Efficient conformer-based speech recognition with linear attention

Shengqiang Li, Menglong Xu, Xiao-Lei Zhang
CIAIC, School of Marine Science and Technology, Northwestern Polytechnical University, China
E-mail: shengqiangli,mlxu@mail.nwpu.edu.cn, xiaolei.zhang@nwpu.edu.cn

***Abstract*—Recently, conformer-based end-to-end automatic speech recognition, which outperforms recurrent neural network based ones, has received much attention. Although the parallel computing of conformer is more efficient than recurrent neural networks, the computational complexity of its dot-product self-attention is quadratic with respect to the length of the input feature. To reduce the computational complexity of the self-attention layer, we propose multi-head linear self-attention for the self-attention layer, which reduces its computational complexity to linear order. In addition, we propose to factorize the feed forward module of the conformer by low-rank matrix factorization, which successfully reduces the number of the parameters by approximate 50% with little performance loss. The proposed model, named *linear attention based conformer* (LAC), can be trained and inferenced jointly with the connectionist temporal classification objective, which further improves the performance of LAC. To evaluate the effectiveness of LAC, we conduct experiments on the AISHELL-1 and LibriSpeech corpora. Results show that the proposed LAC achieves better performance than 7 recently proposed speech recognition models, and is competitive with the state-of-the-art conformer. Meanwhile, the proposed LAC has a number of parameters of only 50% over the conformer with faster training speed than the latter.**

I. Introduction

End-to-end automatic speech recognition has received more and more attention in both academia and industry. It combines acoustic model, pronunciation model, and language model into a single neural network. It achieves competitive results with conventional speech recognition, and simplifies the training and decoding pipelines. There are mainly three popular end-to-end approaches, named connectionist temporal classification (CTC) [1], recurrent neural network transducer [2], [3], and attention based encoder-decoder [4], [5]. This paper focuses on the attention based encoder-decoder model. There are two main kinds of attention based encoder-decoder models. The early models use recurrent neural networks as the building blocks of its encoder and decoder. Recently, a transformer architecture with a multi-head self-attention mechanism has become prevalent because of its high accuracy and fast training speed.

Although transformer is good at capturing content-based global interactions and can be trained in parallel, the computational complexity of its dot-product self-attention is quadratic with respect to the input feature length, which requires large GPU memory and long training time. Several studies have been conducted to reduce the computational complexity of the dot-product self-attention [6], [7], [8], [9], [10], [11], [12]. In [6], Fujita *et al.* replaced the self-attention layer with lightweight and dynamic convolution, where the convolution weight is dynamically predicted through an additional linear layer. In [7], [8], choromanski *et al.* introduced performer, which uses a novel fast attention via positive orthogonal random features approach (FAVOR+). Its complexity scales linearly rather than quadratically with respect to the number of the elements of an input sequence. Wang *et al.* [9] proposed to adopt the performer for efficiency improvement. In our previous work [10], we proposed local dense synthesizer attention which restricts the length of the attention weights for the reduction of the storage and computational complexity.

On the other hand, many works have shown that speech recognition models with small sizes can perform comparably to large models. For example, QuartzNet [13], which uses one-dimension separable convolution based models trained with the CTC loss, achieves the state-of-the-art accuracy on LibriSpeech. Winata *et al.* [14] adopted a low-rank matrix factorization to compress models. The compressed models suffer little performance degradation. ContextNet [15] combined a recurrent neural network transducer decoder with a fully convolutional encoder, where the encoder incorporates global context information into its convolution layers by squeeze-and-excitation modules.

In this paper, motivated by [11] on the object detection task in video processing, we propose to apply the efficient attention [11] to conformer and further extend it to multi-head version, named multi-head linear self-attention (MHLSA), for speech recognition. Specifically, MHLSA replaces the dot-product self-attention in the encoders of the conformer. It is functionally equivalent to the dot-product self-attention, but with substantially less computational costs than the latter, given that MHLSA does not need to calculate the correlation matrix in the dot-product self-attention.

To further reduce the model size of conformer, we factorize the feed forward module in both the encoder and decoder by low-rank matrix factorization. The overall model is named *linear attention based conformer* (LAC). To improve the performance of LAC, we train and decode LAC jointly with the CTC objective. We conduct experiments on the AISHELL-1 [16] and LibriSpeech [17] corpora. Experimental results show that the proposed LAC achieves a character error rate (CER) of 5.02% on AISHELL-1, and a word error rate (WER) of 2.1% and 2.3% on the 'dev-clean' and 'test-other' sets of LibriSpeech respectively, which is competitive with the conformer [18]. Meanwhile, LAC has a number of parameters

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on August 28,2022 at 14:48:14 UTC from IEEE Xplore. Restrictions apply.

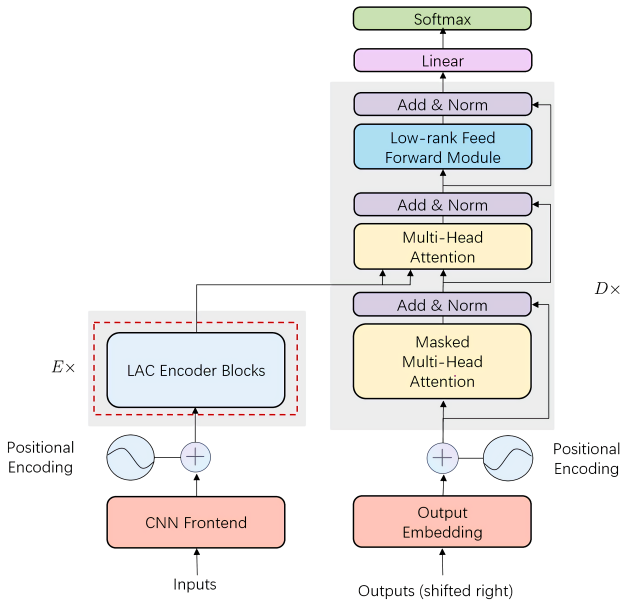


Fig. 1: The model architecture of LAC.

of only 50% over the conformer, and accelerates the training speed by 1.23 times and 1.18 times on AISHELL-1 and LibriSpeech respectively.

The remainder of this paper is organized as follows. Section II describes the architecture of the proposed LAC. Section III presents experiments. Conclusion is given in Section IV.

II. Linear attention based conformer

The architecture of the proposed LAC is shown in Figure 1. It includes a convolution neural network (CNN) frontend, a stack of encoders, and a stack of decoders. In this section, we focus on describing the CNN frontend in Section II-A and the encoder blocks of LAC in Section II-B, as emphasized in the red dot box of Figure 1. Section II-C analyzes the complexity of the linear attention.

We will not describe the decoders of LAC anymore, given that the multi-head attention in the decoders has been described in detail in [5], and also the low-rank feed forward module in the decoders is similar to the low-rank feed forward module in the encoders.

A. CNN frontend

First, after passing through two CNN layers and an embedding layer, the input acoustic features of an utterance are transformed into a feature matrix $\mathbf{X}^{emb} \in \mathbb{R}^{T \times d_e}$, where T is the sequence length after subsampling, and d_e is the dimension of the embedding layer. Because our MHLSA does not need to compute the correlation matrix, we replace the relative position encoding in the conformer by absolute position encoding [5]. In this work, the sine and cosine functions are used for the

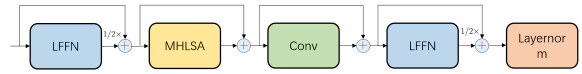


Fig. 2: The encoder architecture of LAC.

absolute position encoding $\mathbf{P} \in \mathbb{R}^{T \times d_e}$, which is defined as:

$$\begin{aligned} p_{i,2j} &= \sin\left(i/10000^{2j/d_e}\right) \\ p_{i,2j+1} &= \cos\left(i/10000^{2j/d_e}\right) \end{aligned} \tag{1}$$

where i is the position, and j is the dimension. Then, the position encoding $\mathbf{P}$ is added to the feature matrix $\mathbf{X}^{emb}$:

$$\mathbf{X}^{pos} = \mathbf{X}^{emb} + \mathbf{P} \tag{2}$$

Finally, the matrix $\mathbf{X}^{pos}$ passes through a stack of LAC encoder blocks for the output of the encoder.

B. Linear attention based conformer encoder

The distinctive feature of our model is the use of LAC encoder blocks. Each LAC encoder block contains two low-rank feed forward (LFFN) modules sandwiching the MHLSA module and the convolution (Conv) module, as shown in Figure 2.

Given an input $\mathbf{X}_i$ to the i-th encoder block, the output $\mathbf{Y}_i$ of the block is:

$$\tilde{\mathbf{X}}_i = \mathbf{X}_i + \frac{1}{2}\,\mathrm{LFFN}\left(\mathbf{X}_i\right) \tag{3}$$

$$\mathbf{X}'_i = \tilde{\mathbf{X}}_i + \mathrm{MHLSA}\left(\tilde{\mathbf{X}}_i\right) \tag{4}$$

$$\mathbf{X}''_i = \mathbf{X}'_i + \mathrm{Conv}\left(\mathbf{X}'_i\right) \tag{5}$$

$$\mathbf{Y}_i = \mathrm{Layernorm}\left(\mathbf{X}''_i + \frac{1}{2}\,\mathrm{LFFN}\left(\mathbf{X}''_i\right)\right) \tag{6}$$

where $\mathrm{LFFN}(\cdot)$ refers to the low-rank feed forward module, $\mathrm{MHLSA}(\cdot)$ refers to the multi-head linear self-attention module, $\mathrm{Conv}(\cdot)$ refers to the convolution module, and $\mathrm{Layernorm}$ refers to layer normalization. The convolution module is the same as that in [19]. It starts with a pointwise convolution layer and a gated linear unit, followed by a 1-D depthwise convolution layer. Batch normalization is applied after the convolution layers. In the following subsections, we will introduce the proposed $\mathrm{LFFN}(\cdot)$ and $\mathrm{MHLSA}(\cdot)$ modules respectively.

1) Low-rank feed forward module: Each of the layers in the encoder or decoder contains multiple feed forward modules. Each feed forward module consists of two linear transformations with an activation between them. Originally, the feed forward module is formulated as follows. Given an input $\mathbf{X}$ to the feed forward module, the output is:

$$\mathrm{FFN}(\mathbf{X}) = \mathrm{Dropout}(\mathrm{Swish}(\mathbf{X}\mathbf{W}_1))\mathbf{W}_2 \tag{7}$$

where $\mathrm{Swish}(\cdot)$ refers to the Swish activation, $\mathrm{Dropout}(\cdot)$ refers to the dropout operator, $\mathbf{W}_1 \in \mathbb{R}^{d \times d_{ff}}$, $\mathbf{W}_2 \in \mathbb{R}^{d_{ff} \times d}$ are the weight matrices of the two linear layers, d and d_{ff}

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on August 28,2022 at 14:48:14 UTC from IEEE Xplore. Restrictions apply.

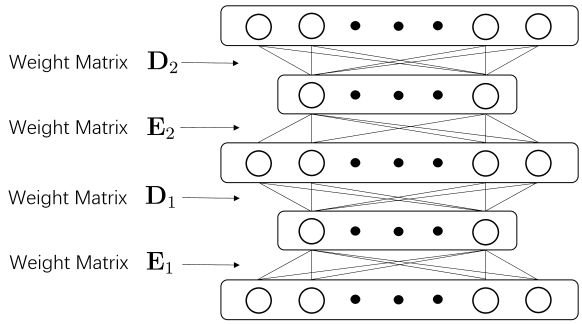


Fig. 3: The low-rank feed forward module.

denote the output dimension and hidden dimension of the feed forward module respectively.

To reduce the model size of the feed forward module, we factorize it by low-rank matrix factorization. Specifically, as shown in Figure 3, we factorize the weight matrix of the original linear layer $\mathbf{W}$ by two small matrices $\mathbf{E}$ and $\mathbf{D}$:

$$\mathrm{LFFN}(\mathbf{X}) = \mathrm{Dropout}(\mathrm{Swish}(\mathbf{X}\mathbf{E}_1\mathbf{D}_1))\mathbf{E}_2\mathbf{D}_2 \tag{8}$$

where $\mathbf{E}_1 \in \mathbb{R}^{d \times d_{bn}}$ and $\mathbf{D}_1 \in \mathbb{R}^{d_{bn} \times d_{ff}}$ are the factorized matrices of $\mathbf{W}_1$, $\mathbf{E}_2 \in \mathbb{R}^{d_{ff} \times d_{bn}}$ and $\mathbf{D}_2 \in \mathbb{R}^{d_{bn} \times d}$ are the factorized matrices of $\mathbf{W}_2$, and d_{bn} denotes the dimension of the bottleneck layer in the low-rank feed forward module.

Comparing the feed forward module with the low-rank feed forward module, we see that, the matrix $\mathbf{W}$ requires $d \times d_{ff}$ parameters and $d \times d_{ff}$ flops, while $\mathbf{E}$ and $\mathbf{D}$ only require $d_{bn} \times (d + d_{ff})$ parameters and flops. Because d_{bn} is much smaller than d_{ff} and d, the number of the parameters and flops of $\mathbf{E}$ and $\mathbf{D}$ are much smaller than that of $\mathbf{W}$.

2) Multi-head linear self-attention: We first revisit the multi-head dot-product self-attention in the conventional conformer as follows. For the h-th head of the multi-head dot-product self-attention, we pass the input $\mathbf{X} \in \mathbb{R}^{T \times d_m}$ through three linear projection layers which derives the query matrix $\mathbf{Q}_h \in \mathbb{R}^{T \times d_k}$, key matrix $\mathbf{K}_h \in \mathbb{R}^{T \times d_k}$, and value matrix $\mathbf{V}_h \in \mathbb{R}^{T \times d_k}$ by:

$$\mathbf{Q}_h = \mathbf{X}\mathbf{W}^{\mathrm{Q}_h}, \mathbf{K}_h = \mathbf{X}\mathbf{W}^{\mathrm{K}_h}, \mathbf{V}_h = \mathbf{X}\mathbf{W}^{\mathrm{V}_h} \tag{9}$$

where $\mathbf{W}^{\mathrm{Q}_h}, \mathbf{W}^{\mathrm{K}_h}, \mathbf{W}^{\mathrm{V}_h} \in \mathbb{R}^{d_m \times d_k}$ are the weight matrices of the three linear projection layers of the h-th head, T is the length of the input feature, and d_m is the hidden dimension of the multi-head self-attention layer. Assuming that there are H heads in the multi-head self-attention layer, then $d_k = d_m/H$ is the dimension of each head. The output of the h-th dot-product self-attention module is

$$\mathrm{Attention}(\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h) = \mathrm{softmax}\left(\frac{\mathbf{Q}_h\mathbf{K}_h^{\mathrm{T}}}{\sqrt{d_k}}\right)\mathbf{V}_h \tag{10}$$

The dot-product self-attention based multi-head self-attention is formulated as:

$$\mathrm{MHSA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{Concat}\left(\mathbf{U}_1, \cdots, \mathbf{U}_H\right)\mathbf{W}^{\mathrm{O}} \tag{11}$$

where

$$\mathbf{U}_h = \mathrm{Attention}\left(\mathbf{X}\mathbf{W}^{\mathrm{Q}_h}, \mathbf{X}\mathbf{W}^{\mathrm{K}_h}, \mathbf{X}\mathbf{W}^{\mathrm{V}_h}\right) \tag{12}$$

and $\mathbf{W}^{\mathrm{O}} \in \mathbb{R}^{d_m \times d_m}$ is the parameter matrix of the output projection layer.

To reduce the computational complexity of the self-attention layer to linear order, we first replace the dot-product self-attention by linear (self-)attention, and then extends the linear attention to MHLSA. Specifically, for the h-th head in MHLSA, the input $\mathbf{X} \in \mathbb{R}^{T \times d_m}$ is transformed to the query matrix $\mathbf{Q}_h \in \mathbb{R}^{T \times d_k}$, key matrix $\mathbf{K}_h \in \mathbb{R}^{T \times d_k}$, and value matrix $\mathbf{V}_h \in \mathbb{R}^{T \times d_k}$ by three linear projection layers respectively. The output of the h-th head MHLSA is

$$\mathrm{LinearAtt}(\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h) = \sigma_{\mathrm{row}}\left(\frac{\mathbf{Q}_h}{d_k^{\frac{1}{4}}}\right)\left(\sigma_{\mathrm{col}}\left(\frac{\mathbf{K}_h}{d_k^{\frac{1}{4}}}\right)^{\mathrm{T}}\mathbf{V}_h\right) \tag{13}$$

where $\sigma_{\mathrm{row}}(\cdot)$ and $\sigma_{\mathrm{col}}(\cdot)$ denote the operators of applying the softmax function along the rows and columns of a matrix respectively, and ^{T} denotes the transpose operator. Finally, MHLSA is formulated as:

$$\mathrm{MHLSA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{Concat}\left(\mathbf{Z}_1, \cdots, \mathbf{Z}_H\right)\mathbf{W}^{\mathrm{O}} \tag{14}$$

where

$$\mathbf{Z}_h = \mathrm{LinearAtt}(\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h). \tag{15}$$

C. Computational complexity analysis

As shown in Figure 4(a), because the dot-product self-attention needs to calculate an attention weight matrix $\mathbf{S} \in \mathbb{R}^{T \times T}$, the computational complexity of each head is as high as $O\left(T^2 d_k\right)$.

As a comparison, the linear attention, which is described in Figure 4(b), switches the matrix multiplication order from $\left(\mathbf{Q}\mathbf{K}^{\mathrm{T}}\right)\mathbf{V}$ to $\mathbf{Q}\left(\mathbf{K}^{\mathrm{T}}\mathbf{V}\right)$, which reduces the computational complexities of each head from $O\left(T^2 d_k\right)$ to $O\left(T d_k^2\right)$ without affecting the effectiveness of the matrix multiplication. One can see that the linear attention is mathematically equivalent to the dot-product self-attention, however, it is substantially faster and needs less memory than the latter.

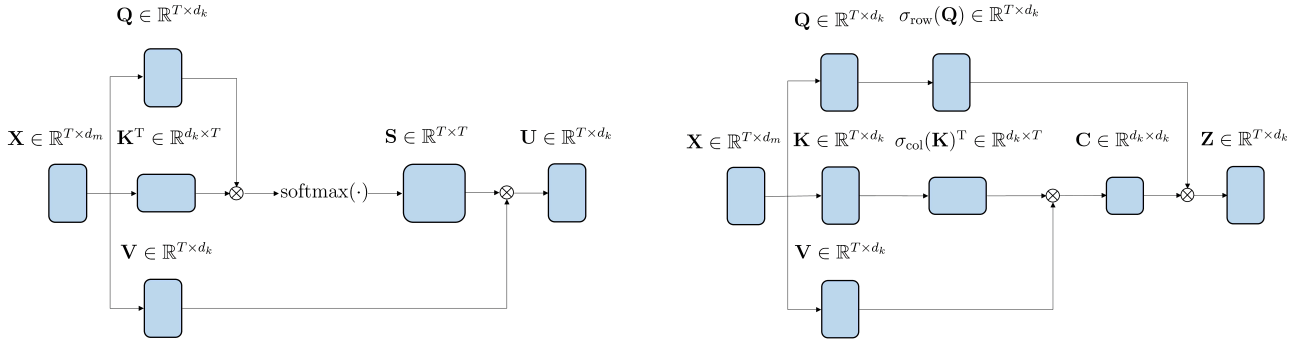
III. Experiments

A. Experiment Setup

Our experiments were conducted on a Mandarin speech corpus AISHELL-1 [16] and an English speech corpus LibriSpeech [17]. The former has 170 hours labeled speech, while the latter consists of 970 hours labeled speech and an additional 800M word token text-only corpus for building language model.

We used 80-channel log-mel filterbank coefficients (Fbank) features computed on a 25ms window with a 10ms shift. The features for each speaker were rescaled to have zero mean and unit variance. The token vocabulary of AISHELL-1 contains 4231 characters. We used a 5000 token vocabulary based on the byte pair encoding algorithm [20] for LibriSpeech. Moreover, the vocabularies of AISHELL-1 and LibriSpeech have

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on August 28,2022 at 14:48:14 UTC from IEEE Xplore. Restrictions apply.



(a) Dot-product self-attention. (b) Linear self-attention

Fig. 4: Illustration of the architecture of dot-product self-attention and linear self-attention. T denotes transpose operator, σ_{row} and σ_{col} refer to normalization functions operating in row and column of matrix $\mathbf{Y}$ respectively.

a padding symbol '$\langle PAD\rangle$' , an unknown symbol '$\langle UNK\rangle$', and an end-of-sentence symbol '$\langle EOS\rangle$'.

Our LAC model contains 12 encoder blocks and 6 decoder blocks. There are 4 heads in both the multi-head linear self-attention and the encoder-decoder attention. The 2D-CNN frontend utilizes two 3×3 convolution layers with 256 channels. The rectified linear units were used as the activation. The stride was set to 2. The hidden dimension of the attention layer is 256. The hidden dimension and output dimension of the feed forward layer are 256 and 2048 respectively. We used the Adam optimizer and a transformer learning rate schedule [5] with 30000 warm-up steps and a peak learning rate of 0.0005. We used SpecAugment [21] for data augmentation. We set the CTC weight to 0.3 for the joint training with the attention model. In the test stage, we set CTC weight to 0.6 for the joint decoding. We used a transformer-based language model to refine the results.

We choose 7 representative transformer-based speech recognition models as the comparison methods so that we can fairly compare our model with other state-of-the-art models in the computational complexity of the self-attention, number of parameters, CER and WER. The comparison methods are the speech transformer (ST) [22], local dense synthesizer attention (LDSA) [10], low-rank transformer (LRT) [14], dynamic convolution (DC) [6], self-attention with 2D dynamic convolution (SA-DC2D) [6], performers in conformer (PIC) [9], and conformer [18]. Speech transformer adopts the transformer architecture as its encoder network and prediction network. LRT applies matrix decomposition to the transformer architecture. LDSA replaces the self-attention in transformer encoder with the local dense synthesizer attention. PIC adopts the performer to improve the efficiency of conformer. DC and SA-DC2D apply dynamic convolution to transformer as an alternative architecture to the self-attention module. Conformer combines convolution neural networks and transformers to model both local and global dependencies. To compare fairly with conformer, we implemented an absolute position encoding based conformer whose parameter settings are the same with LAC except the two novel points, i.e. the attention module in the encoder and the low-rank feed forward module.

TABLE I: Comparison results on AISHELL-1. The symbol 'T' represents the length of input feature. The symbol 'c' in LDSA denotes the context width. The term '# Param.' is short for the number of parameters.

Model	Complexity	# Param.	CER (%)	
			Dev set	Test set
ST [22]	$O(T^2)$	33.91M	6.57	7.37
LRT [14]	$O(T^2)$	12.70M	-	13.09
LDSA [10]	$O(Tc)$	52.5M	5.79	6.49
Conformer [18]	$O(T^2)$	45.15M	4.52	4.88
LAC	$O(T)$	22.83M	4.73	5.02

TABLE II: Comparison results on LibriSpeech.

Model	Complexity	# Param.	WER(%)			
			Dev		Test	
			Clean	Other	Clean	Other
DC [6]	$O(T)$	-	3.5	10.5	3.6	10.8
SA-DC2D [6]	$O(T^2)$	-	3.5	9.6	3.9	9.6
PIC [9]	$O(T)$	10.4M	2.9	7.2	3.1	7.7
Conformer [18]	$O(T^2)$	45.16M	2.1	5.5	2.3	5.5
LAC	$O(T)$	23.47M	2.1	5.6	2.3	5.8

B. Main results

Table I lists the comparison result on AISHELL-1. From the table, we see that the proposed LAC achieves a CER of 4.73% on the development set and 5.02% on the test set, which is slightly worse than conformer [18]. The main advantage of LAC is that its parameter number and computational complexity are significantly lower than conformer, with a training speed of 1.23 times faster than the latter. Moreover, LAC significantly outperforms the other comparison methods.

Table II lists the comparison result on the LibriSpeech dataset. From the table, we see that the proposed LAC achieves a competitive result with conformer [18], and outperforms the other comparison methods. Although PIC has the smallest model size and a low computational complexity, its performance is not as good as LAC.

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on August 28,2022 at 14:48:14 UTC from IEEE Xplore. Restrictions apply.

TABLE III: Effect of the bottleneck dimension d_{bn} of LAC on performance.

Model	Complexity	# Param.	CER	
			Dev Set	Test Set
LAC ($d_{bn} = 50$)	$O(T)$	17.07M	4.85	5.16
LAC ($d_{bn} = 75$)	$O(T)$	19.95M	4.81	5.12
LAC ($d_{bn} = 100$)	$O(T)$	22.83M	4.73	5.02
LAC ($d_{bn} = 125$)	$O(T)$	25.71M	4.71	5.02

TABLE IV: Disentangling LAC. We replaced the core modules of the LAC encoder with the counterpart of conformer by (i) replacing low-rank feed forward (LFFN) module in LAC with feed forward (FFN) module in conformer, and (ii) replacing the MHLSA in LAC with the MHSA in conformer.

Model	Complexity	# Param.	CER	
			Dev set	Test set
LAC	$O(T)$	22.83M	4.73	5.02
−LFFN+FFN	$O(T)$	45.10M	4.54	4.90
−MHESA+MHSA	$O(T^2)$	22.83M	4.67	4.99
Conformer [18]	$O(T^2)$	45.15M	4.52	4.88

C. Effect of the bottleneck dimension d_{bn} of LAC.

We studied the effect of the bottleneck dimension d_{bn} of LAC on AISHELL-1. The experimental result in Table III shows that setting $d_{bn} = 50$ achieves the smallest model size with worst performance and setting $d_{bn} = 100$ yields a good trade-off between the model size and the performance.

D. Ablation studies

The proposed LAC differs from conformer in the multi-head linear self-attention and low-rank feed forward module. Table IV shows the effect of each change to conformer. Low-rank feed forward module reduces the number of parameters by approximate 50% with little performance degradation. Multi-head linear self-attention reduces the computational complexity of encoder from $O(T^2)$ to $O(T)$ without performance degradation. The proposed LAC achieves a good trade-off among model size, computational complexity, and performance.

IV. Conclusions

In this paper, we have proposed the acoustic model "linear attention based conformer" for speech recognition. Specifically, we first extended the linear attention originally proposed for object detection to a multi-head version and then applied it to speech recognition. It reduces the computational complexity of the self-attention layers in the conformer encoder to a linear order with respect to the input feature length. In addition, we also applied the low-rank matrix factorization to the feed forward module in conformer, which reduces the number of the parameters of conformer by approximate 50%. To improve the performance of LAC, we train and decode it jointly with the CTC objective. Experimental results on AISHELL-1 and LirbriSpeech show that the proposed LAC achieves competitive performance with conformer, and performs better than 7 other recent speech recognition models.

References

[1] 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.

[2] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[3] T. N. Sainath, Y. He, B. Li, A. Narayanan, R. Pang, A. Bruguier, S.-y. Chang, W. Li, R. Alvarez, Z. Chen *et al.*, "A streaming on-device end-to-end model surpassing server-side conventional model quality and latency," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6059–6063.

[4] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[6] Y. Fujita, A. S. Subramanian, M. Omachi, and S. Watanabe, "Attention-based asr with lightweight and dynamic convolutions," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7034–7038.

[7] K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, D. Belanger, L. Colwell *et al.*, "Masked language modeling for proteins via linearly scalable long-context transformers," *arXiv preprint arXiv:2006.03555*, 2020.

[8] K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser *et al.*, "Rethinking attention with performers," *arXiv preprint arXiv:2009.14794*, 2020.

[9] P. Wang and D. Wang, "Efficient end-to-end speech recognition using performers in conformers," *arXiv preprint arXiv:2011.04196*, 2020.

[10] M. Xu, S. Li, and X.-L. Zhang, "Transformer-based end-to-end speech recognition with local dense synthesizer attention," *arXiv preprint arXiv:2010.12155*, 2020.

[11] Z. Shen, M. Zhang, H. Zhao, S. Yi, and H. Li, "Efficient attention: Attention with linear complexities," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 3531–3539.

[12] I. Bello, "Lambdanetworks: Modeling long-range interactions without attention," *arXiv preprint arXiv:2102.08602*, 2021.

[13] S. Kriman, S. Beliaev, B. Ginsburg, J. Huang, O. Kuchaiev, V. Lavrukhin, R. Leary, J. Li, and Y. Zhang, "Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6124–6128.

[14] G. I. Winata, S. Cahyawijaya, Z. Lin, Z. Liu, and P. Fung, "Lightweight and efficient end-to-end speech recognition using low-rank transformer," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6144–6148.

[15] W. Han, Z. Zhang, Y. Zhang, J. Yu, C.-C. Chiu, J. Qin, A. Gulati, R. Pang, and Y. Wu, "Contextnet: Improving convolutional neural networks for automatic speech recognition with global context," *arXiv preprint arXiv:2005.03191*, 2020.

[16] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*. IEEE, 2017, pp. 1–5.

[17] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[18] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi *et al.*, "Recent developments on espnet toolkit boosted by conformer," *arXiv preprint arXiv:2010.13956*, 2020.

[19] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on August 28,2022 at 14:48:14 UTC from IEEE Xplore. Restrictions apply.

[20] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.
[21] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *Proc. Interspeech 2019*, pp. 2613–2617, 2019.
[22] Z. Tian, J. Yi, J. Tao, Y. Bai, S. Zhang, and Z. Wen, "Spike-triggered non-autoregressive transformer for end-to-end speech recognition," *arXiv preprint arXiv:2005.07903*, 2020.

Authorized licensed use limited to: NORTHWESTERN POLYTECHNICAL UNIVERSITY. Downloaded on August 28,2022 at 14:48:14 UTC from IEEE Xplore. Restrictions apply.