INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Multi-Stride Self-Attention for Speech Recognition

Kyu J. Han, Jing Huang, Yun Tang, Xiaodong He, Bowen Zhou

JD AI Research
675 East Middlefield Road
Mountain View, CA 94043, USA
{kyu.han,jing.huang,yun.tang,xiaodong.he,bowen.zhou}@jd.com

Abstract

In contrast to the huge success of self-attention based neural networks in various NLP tasks, the efficacy of self-attention in speech applications is yet limited. This is partly because the full effectiveness of the self-attention mechanism could not be achieved without proper down-sampling schemes in speech tasks. To address this issue, we propose a new self-attention mechanism suitable for speech recognition, namely, *multi-stride self-attention*. The proposed multi-stride approach lets each group of heads in self-attention process speech frames with a unique stride over neighboring frames. Thus, the entire attention mechanism would not be confined in a fixed frame shift and can have diverse contextual views for a given frame to determine attention weights more effectively. To validate our proposal we evaluated it on various speech corpora for speech recognition, both English and Chinese, and observed a consistent improvement, especially in terms of substitution and deletion errors, without the increase of model complexity. The average WER improvement of 7.5% (relative) obtained by the TDNNs having the multi-stride self-attention layer as compared to the baseline TDNN model shows the effectiveness of the proposed multi-stride self-attention mechanism.

Index Terms: multi-stride self-attention, speech recognition, word error rate (WER)

1. Introduction

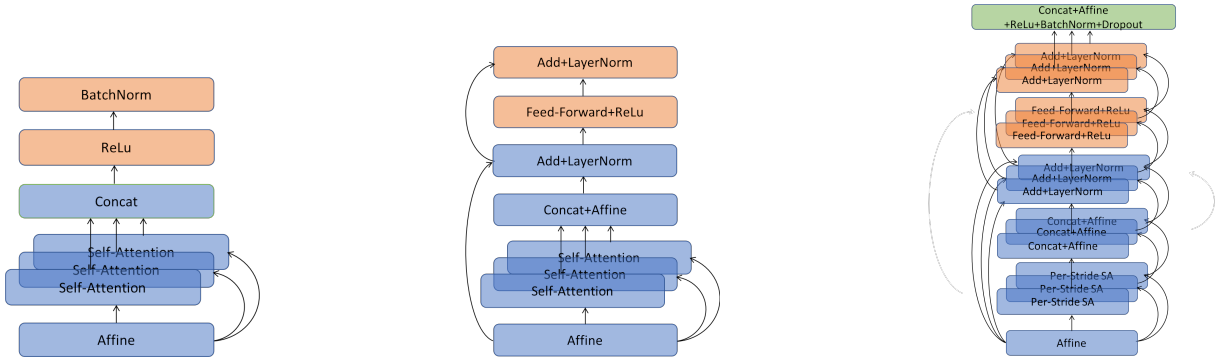
Self-attention has been a huge success in a number of downstream natural language processing (NLP) tasks such as machine translation and question answering since it was introduced in [1, 2]. The basic principle for self-attention follows the common concept discussed in neural Turing machines [3] or neural machine translations [4, 5] or memory networks [6]. That is, for a given key or query vector, attention weights are distributed across surrounding or accessible word vectors to filter out irrelevant information while weighing on relevant one. What differentiates self-attention from the normal attention mechanisms is 1) to self-generate key, query and value vectors through learned projections and 2) to process them with multiple heads to offer diverse perspectives in determining attention weights for surrounding context. With this multi-head self-attention approach, both Transformer and BERT [1, 2] were able to achieve the state-of-the-art performances on many NLP tasks without recurrent units in long short-term memory (LSTM) networks, thus with better parallelization in training.

Inspired by the success of self-attention in the field of NLP, there have been a few efforts lately to utilize the idea in speech recognition. In [7], the authors incorporated the multi-head approach to the well-known Listen, Attend and Spell (LAS) [8] framework for end-to-end speech recognition by extending the single-head attention component to the one with multiple heads, and reported a noticeable WER improvement. More direct application of the self-attention mechanism in the original Transformer paper to speech recognition was presented in [9, 10] in the form of Speech-Transformer. In those works, the almost same encoder-decoder structure used in Transformer was applied to end-to-end speech recognition tasks for English and Chinese, having achieved head-to-head with or even better results than LSTM-based sequence-to-sequence systems. A variant of Transformer was also proposed in [11] where a stack of Transformer's encoder blocks was trained with the Connectionist Temporal Classification (CTC) loss [12], but shown to be still far from the state-of-the-art hybrid DNN/HMM systems trained with the lattice-free maximum mutual information (LF-MMI) objective [13]. In [14, 15], a few tweaks were proposed to address practical issues when applying self-attention to speech recognition. To avoid prohibitively large projection matrices for key, query and value vectors, in [14], down-sampling or reshaping input speech frames via frame concatenation were discussed. In [15], restricting a range of neighboring speech frames with a fixed frame stride to limit the context length and reduce the number of surrounding frames for attention weight computation was introduced.

As addressed in [11, 14, 15], down-sampling speech frames is critical when applying self-attention to speech recognition. It is not only because down-sampling can reduce the entire sequence length to fit computations in memory but also because speech frames with a very short frame shift such as 10ms are highly correlated in a close proximity. Such neighboring frames are hardly considered as a distinct information units such as words or word pieces in NLP, and make the self-attention mechanism hard to effectively distribute attention weights across more informative units like phonemes. To tackle this problem in a different perspective from the previous works, in this paper, we propose a new self-attention mechanism more effective for speech recognition tasks, namely *multi-stride self-attention*. The proposed multi-stride self-attention mechanism processes speech frames with different frame strides in separate pipelines and combines them in a later stage. Thus, it can have the same benefit of down-sampling input frames to mitigate the burden of computations for self-attention, as well as compute attention weights on a diverse range of neighboring frames. To validate our proposed idea we evaluate it on various speech recognition corpora in both English and Chinese. From the evaluations we observe a consistent improvement across data sets, especially when replacing the last layer of the baseline time-delay neural networks (TDNNs) with the multi-stride self-attention layer, without increasing model complexity.

The paper is organized as follows. In Section 2 we introduce the proposed multi-stride self-attention mechanism, in comparison with the original time-restricted self-attention method in [15]. In Section 3 we share the details of the se-

http://dx.doi.org/10.21437/Interspeech.2019-1973



(a) *Time-restricted self-attention* (b) *Self-attention in Transformer* (c) *Multi-stride self-attention*

Figure 1: *Various self-attention mechanisms illustrated for comparison. (a) and (b) are from the literatures in [15, 1], respectively while (c) is our proposed multi-stride self-attention mechanism. Since we use narrower neural layers in the multi-stride approach, there is almost no increase in model complexity.*

tups and data used in the experiments discussed throughout the paper, and present experimental results both when stacking the multi-stride self-attention layers and when the proposed layer is combined with the factorized TDNNs [16] on various evaluation corpora. We conclude the paper in Section 4 with a few remarks on our contributions and notes for future directions.

2. Multi-Stride Self-Attention

The time-restricted self-attention mechanism [15] was motivated by [5] in restricting the context used for attention weight computation within a local region. In this paper, we show how effective our proposed multi-stride approach in this time-restricted setup for self-attention would be for various speech recognition tasks.

2.1. Time-restricted self-attention

To formulate the time-restricted self-attention mechanism in a mathematical manner, we adopt the following expressions from [1, 14, 15]. Given the time-restricted setup, we define an input matrix $\boldsymbol{X} \in \mathbb{R}^{T \times d_{model}}$ where T is the input sequence length restricted by the left and right context (c_l and c_r, respectively), and thus $T = c_l + c_r + 1$. d_{model} is the dimension of embedding vectors in self-attention. Note that the frame stride f_s is applied to down-sample input speech frames, and it was fixed to 3 in [15]. For the projected query, key and value matrices, $\boldsymbol{Q}_i, \boldsymbol{K}_i$ and $\boldsymbol{V}_i$, self-attention for the i^{th} head is computed as follows:

$$\mathit{Head}_i = \text{softmax}\left(\frac{\boldsymbol{Q}_i \boldsymbol{K}_i^T}{\sqrt{d_k}}\right)\boldsymbol{V}_i \tag{1}$$

where $\boldsymbol{Q}_i = \boldsymbol{X}\boldsymbol{W}_i^Q$ and $\boldsymbol{W}_i^Q \in \mathbb{R}^{d_{model} \times d_q}$, $\boldsymbol{K}_i = \boldsymbol{X}\boldsymbol{W}_i^K$ and $\boldsymbol{W}_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $\boldsymbol{V}_i = \boldsymbol{X}\boldsymbol{W}_i^K$ and $\boldsymbol{W}_i^V \in \mathbb{R}^{d_{model} \times d_v}$, d_q, d_k and d_v are the dimensions of query, key and value vectors in self-attention, respectively. The output of the self-attention layer is given in the following form, after ReLu activation [17] and batch normalization [18] on the concatenated vector from the entire n_h heads in self-attention:

$$\mathit{SelfAttention} = \text{BatchNorm}\left(\text{ReLu}\left(\mathit{MultiHead}\right)\right) \tag{2}$$

where $\mathit{MultiHead} = \text{Concat}\left(\mathit{Head}_1, \ldots, \mathit{Head}_{n_h}\right)$. The layer structure of the time-restricted self-attention mechanism is illustrated in Figure 1.(a).

2.2. Self-attention block in Transformer

In its implementation, the time-restricted self-attention mechanism is quite different from the self-attention block introduced in the original Transformer paper [1] or used in [11, 14]. As compared to the original self-attention block depicted in Figure 1.(b), it has batch normalization instead of layer normalization [19] and no position-wise feed-forward network after the multi-head self-attention sublayer. The mathematical formation for the self-attention block considering the time-restricted input matrix $\boldsymbol{X}$ can be written as below:

$$\mathit{MultiHeadProj} = \text{Concat}\left(\mathit{Head}_1, \ldots, \mathit{Head}_{n_h}\right)\boldsymbol{W}^O \tag{3}$$

$$\mathit{MidLayer} = \text{LayerNorm}\left(\mathit{MultiHeadProj} + \boldsymbol{X}\right) \tag{4}$$

$$\mathit{SelfAttention} = \text{LayerNorm}\left(\text{FF}\left(\mathit{MidLayer}\right) + \mathit{MidLayer}\right) \tag{5}$$

where $\boldsymbol{W}^O$ is the projection matrix for the concatenated multi-head vectors and FF is the position-wise feed-forward network, i.e., $\text{FF} = \text{Affine}\left(\text{ReLu}\left(\text{Affine}\left(\mathit{MidLayer}\right)\right)\right)$, with the hidden layer dimension d_{ff}. Note that there are skip connections involved for both *MidLayer* and *SelfAttention* in Eqs. (4) and (5).

2.3. Multi-stride approach

Down-sampling input speech frames is critical in self-attention for speech applications not only to fitting computations in memory but also to the effectiveness of the self-attention mechanism. Speech frames with very short frame shifts are highly correlated and could cause inefficient attention weight distribution. However it would be challenging to tackle this issue with such rigid approaches as simple down-sampling or frame reshaping in [11, 14, 15]. This requires more flexible consideration of how to down-sample speech frames. In this subsection, we introduce our proposal to address the issue, namely *multi-stride self-attention*.

The proposed multi-stride self-attention mechanism, illustrated in Figure 1.(c), processes speech frames in the framework of the self-attention block discussed in Section 2.2, in the following way:

1. Accepts speech frames with a different frame stride per group of heads. We use three different frame strides ($n_s = 3$ and $f_s = 1, 3, 5$) with the fixed context lengths ($c_l = c_r = 5$), thus have different context scopes (i.e., $5, 15, 25$ for $f_s = 1, 3, 5$, respectively). For each group of heads s, a unique input matrix $\boldsymbol{X}^s$ is thus processed.

2. Computes attention weights within each group of heads s, pipelining the process up to the layer normalization after the point-wise feed-forward network.

$$\mathit{Head}_i^s = \text{softmax}\left(\frac{\boldsymbol{Q}_i^s \boldsymbol{K}_i^{s\,T}}{\sqrt{d_k}}\right)\boldsymbol{V}_i^s \tag{6}$$

$$\mathit{MultiHeadProj}^s = \text{Concat}\left(\mathit{Head}_1^s, \ldots, \mathit{Head}_{n_h}^s\right)\boldsymbol{W}^{O_s} \tag{7}$$

$$\mathit{MidLayer}^s = \text{LayerNorm}\left(\mathit{MultiHeadProj}^s + \boldsymbol{X}^s\right) \tag{8}$$

$$\mathit{Stride}^s = \text{LayerNorm}\left(\text{FF}\left(\mathit{MidLayer}^s\right) + \mathit{MidLayer}^s\right) \tag{9}$$

3. Concatenates the processed vector from each pipeline and performs the affine transform, ReLu activation, batch normalization and dropout [20] to generate the final output vector with the dimension of d_{model}.

$$\mathit{MultiStride} = \text{Concat}\left(\mathit{Stride}^1, \ldots, \mathit{Stride}^{n_s}\right)\boldsymbol{W}^O \tag{10}$$

$$\mathit{SelfAttention} = \text{Dropout}\left(\text{BatchNorm}\left(\text{ReLu}\left(\mathit{MultiStride}\right)\right)\right) \tag{11}$$

To prevent the increase of model complexity, we keep the total number of heads in self-attention as is, by assigning n_h/n_s heads per pipeline corresponding to a given frame stride as well as reducing the hidden layer dimension of the position-wise feed-forward networks to half. More details on the hyper-parameters in the multi-stride self-attention mechanism will be presented in Section 3 when discussing the experimental setups and results.

3. Experimental Results and Discussions

In this section we validate the effectiveness of our proposed multi-stride self-attention mechanism on various speech corpora for speech recognition, both in English and Chinese.

3.1. Experimental setups and data

For the experiments discussed in the paper, we considered various evaluation data sets in TED-LIUM [21] for English broadcast speech in TED talks, HUB5 [22] for English telephone speech, Librispeech [23] for English read speech and AISHELL-2 iOS [24] for Mandarin Chinese read speech on the mobile channel condition. To train each system we followed Kaldi's example recipe [25] with the corresponding training data up to the speaker adaptive training stages for Gaussian mixture models. Then, we trained different neural network acoustic models having the lattice alignments given by the GMMs as soft targets. For the telephony models, we used the 300hr Switchboard corpus [26] for acoustic modeling and added the Fisher data [27] to strengthen language models. To train language models we used the SRILM tooklit [28]. The first-pass decoding was conducted with the 3-gram LMs and the resultant lattices were rescored with the larger LMs with 4-grams later in the second-pass.

The LF-MMI objective was used to train the neural network acoustic models with the three regularization methods in cross-entropy, $L2$ and leaky HMM [13] . The gradual decrease of learning rates were exploited from $1.0e^{-4}$ to $1.0e^{-5}$ to make the entire training procedure stable and have better convergence. The trainings were conducted on either 8 or 16 GPUs on the Nvidia P40 chips, depending upon the size of the training materials. The number of nodes in the final layer of each model should depend upon the number of tri-phone states in the corresponding HMM per corpus, ranging from 3.5K to 6K after tree clustering.

Table 1: *WER (in %) comparison of the time-restricted self-attention layer in [15] and the self-attention block in Transformer [1] for speech recognition on the TED-LIUM dev set [21]. SA: Self Attention.*

Number of SA Layers	1	2	3	5
Time-Restricted SA	11.0	10.3	9.6	9.3
–BatchNorm & +LayerNorm	10.9	9.9	9.4	9.1
Transformer-like SA Block	**10.1**	**9.0**	**8.8**	**8.5**

Table 2: *WER (in %) comparison of the fixed stride approaches and the proposed multi-stride self-attention mechanism for speech recognition on the TED-LIUM dev set. The multi-stride self-attention layer has as similar model complexity of 5M parameters as the fixed-stride self-attention layers have. The context lengths are fixed in all the cases ($c_l = c_r = 5$). SA means the Transformer-like SA block here.*

Number of SA Layers	1	2	3	5
Frame Stride $f_s = 1$	10.2	9.1	8.6	8.3
Frame Stride $f_s = 3$	10.1	9.0	8.8	8.5
Frame Stride $f_s = 5$	10.2	9.1	8.8	8.4
Multi-Stride $f_s = 1, 3, 5$	**9.5**	**8.6**	**8.4**	**8.1**

3.2. Ablation on multi-stride self-attention

We first conducted the ablation tests on the proposed multi-stride self-attention mechanism in comparison with the original time-restricted self-attention method in [15]. Tables 1 and 2 show the experimental comparisons on the TED-LIUM dev set, as we stack the self-attention layers up to 5. For the experiments, we set $n_h = 12, d_q = d_k = 40, d_v = 60, d_{model} = 256, d_{ff} = 1,024, c_l = c_r = 5, f_s = 3$. For the multi-stride cases, we set $d_{ff} = 512$ not to increase the model complexity, and $f_s = 1, 3, 5$.

Table 1 presents the comparative results between the original time-restricted self-attention layer and the self-attention block in Transformer [1]. In Section 2.2, we pointed out the differences between them in terms of the normalization method and the position-wise feed-forward network with the skip connections. As for the normalization method, the results show layer normalization would not make a significant improvement over batch normalization, although it gives a marginal WER boost consistently as we stack the self-attention layers. In contrast, the self-attention block in Eq. (5), similar with the one in Transformer's encoder stack [1][1], worked well in this time-restricted setup for self-attention. The average improvement of roughly 10% (relative) across stacked self-attention layers seems to come from adding the position-wise feed-forward network and skip connections. Our proposed multi-stride self-attention mechanism has this self-attention block structure as the base.

Table 2 shows the effectiveness of the proposed multi-stride self-attention mechanism for speech recognition in comparison with the fixed frame stride cases. While the different frame strides would not make any big impact individually, the multi-

[1]This is not exactly the same with the encoder block in Transformer as we use the relative positional encoding by identifying relative positions using one-hot vectors in the time-restricted self-attention framework while the sinusoidal positional encoding was exploited in [1].

Table 3: *WER (in %) comparison of the TDNN-F acoustic models equipped with the self-attention layers on various evaluation data sets. TDNN-F with the 10M parameters is the factorized TDNN baseline model with 6 layers. TR-SA: Time-Restricted Self-Attention [15], MS-SA: Multi-Stride Self-Attention.*

Acoustic Model	TED-LIUM		HUB5		Librispeech				AISHELL-2	
	DEV	TEST	SWBD	ALL	DEV		TEST		DEV	TEST
					Clean	Other	Clean	Other		
TDNN-F (10M)	8.2	8.7	10.0	14.9	3.8	9.7	4.1	10.1	9.4	9.2
+TR-SA (11M)	8.0	8.4	9.5	14.0	3.6	9.3	4.0	9.4	8.9	8.7
+MS-SA (11M)	**7.7**	**8.0**	**9.1**	**13.4**	**3.5**	**9.3**	**3.9**	**9.3**	**8.4**	**8.5**

Table 4: *Averaged WER (in %) comparison of the time-restricted self-attention layer and the proposed multi-stride self-attention layer on top of the TDNN-F model over the conversational evaluation sets in TED-LIUM and HUB5.*

	Sub.	Del.	Ins.	Total
TDNN-F	6.3	3.0	1.4	10.7
+TR-SA	6.0	2.9	1.3	10.2
+MS-SA	**5.7**	**2.6**	1.3	**9.6**

stride case where the three frame strides are all considered together produced noticeable improvements across stacked self-attention layers. Given that the model complexities remain almost same for all the experiments with the same number of layers in the table, we presume that these improvements result from the diverse contextual views provided by the proposed approach to the self-attention mechanism. Hence the entire self-attention mechanism would get benefited in terms of distributing attention weights effectively over much longer span of speech frames with dynamic strides and result in capturing phonemic activities more correctly. This claim can be supported as well in the extended experiments with the factorized TDNNs shown in Table 3.

3.3. Multi-stride self-attention with TDNN-F

Table 3 shows the performance comparison between the self-attention mechanisms on the baseline TDNN-F (factorized TDNN) model across various evaluation data sets. The baseline model is the 6-layer TDNN-F with the number of parameters in 10M and we replaced the last layer only with either time-restricted or multi-stride self-attention layer to better understand how much each mechanism would influence to improve speech recognition accuracy in the existing framework of TDNN-F neural layers. This is the similar evaluation setup used in [15], where 6 or 7-layer TDNNs were considered as a baseline neural network model architecture. Note that replacing the last layer with the self-attention layers increased the model complexity by 1M parameters in both of the self-attention mechanisms, with which we safely assume that any performance difference between the self-attention mechanisms on top of the baseline model would come from the discrepancy of the effectiveness in the two methods.

The effectiveness of the proposed multi-stride self-attention mechanism is clearly shown in the table. Comparing the baseline model and the model equipped with the multi-stride self-attention layer highlights the consistent WER improvement by the proposed method across various evaluation data sets over two languages, even without much increase in model complexity. The overall performance enhancement by average 7.5% WER (relative) from the TDNN-F models with the multi-stride self-attention layer as compared to the baseline TDNN-F model with no self-attention layer verifies that the proposed method can be generalized quite well even in different languages for speech recognition. This improvement is shown more outstandingly in the conversational data sets such as TED-LIUM and HUB5, for which Table 4 has more detailed analyses on the WERs. The proposed method seems to improve WERs in terms of substitution and deletion errors, which is along the same line with our claim that the multi-stride approach can capture phonemic activities more correctly with dynamic consideration of surrounding speech frames and it could be signified in more conversational speech.

4. Conclusions

In this paper we proposed the multi-stride self-attention mechanism suitable for speech recognition, as a flexible downsampling approach to accommodate diverse frame strides and context spans for effective attention weight computation in speech applications. With experiments on various evaluation data sets in different acoustic conditions over two languages in English and Chinese, we validated that our proposed method consistently improves the WER performance of neural network acoustic models over the baseline, without increasing the model complexity.

Motivated by these promising observations, we plan to extend our research effort to end-to-end speech recognition systems with the proposed multi-stride self-attention mechanism. Incorporating the multi-stride approach to self-attention in Transformer-like encoder-decoder architectures would be a reasonable first step toward that direction.

5. References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Neural Information Processing Systems*, 2017, pp. 5998–6008.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional Transformers for language understanding," 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[3] A. Graves, G. Wayne, and I. Danihelka, "Neural Turing machines," 2014. [Online]. Available: http://arxiv.org/abs/1410.5401

[4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations*, 2015. [Online]. Available: http://arxiv.org/abs/1409.0473

[5] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Conference on Empirical Methods in Natural Language Processing*, 2015. [Online]. Available: https://arxiv.org/abs/1508.04025

[6] J. Weston, S. Chopra, and A. Bordes, "Memory networks," 2014. [Online]. Available: http://arxiv.org/abs/1410.3916

[7] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," in *International Conference on Acoustics, Speech, and Signal Processing*, 2018. [Online]. Available: https://arxiv.org/abs/1712.01769

[8] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, Attend and Spell: A neural network for large vocabulary conversational speech recognition," in *International Conference on Acoustics, Speech, and Signal Processing*, 2016. [Online]. Available: https://arxiv.org/abs/1508.01211

[9] L. Dong, S. Xu, and B. Xu, "Speech-Transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *International Conference on Acoustics, Speech, and Signal Processing*, 2018. [Online]. Available: https://arxiv.org/abs/1901.10055

[10] S. Zhou, L. Dong, S. Xu, and B. Xu, "Syllable-based sequence-to-sequence speech recognition with the Transformer in Mandarin Chinese," in *Interspeech*, 2018. [Online]. Available: https://arxiv.org/abs/1804.10752

[11] J. Salazar, K. Kirchhoff, and Z. Huang, "Self-attention networks for connectionist temporal classification in speech recognition," in *International Conference on Acoustics, Speech, and Signal Processing*, 2019. [Online]. Available: https://arxiv.org/abs/1901.10055

[12] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *International Conference on Machine Learning*, 2006, pp. 369–376.

[13] D. Povey, V. Peddinti, D. Galvez, P. Ghahrmani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Interspeech*, 2016, pp. 2751–2755.

[14] M. Sperber, J. Niehues, G. Neubig, S. Stuker, and A. Waibel, "Self-attention acoustic models," in *Intespeech*, 2018. [Online]. Available: https://arxiv.org/abs/1803.09519

[15] D. Povey, H. Hadian, P. Ghahremani, K. Li, and S. Khudanpur, "A time-restricted self-attention layer for ASR," in *International Conference on Acoustics, Speech, and Signal Processing*, 2018.

[16] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohamadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," in *Interspeech*, 2018.

[17] M. D. Zeiler, M. Ranzato, R. Monga, M. Z. Mao, K. Yang, Q. V. Le, P. Nguyen, A. W. Senior, V. Vanhoucke, J. Dean, and G. E. Hinton, "On rectified linear units for speech processing," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 3517–3521.

[18] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456. [Online]. Available: https://arxiv.org/abs/1806.02375

[19] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016. [Online]. Available: https://arxiv.org/abs/1607.06450

[20] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012. [Online]. Available: http://arxiv.org/abs/1207.0580

[21] A. Rousseau, P. Deléglise, and Y. Esteve, "Enhancing the TED-LIUM corpus with selecting data for language modeling and more TED talks," in *International Conference on Language Resources and Evaluation*, 2014.

[22] J. F. William, W. M. Fisher, A. F. Martin, M. A. Przybocki, and D. S. Pallett, "NIST evaluation of conversational speech recognition over the telephone: English and Mandarin performance results," in *NIST*, 2000.

[23] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: An ASR corpus based on public domain audio books," in *International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 5206–5210.

[24] J. Du, X. Na, X. Liu, and H. Bu, "AISHELL-2: Transforming Mandarin ASR research into industrial scale," 2018. [Online]. Available: https://arxiv.org/abs/1808.10583

[25] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proc of ASRU 2011*, 2011.

[26] J. Godfrey, E. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 1992, pp. 517–520.

[27] C. Cieri, D. Miller, and K. Walker, "The Fisher corpus: A resource for the next generations of speech-to-text," in *International Conference on Language Resources and Evaluation*, 2004, pp. 69–71.

[28] A. Stolcke, "SRILM – An extensible language modeling toolkit," in *Proc of ICSLP 2002*, 2002, pp. 901–904.