.6	14.8	0.8	9.6	14.8	0.9
LFR-LSTM	$L_f L_f L_f$	70	10.1	15.6	2.5	10.7	16.2	1.2
MFR-LSTM	$L_f^{100} L_f L_f$	70	9.9	14.8	2.7	10.2	15.3	1.8
TDNN-LSTM-A	$T^{100}T^{100}T^{100}TTTTL_f L_f L_f$	200	9.5	14.6	2.7	9.7	14.8	1.8
TDNN-LSTM-B	$L_f^{100} L_f^{100} L_f^{100} T^{100} T^{100} T^{100} TTTT$	200	9.4	14.3	4.0	9.3	14.3	2.3
TDNN-LSTM-C	$T^{100}T^{100}T^{100}L_f TTL_f TTL_f$	200	9.2	14.2	3.7	9.4	14.4	1.9
TDNN-LSTM-D	$T^{100}T^{100}T^{100}L_f^{100} TTL_f TTL_f$	200	9.0	13.9	4.8	9.4	14.4	2.4
LFR-BLSTM	$[L_f, L_b], [L_f, L_b], [L_f, L_b]$	2020	9.6	14.5	4.7			
MFR-BLSTM	$[L_f^{100}, L_b^{100}][L_f, L_b][L_f, L_b]$	2020	9.0	13.6	6.6			
TDNN-BLSTM-A	$T^{100}T^{100}T^{100}[L_f T, L_b T][L_f T, L_b T][L_f, L_b]$	2170	9.2	14.1	4.6			
TDNN-BLSTM-B	$T^{100}T^{100}T^{100}[L_f, L_b]TT[L_f, L_b]TT[L_f, L_b]$	2170	9.1	13.8	4.7			
TDNN-BLSTM-C	$T^{100}T^{100}T^{100}[L_f^{100} T^{100}, L_b^{100} T^{100}][L_f T, L_b T][L_f, L_b]$	2130	9.0	13.8	9.6			

forward LSTM - L_f, backward LSTM - L_b, TDNN - T, default layer frame-rate - 33 Hz and other frame rates are specified in the super-script.
* $[.,.]$ - layers which operate on the same input and whose outputs are appended e.g.BLSTM = $[L_f, L_b]$.
L_f, L_b : cell size - 1024, recurrent and non-recurrent projections-256; TDNN filters/layer : in TDNN-LSTMs - 1024 and in TDNN-BLSTMs - 512
TDNN layer contexts are same as TDNN-D (see Table I)
† State-saving inference for BLSTMs is difficult to implement in our framework, so we do not present these results.

VI. RESULTS

In this section we provide a comparison of acoustic models in the SWBD and AMI LVCSR tasks. Table III presents a broader comparison of the acoustic models on the 300 Hr Switchboard LVCSR task, and Table IV compares a subset of models on the AMI-LVCSR task. Table V compares performance of systems trained with and without recurrence scaling.

We perform decodes with two different posterior estimation methods, *Matched Inference* where the state of the network for a chunk is estimated using a chunk left context, similar to the training; or *State-saving Inference* where the state is copied from the final state in the previous chunk.

TABLE IV
PERFORMANCE COMPARISON IN THE AMI LVCSR TASK WITH MATCHED INFERENCE‡

Model	WER (%)					
	IHM		SDM		MDM	
	Dev	Eval	Dev	Eval	Dev	Eval
TDNN-D	21.7	22.1	39.9	43.9	36.6	40.1
LFR-BLSTM	21.0	20.9	38.8	42.0	35.4	38.4
MFR-BLSTM	20.6	20.3	37.4	40.5	34.5	37.3
TDNN-LSTM-C†	20.8	20.5	37.3	40.4	34.1	36.8
TDNN-BLSTM-A	20.7	20.7	37.0	40.4	34.2	36.6

† TDNN-LSTM-C has additional TDNN layers
‡ Parameters reduced compared to SWBD models

TABLE V
IMPACT OF RECURRENCE SCALE : 300 HR SWBD LVCSR TASK

Model	Total WER on Hub'00 (%)			
	Without scaling		With scaling	
	MI	SSI	MI	SSI
TDNN-LSTM-A	14.2	15.6	14.6	14.8
TDNN-LSTM-C	13.9	16.0	14.2	14.4
MFR-BLSTM	13.5	-	13.6	-

MI : Matched Inference SSI : State saving Inference

From Table III: TDNN-LSTMs B and C perform better among A, B and C; while C has lower real-time factor (RTF). From Tables III, IV and V: TDNN-LSTM-C performs similar to the MFR-BLSTM in AMI task, but there is a difference in performance in SWBD task which slightly reduces when trained without recurrence scaling.

From Table III: Interleaving TDNNs with BLSTMs rather than forward and backward LSTMs separately was better (TDNN-BLSTM-A vs B); both these models perform better than LFR-BLSTM. However there was no benefit compared to MFR-BLSTM, in terms of performance, though RTFs are lower than MFR-BLSTM.

Operating the lowest (B)LSTM layer in TDNN-(B)LSTMs at 100 Hz, *i.e.*, TDNN-LSTM-D and TDNN-BLSTM-C led to performance gains with additional computational cost, when compared to TDNN-LSTM-C and TDNN-BLSTM-A, respectively.

In preliminary experiments we observed that using additional temporal context was beneficial for TDNN-LSTMs in all three AMI tasks. This might be attributed to the fact that this data is reverberated, at least for the SDM and MDM tasks. This additional context was provided using an additional TDNN layer between successive LSTM layers and this leads to 60 ms of additional latency for AMI models.

From Table V : It can be clearly seen that recurrence scaling helps better generalize to longer sequence lengths i.e., for state-saving inference. We are currently exploring other mechanisms to better generalize to chunk-widths not seen during training. These include use of frame-level dropout of recurrent states [29] with longer chunk-widths.

VII. CONCLUSION

In this paper we proposed interleaving of temporal convolution with LSTM layers which was shown to be effective for modeling of the future temporal context, while affording low latency (200 ms) online inference. We showed that this architecture not only performs better than the stacked LFR-BLSTM network, but also performs similar to the superior stacked MFR-BLSTM.

REFERENCES

[1] H. A. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*. Springer Science & Business Media, 1994, vol. 247.

[2] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 3, pp. 328–339, Mar. 1989.

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

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

[5] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional LSTM networks for improved phoneme classification and recognition," *Artificial Neural Networks: Formal Models and Their Applications*, 2005.

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

[7] Y. Zhang, G. Chen, D. Yu, K. Yaco, S. Khudanpur, and J. Glass, "Highway long short-term memory rnns for distant speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5755–5759.

[8] A. Zeyer, R. Schluter, and H. Ney, "Towards online-recognition with deep bidirectional LSTM acoustic models," *Proceedings of Interspeech*, vol. 08-12-Sept, pp. 3424–3428, 2016.

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

[10] K. Chen, Z.-J. Yan, and Q. Huo, "Training Deep Bidirectional LSTM Acoustic Model for LVCSR by a Context-Sensitive-Chunk BPTT Approach," in *Proceedings of the Interspeech*, 2015.

[11] A.-r. Mohamed, F. Seide, D. Yu, J. Droppo, A. Stoicke, G. Zweig, and G. Penn, "Deep bi-directional recurrent networks over spectral windows," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 78–83.

[12] S. Xue and Z. Yan, "Improving latency-controlled BLSTM acoustic models for online speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2017.

[13] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4580–4584.

[14] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International Conference on Machine Learning*, 2016, pp. 173–182.

[15] *Comparison of TDNN-LSTMs and CNN-TDNN-LSTMs.*, 2017 (accessed June 15 , 2017), https://github.com/kaldi-asr/kaldi/pull/1685.

[16] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi," in *Proceedings of Interspeech*, 2016, pp. 2751–2755.

[17] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proceedings of Interspeech*, 2015.

[18] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," in *Proceedings of Interspeech*, 2015.

[19] G. Pundak and T. N. Sainath, "Lower frame rate neural network acoustic models," in *Proceedings of Interspeech 2016*, 2016, pp. 22–26.

[20] A. Graves, "Hierarchical subsampling networks," *Supervised Sequence Labelling with Recurrent Neural Networks*, pp. 109–131, 2012.

[21] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proceedings of ICASSP*. IEEE, 2016, pp. 4960–4964.

[22] L. Lu, L. Kong, C. Dyer, N. A. Smith, and S. Renals, "Segmental recurrent neural networks for end-to-end speech recognition," in *Proceedings of Interspeech*, 2016.

[23] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *Proceedings of ASRU*, 2011.

[24] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos *et al.*, "The AMI meeting corpus," in *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, vol. 88, 2005.

[25] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal *et al.*, "The AMI meeting corpus: A pre-announcement," in *International Workshop on Machine Learning for Multimodal Interaction*. Springer, 2005, pp. 28–39.

[26] S. Renals, T. Hain, and H. Bourlard, "Recognition and interpretation of meetings: The AMI and AMIDA projects," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU '07)*, 2007.

[27] V. Peddinti, V. Manohar, Y. Wang, D. Povey, and S. Khudanpur, "Far-field asr without parallel data," in *Proceedings of Interspeech*, 2016.

[28] *Code to reproduce results of the experiments in this paper.*, 2017 (accessed March 23 , 2017), https://github.com/vijayaditya/kaldi/blob/tdnnlstmpaper/egs/swbd/s5c/README.

[29] G. Cheng, V. Peddinti, D. Povey, V. Manohar, S. Khudanpur, and Y. Yan, "An exploration of dropout with lstms," in *Proceedings of Interspeech*, 2017.