Journal of Southeast University (English Edition) Vol. 37, No. 2, pp. 164 - 170 June 2021 ISSN 1003—7985

Transformer-like model with linear attention for speech emotion recognition

Du Jing[1] Tang Manting[2] Zhao Li[1]

([1]School of Information Science and Engineering, Southeast University, Nanjing 210096, China)
([2]School of Computational Engineering, Jinling Institute of Technology, Nanjing 211169, China)

Abstract: Because of the excellent performance of Transformer in sequence learning tasks, such as natural language processing, an improved Transformer-like model is proposed that is suitable for speech emotion recognition tasks. To alleviate the prohibitive time consumption and memory footprint caused by softmax inside the multihead attention unit in Transformer, a new linear self-attention algorithm is proposed. The original exponential function is replaced by a Taylor series expansion formula. On the basis of the associative property of matrix products, the time and space complexity of softmax operation regarding the input's length is reduced from $O(N^2)$ to $O(N)$, where N is the sequence length. Experimental results on the emotional corpora of two languages show that the proposed linear attention algorithm can achieve similar performance to the original scaled dot product attention, while the training time and memory cost are reduced by half. Furthermore, the improved model obtains more robust performance on speech emotion recognition compared with the original Transformer.

Key words: transformer; attention mechanism; speech emotion recognition; fast softmax

DOI: 10.3969/j.issn.1003-7985.2021.02.005

Speech emotion recognition, one of the key technologies of intelligent human-computer interaction, has received increasing interest[1]. Recurrent neural networks, especially long short-term memory[2] and gated recurrent[3] neural networks, have been firmly established as the main approaches in sequence modeling problems, such as speech emotion recognition[4-5]. However, a recurrent neural network typically performs recursive computation along the positions of the input and output sequences, which results in the failure of parallel training[6]. Especially when handling ultralong sequences, the training efficiency of the recurrent neural network is extremely low because of computer memory constraints.

The Transformer model, completely based on the self-attention mechanism introduced by Google[7], solves the above problems effectively. By abandoning time-consuming operations, such as loops and convolutions, the time cost, as well as the memory footprint, is greatly reduced during training. In Transformer architecture, multihead attention (MHA) realizes the parallel training process, compared with the traditional self-attention mechanism, by allowing the model to pay attention to the information from multiple representation subspaces of different positions so that more information in the sequence will be retained. At present, MHA has been successfully applied in several fields. For example, India et al.[8] extended multihead self-attention in the field of speech recognition, which mainly solved the speech recognition problem of non-fixed-length input speech and achieved excellent performance. In the multimodal emotion recognition task for the IEMOCAP dataset[9], MHA is used to concentrate on the only relevant utterance of the target utterance[10], which improves the recognition accuracy by 2.42%. In Ref. [11], the dilated residual network combined with MHA was applied to feature learning in speech emotion recognition, which not only alleviated the loss of the feature's time structure but also captured the relative dependence of elements in progressive feature learning, achieving 67.4% recognition accuracy on IEMOCAP dataset.

However, the scaled dot product attention (SDPA) computing unit in MHA has quadratic complexity in time and space, which prohibits its application in the context of ultralong sequence input. Therefore, Taylor linear attention (TLA) is proposed to address this limitation, which has linear complexity in terms of the input sequence length and dramatically shortens the time cost and memory footprint. The proposed algorithm changes the way attention weights are calculated in SDPA by using a Taylor formula instead of the exponential operation in softmax and by making use of the associative property of matrix products to avoid the tremendous memory consumption of intermediate matrices. Transformer has been an exceeding success in the field of natural language processing, such as machine translation[12], since its introduction. In this paper, we extend Transformer to the area of speech emotion recognition, and the Transformer-like model (TLM) is thus proposed. The proposed TLA algo-

Received 2020-08-16, **Revised** 2021-05-10.

Biographies: Du Jing (1997—), female, graduate; Zhao Li (corresponding author), male, doctor, professor, zhaoli@seu.edu.cn.

Foundation items: The National Key Research and Development Program of China (No. 2020YFC2004002, 2020YFC2004003), the National Natural Science Foundation of China (No. 61871213, 61673108, 61571106).

Citation: Du Jing, Tang Manting, Zhao Li. Transformer-like model with linear attention for speech emotion recognition[J]. Journal of Southeast University (English Edition), 2021, 37(2): 164 - 170. DOI: 10.3969/j.issn.1003-7985.2021.02.005.

rithm is shown to have similar emotion recognition performance with SDPA, while the computational power requirement is tremendously reduced. Meanwhile, the TLM can enhance the position information representation of acoustic features and thereby obtain more robust emotion recognition performance.

1 Attention

1.1 Scaled dot product attention

The main implementation unit of MHA in Transformer is scaled dot product attention (SDPA), whose structure is shown in Fig. 1. The main idea is to enhance the representation of the current word by introducing context information. The query vector $\boldsymbol{Q}$ in Fig. 1 represents the content that the network is interested in. The key vector $\boldsymbol{K}$ is equivalent to the labels of all words in the current sample. The result of the dot product of $\boldsymbol{Q}$ and $\boldsymbol{K}$ reflects the influence degree of context words on the central word, and then softmax is used to normalize the correlation weights. Finally, the attention score is obtained by using the correlation matrix to weigh the value vector $\boldsymbol{V}$.

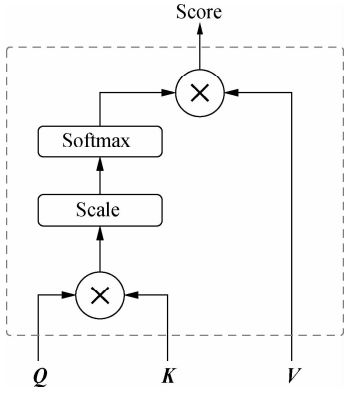

Fig. 1 Scaled dot product attention

SDPA is calculated by

$$\boldsymbol{A} = \frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d}} \tag{1}$$

$$\boldsymbol{S} = \mathrm{softmax}(\boldsymbol{A})\,\boldsymbol{V} \tag{2}$$

$$\mathrm{softmax}(\boldsymbol{A}_i) = \frac{\mathrm{e}^{\boldsymbol{A}_i}}{\sum_{j=1}^{N}\mathrm{e}^{\boldsymbol{A}_j}} \tag{3}$$

where $\boldsymbol{A}$ is the output after scaling; $\boldsymbol{S}$ is the output of the attention unit; $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are generated by the input feature vector with the shape of (N, d), so $\boldsymbol{Q}$, $\boldsymbol{K}$, $\boldsymbol{V} \in \mathbf{R}^{N\times d}$, where N represents the input sequence length, and d is the input sequence's dimension. Generally, $N > d$ or even $N \gg d$ is satisfied in ultralong sequence situations.

According to the definition of softmax, Eq. (2) can be mathematically expanded as

$$S_i = \frac{\sum_{j=1}^{N}\exp\left(\frac{\boldsymbol{q}_i^{\mathrm{T}}\boldsymbol{k}_j}{\sqrt{d}}\right)\boldsymbol{v}_j}{\sum_{j=1}^{N}\exp\left(\frac{\boldsymbol{q}_i^{\mathrm{T}}\boldsymbol{k}_j}{\sqrt{d}}\right)} \tag{4}$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are expressed as column vectors $\boldsymbol{q}_i$, $\boldsymbol{k}_i$, and $\boldsymbol{v}_i$, respectively. Therefore, the mathematical essence of SDPA is to use $\exp(\boldsymbol{q}_i^{\mathrm{T}}\boldsymbol{k}_j/\sqrt{d})$ to perform a weighted average of $\boldsymbol{v}_i$.

1.2 MHA

Multihead attention (MHA) is critically significant in parallel training for Transformer. By dividing the input vector into multiple feature subspaces and then applying the self-attention mechanism, the model may be trained in parallel while extracting the main information. Compared with the current mainstream single-head average attention weighting, MHA can improve the effective resolution to enhance the model's different characteristics of speech features in different subspaces, which avoids the inhibition by average pooling of such characteristics. MHA is calculated by

$$\left.\begin{aligned}\boldsymbol{Q}_i &= \boldsymbol{X}\boldsymbol{W}_{\boldsymbol{Q}_i}\\ \boldsymbol{K}_i &= \boldsymbol{X}\boldsymbol{W}_{\boldsymbol{K}_i}\\ \boldsymbol{V}_i &= \boldsymbol{X}\boldsymbol{W}_{\boldsymbol{V}_i}\\ \boldsymbol{H}_i &= \mathrm{SDPA}(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i)\end{aligned}\right\} \quad \forall\, i \in [1, n] \tag{5}$$

$$\boldsymbol{S} = \mathrm{Concat}(\boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_n)\,\boldsymbol{W} \tag{6}$$

where $\boldsymbol{X}$ is the input feature sequence; $\boldsymbol{Q}_i$, $\boldsymbol{K}_i$, and $\boldsymbol{V}_i$ represent query, key, and value, respectively; $\boldsymbol{H}_i$ is the attention score of each head; SDPA is the self-attention unit of each head; $\boldsymbol{W}$ is the linear transformation weight; $i = 1, 2, \ldots, n$, and n is the number of heads, and i is the index of each head.

First, the input feature sequence $\boldsymbol{X}$ is equally divided into n segments in the feature dimension, and each segment generates a group of ($\boldsymbol{Q}_i$, $\boldsymbol{K}_i$, and $\boldsymbol{V}_i$) after a linear transformation. Then, $\boldsymbol{H}_i$ is respectively calculated for each head. The n attention scores are spliced successively. Finally, the total attention score is generated from spliced vectors by performing the linear transformation.

1.3 Taylor linear attention

An obvious problem pertains to the use of MHA. When calculating SDPA, each head needs to use softmax to normalize the dot product of $\boldsymbol{Q}$ and $\boldsymbol{K}$ so that $\boldsymbol{V}$ can be weighted to obtain the score. As dividing subspaces by MHA will not affect the input sequence length, the length of $\boldsymbol{Q}$ and $\boldsymbol{K}$ is still N. With an increase in the input sequence length, the computational resource demand of each head during training will increase in quadratic order, which is unbearable and leads to a decrease in the quality of long-distance dependent modeling in sequence learning as well.

As a result, Taylor linear attention (TLA) is proposed to alleviate this problem. It can be concluded from Section 1.1 that the essence of the self-attention mechanism in MHA is to construct the weight matrix using the inner product form of $\boldsymbol{Q}$ and $\boldsymbol{K}$ and then to weigh $\boldsymbol{V}$, where the weight matrix is nonnegative. Accordingly, the Taylor

series expansion of $e^{q_i^T k_j}$ (because $e^{-\sqrt{d}}$ is a constant, its influence is temporarily ignored for the convenience of description and simplification of the derivation process) in Eq. (4) can be obtained as

$$e^{q_i^T k_j} = 1 + q_i^T k_j + \frac{(q_i^T k_j)^2}{2!} + \frac{(q_i^T k_j)^3}{3!} + \ldots \approx 1 + q_i^T k_j \tag{7}$$

If $q_i^T k_j \geqslant -1$, the left-hand side of Eq. (7) is guaranteed to be nonnegative. Moreover, the smaller $q_i^T k_j$ is, the closer the left and right sides of Eq. (7) are. Therefore, we can set $\mathrm{sim}(q_i, k_j) = 1 + q_i^T k_j$, and l_2 normalization is performed for q_i and k_j as follows:

$$\mathrm{sim}(q_i, k_j) = 1 + \left(\frac{q_i}{\| q_i \|}\right)^T \left(\frac{k_j}{\| k_j \|}\right) \tag{8}$$

Therefore, the inequality $\mathrm{sim}(q_i, k_j) \geqslant 0$ always holds. Based on the previous conclusion, TLA is equivalent to using $1 + q_{ei}^T k_{ej}$ to weigh v_j, where q_e and k_e represent the normalized column vectors of query and key, respectively. Eq. (4) can be equivalently written as

$$S_i = \frac{\sum_{j=1}^{N} (1 + q_{ei}^T k_{ej}) v_j}{\sum_{j=1}^{N} (1 + q_{ei}^T k_{ej})} \tag{9}$$

Eq. (9) can be rewritten as

$$S_i = \frac{\sum_{j=1}^{N} (v_j + q_{ei}^T k_{ej} v_j)}{\sum_{j=1}^{N} (1 + q_{ei}^T k_{ej})} \tag{10}$$

Eq. (10) may be further simplified to

$$S_i = \frac{\sum_{j=1}^{N} v_j + \sum_{j=1}^{N} q_{ei}^T k_{ej} v_j}{\sum_{j=1}^{N} 1 + \sum_{j=1}^{N} q_{ei}^T k_{ej}} \tag{11}$$

On the basis of the associative property of matrix multiplication, i. e., $(QK^T)V = Q(K^T V)$, Eq. (11) can be further simplified to

$$S_i = \frac{\sum_{j=1}^{N} v_j + q_{ei}^T \sum_{j=1}^{N} k_{ej} v_j^T}{N + q_{ei}^T \sum_{j=1}^{N} k_{ej}} \tag{12}$$

For Eq. (4), QK^T should be computed first when computing softmax, and the time complexity of SDPA is $O(N^2 d)$, which is approximately $O(N^2)$ in terms of $N \gg d$. For Eq. (12), according to the associative property of matrix multiplication, $K^T V$ can first be computed and then used to multiply Q, so the complexity is $O(Nd^2)$, which is approximately $O(N)$ when $N \gg d^2$. Moreover, $\sum_{j=1}^{N} k_{ej}$ obtained from Eq. (12) can be reused to decrease the memory footprint.

2 Model Structure

The TLM structure is shown in Fig. 2.

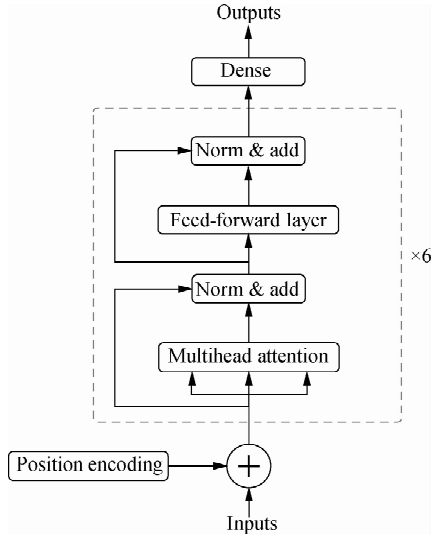

Fig. 2 Model structure

In the position encoding layer, because the expression of speech emotion is related to the position of emotional stimulation, and the model completely adopts the attention mechanism, it cannot learn the positional relationship among features, which means that input features need to be encoded additionally as follows:

$$P_{(p,2i)} = \sin\left(\frac{p}{10\,000^{2i/d}}\right) \tag{13}$$

$$P_{(p,2i+1)} = \cos\left(\frac{p}{10\,000^{2i/d}}\right) \tag{14}$$

where the shape of the original input vector is (N, d); $p \in [0, N)$ represents the p-th frame of inputs; $i \in [0, d/2 - 1]$, $2i$ and $2i + 1$ represent the even and odd dimensions of the current inputs, respectively. The position encoding vector retains the same shape as the original inputs, which are then concatenated with the audio feature vector in the feature dimension to generate the input vector of subsequent network layers with the shape of $(N, 2d)$.

Then, the TLA unit is adopted in the MHA layer. Considering that MHAs at different levels in BERT represent different functions, the bottom layer is usually more focused on grammar, while the top layer is more focused on semantics. Therefore, in this paper, multi-layer MHA is also adopted to learn different levels of speech emotion representation.

The feed-forward layer is composed of two linear transformations, and the calculation process is shown as

$$F(x) = \mathrm{GELU}(xW_1 + b_1) W_2 + b_2 \tag{15}$$

$$\mathrm{GELU}(x) = 0.5x\left\{1 + \tanh\left[\sqrt{\frac{2}{\pi}}(x + 0.044\,715x^3)\right]\right\} \tag{16}$$

where x is the input to the current layer; and W_i, b_i (i =

1, 2) denote the weights and biases to be trained in the i-th dense layer. The Gaussian error linear unit (GELU)[13] is adopted as the activation function to randomly regularize the input vector and match it with a random weight according to the size of the input.

As for the connection between layers, a residual connection is adopted, and the final output of the sublayer is normalized by

$$\boldsymbol{O} = \text{BatchNorm}[\boldsymbol{x} + \text{Sublayer}(\boldsymbol{x})] \tag{17}$$

where $\boldsymbol{x}$ is the input of the sublayer; Sublayer$(\cdot)$ denotes the implementation function of the sublayer. To facilitate the connection of residuals, the input and output of each sublayer remain the same dimension.

Finally, the predicted label is output from a fully connected layer through the softmax activation function.

To prevent overfitting, two regularization methods are used. One method is to use dropout before the final output of all sublayers and the dropout ratio $P_d = 0.1$. The other method is to adopt label smoothing, and all one-hot encoded label vectors are smoothed by

$$L' = (1 - \epsilon) L + \frac{\epsilon}{N} \tag{18}$$

where L is in the form of one-hot encoding; L' represents the label after smoothing; N is the number of one-hot encoding states; and $\epsilon = 0.1$.

The Adam optimizer is adopted in the training process. Moreover, the warmup learning rate[14] used in the experiments is calculated as follows:

$$r_s = (r_0 w)^{-0.5} \min(s^{-0.5}, sw^{-1.5}) \tag{19}$$

where r_0 is the initial learning rate; r_s is the learning rate at current training step s; and w denotes the warmup step. When the current step is less than w, the learning rate increases linearly; on the contrary, the learning rate decreases proportionally with the inverse square root of the number of steps. All parameter settings of the model are shown in Tab. 1.

Tab. 1 Model parameters

Parameters	Value	Parameters	Value
LFBE frames	300	Hidden layers' activation	GELU
LFBE features	64	Output activation	Softmax
Input sequence length	300	Batch size	32
Position encoding size	64	Epoch	500
Number of MHA layers	6	Dropout	0.1
Number of heads	8	Optimizer	Adam
Size per head	16	Initial learning rate	0.001
Feed-forward layers	6	Warmup step	1 000
Feed-forward size	512		

3 Experiments

3.1 Datasets

The experiments are performed on EmoDB[15] and URDU[16]. The information of each dataset is shown in Tab. 2. Four emotions, anger, happiness, neutral, and sadness, are selected in the experiment.

Tab. 2 Dataset information

Dataset	Language	Size	Emotions	Type
EmoDB[15]	German	535	Anger, boredom, disgust, fear, happiness, sadness, neutral	Acted
URDU[16]	URDU	400	Anger, happy, neutral, sad	Natural

3.2 Preprocessing

All data samples were resampled with 16 kHz, with a pre-emphasis coefficient of 0.97. Each file was divided into frames of 25-ms width with a stride of 10 ms. Any audio file longer than 300 frames was truncated to 300 frames, while files shorter than 300 frames were padded with zeros, where 300 was regarded as the sequence length. Log Mel-filter bank energies (LFBE) were then subsequently calculated for each frame with the number of filter banks set to 64. Each dataset was divided into a training set, validation set, and test set at a ratio of 8∶1∶1.

4 Results

4.1 Effect of the number of heads on performance

In most of the MHA models, such as BERT, the feature representation sizes (word embedding) are approximately 300-1 024 so that the number of heads is empirically set from 12 to 16[17]. Considering that our feature dimension is 128, we tried 2, 4, 8, 16, and 32 (factors of 128) to study the effect of the number of heads in MHA on the performance of speech emotion recognition, as shown in Fig. 3. In this experiment, the head number was the only variable, and other parameters remained the same as listed in Tab. 1.

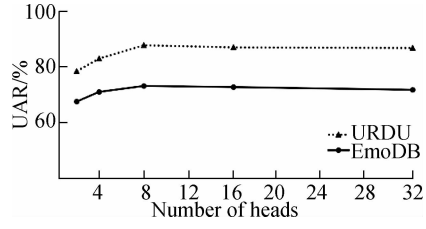

Fig. 3 Effect of the number of heads on the performance of emotion recognition

Fig. 3 shows that the number of heads does not have a significant effect on the performance of emotion recognition. Because of the redundancy of the attention mechanism, even if the attention head is calculated independently, there is a high probability that the emotional information paid attention to is consistent. Notably, UAR in-

crease with the number of heads on URDU and EmoDB, indicating that more attention can be paid to the local emotional information from those relative outlier attention heads with the increase in the number of heads so that the model is further optimized. However, when the number of heads reaches 8, UAR is almost unchanged or even slightly decreased, indicating that after the number of heads increases to a certain number, the expression ability of emotional information brought by multiple subspaces is enhanced to reach the upper bound. The increase in the number of heads may lead to an excessively scattered distribution of emotional information in the feature subspace, which results in a decline in the emotion recognition performance of the model. Therefore, appropriate head cardinality should be selected in the experiment not only to ensure the occurrence probability of outlier heads to learn more subtle emotion expression but also to prevent the distribution of emotional information from being too discrete to reduce the recognition performance. In this paper, the number of heads is set to eight in the subsequent experiments.

4.2 Effect of position embedding type on performance

In Transformer, a word vector is added to a positional encoding vector to embed positional information, which may not be applicable in speech emotion recognition. Therefore, we selected two embedding methods, named add and concatenation, to study the influence of embedding type on the recognition performance. Other parameters were kept consistent with those shown in Tab. 1.

Fig. 4 shows the UAR curve on the test set during the training. It can be intuitively seen that the recognition performance of the model with feature concatenation is better than that with feature addition. Moreover, the UAR of the model using the add method has greater volatility after convergence, reflecting that the Add embedding method causes the model's emotion recognition performance to be more unstable, which infers that directly adding or subtracting the position encoding vector to the input speech feature may result in invalidation of the position information embedding and even loss of the original emotional information. Consequently, using Concatenation on the TLM increases the robustness and improves the recognition performance to varying degrees.

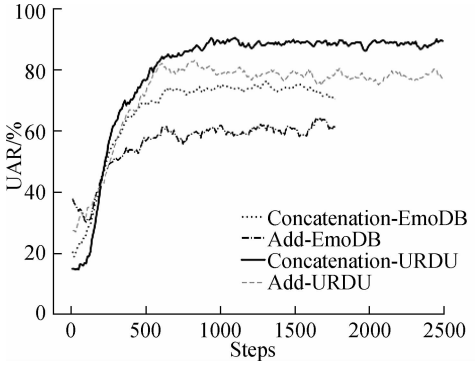

Fig. 4 Effect of embedding type for position encoding vectors on the test set

4.3 Emotion recognition performance

To verify the speech emotion recognition performance of the proposed method, we chose the TLM with the SDPA unit as the baseline[17], where eight heads and the concatenation method were adopted, and other parameters were consistent with those in Tab. 1. Additionally, we also chose some classical models for further comparison, such as the support vector machine (SVM)[18] and ResNet[19], which represent the traditional machine learning method and prevailing CNN framework, respectively. Each model adopted the same input as described in Section 3. The UAR accuracy results on each dataset are shown in Tab. 3.

Tab. 3 Recognition accuracy of different models on different emotion categories %

Datasets	Model	Recognition accuracy				UAR
		Anger	Neutral	Happy	Sad	
EmoDB	SVM[18]	64.3	100.0	14.3	85.7	66.1
	ResNet-50[19]	100.0	50.0	21.4	92.3	65.9
	Baseline	85.7	71.4	71.4	85.7	78.6
	Proposed	71.4	71.4	71.4	85.7	74.9
URDU	SVM[18]	80.0	80.0	80.0	70.0	77.5
	ResNet-50[19]	90.0	60.0	80.0	40.0	60.0
	Baseline	90.0	80.0	80.0	90.0	85.0
	Proposed	80.0	80.0	70.0	90.0	80.0

The Transformer-like model outperforms SVM and ResNet-50, signifying that the TLM is more suitable in the field of speech. Compared with the baseline, the emotion recognition performance using TLA is not significantly different from that of SDPA on the whole, which indicates the effectiveness of the attention unit algorithm proposed in this paper.

4.4 Model complexity

The change in the UAR with step number and time after iterating 3 000 steps on the baseline and proposed model are shown in Figs. 5 and 6, respectively, under the parameter settings shown in Tab. 1. As can be seen, the proposed TLA algorithm and the SDPA algorithm perform similarly at emotion recognition, but the proposed TLA algorithm is far lower than the baseline SDPA algorithm in training time cost, indicating that TLA has lower time complexity.

To further compare the complexity of the proposed TLA, four groups of Transformer-like models were trained on EmoDB. The lengths of the input sequence (LFBE frames) were chosen as 256, 512, 768, and 1 024. The processor used in the experiment was Inter © Core(TM) I7-8700 CPU @ 3.20 GHz, the GPU was NVIDIA Geforce RTX 2080Ti, and the memory size was

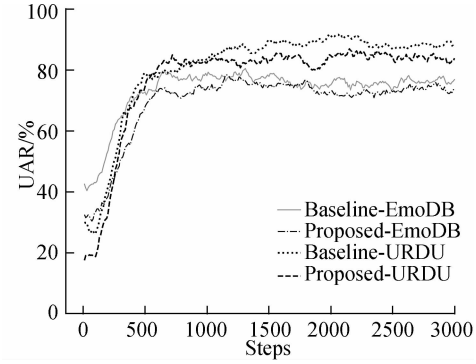

Fig. 5 UAR comparison between the baseline and proposed models within 3 000 steps

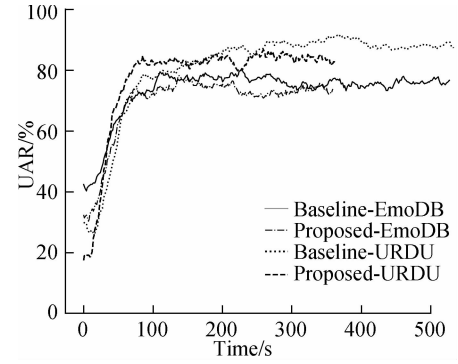

Fig. 6 UAR comparison between the time use of the baseline and proposed models within 3 000 steps

16.0 GB. To avoid the overflow of memory errors, the batch size was selected as eight for training. The other parameters were kept consistent with Tab. 1, and each model was iterated for 1 500 steps.

The training time of the model as the length of input sequence increases is shown in Fig. 7 when iterating the same steps, where the time of the baseline approximately conforms to the square distribution, while that of the proposed TLM roughly meets the linear distribution. The proposed TLM obviously has linear time complexity regarding the input sequence length. Regarding the memory usage, as shown in Fig. 8, the memory footprint of TLM is much smaller than that of the baseline. In addition, when the input feature length is 768, the memory usage

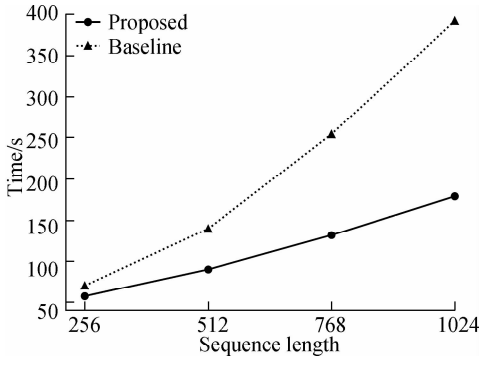

Fig. 7 Comparison between the time use of the baseline and proposed methods with different sequence lengths

has reached the upper limit of available memory so that although the number of input feature frames increases in subsequent experiments theoretically, the actual memory usage of the model remains unchanged. Similar to the time consumption distribution, the memory use of the baseline approximately conforms to a square distribution, while the memory occupation of the TLM roughly satisfies a linear distribution, indicating that the proposed model has a linear space complexity in terms of the sequence length.

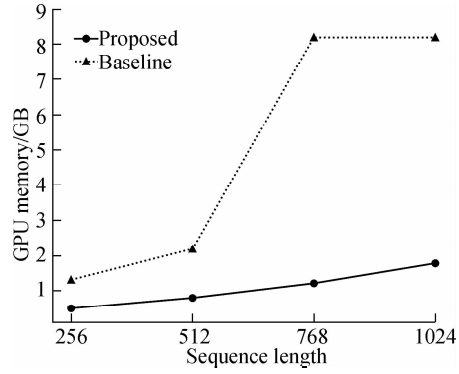

Fig. 8 Comparison between the GPU memory use of the baseline and proposed methods with different sequence lengths

5 Conclusions

1) The best performance of MHA is found with eight heads, indicating a certain limit on the recognition accuracy brought by the number of heads.

2) For the attention computing unit, the proposed TLA algorithm not only has similar emotion recognition performance to SDPA but also greatly reduces the time cost and memory footprint during training by making use of the Taylor formula and the associative property of matrix products, leading to linear complexity in time and space.

3) For speech emotion recognition tasks, a novel TLM is proposed, achieving a final UAR of 74.9% and 80.0% on EmoDB and URDU, respectively. The experimental results demonstrate that the TLM has certain advantages in handling ultralong speech sequences and has bright, practical application prospects due to the greatly reduced demand for computing power.

References

[1] Akçay M B, Oguz K. Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers[J]. *Speech Communication*, 2020, **116**: 56 − 76. DOI: 10.1016/j.specom.2019.12.001.

[2] Hochreiter S, Schmidhuber J. Long short-term memory [J]. *Neural Computation*, 1997, **9**(8): 1735 − 1780. DOI: 10.1162/neco.1997.9.8.1735.

[3] Chung J, Gulcehre C, Cho K, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling [EB/OL]. (2014) [2020-08-01]. https://arxiv.org/abs/

1412.3555.
[4] Mirsamadi S, Barsoum E, Zhang C. Automatic speech emotion recognition using recurrent neural networks with local attention[C]//2017 *IEEE International Conference on Acoustics, Speech and Signal Processing*. New Orleans, LA, USA, 2017: 2227 - 2231. DOI: 10.1109/ICASSP.2017.7952552.
[5] Greff K, Srivastava R K, Koutník J, et al. LSTM: A search space odyssey[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2017, **28**(10): 2222 - 2232. DOI: 10.1109/TNNLS.2016.2582924.
[6] Thakker U, Dasika G, Beu J, et al. Measuring scheduling efficiency of RNNs for NLP applications[EB/OL]. (2019)[2020-08-01]. https://arxiv.org/abs/1904.03302.
[7] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//*Advances in Neural Information Processing Systems*. Long Beach, CA, USA, 2017: 5998 - 6008.
[8] India M, Safari P, Hernando J. Self multi-head attention for speaker recognition[C]//*Interspeech* 2019. Graz, Austrilia, 2019: 4305 - 4309. DOI: 10.21437/interspeech.2019 - 2616.
[9] Busso C, Bulut M, Lee C C, et al. IEMOCAP: interactive emotional dyadic motion capture database[J]. *Language Resources and Evaluation*, 2008, **42**(4): 335 - 359. DOI: 10.1007/s10579-008-9076-6.
[10] Lian Z, Tao J H, Liu B, et al. Conversational emotion analysis via attention mechanisms[C]//*Interspeech* 2019. Graz, Austrilia, 2019: 1936 - 1940. DOI: 10.21437/interspeech.2019 - 1577.
[11] Li R N, Wu Z Y, Jia J, et al. Dilated residual network with multi-head self-attention for speech emotion recognition[C]//2019 *IEEE International Conference on Acoustics, Speech and Signal Processing*. Brighton, UK, 2019: 6675 - 6679. DOI: 10.1109/ICASSP.2019.8682154.
[12] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training ofdeep bidirectional transformers for language understanding[EB/OL]. (2019)[2020-08-01]. https://arxiv.org/abs/1810.04805
[13] Hendrycks D, Gimpel K. Gaussian error linear units (GELUs)[EB/OL]. (2016)[2020-08-01]. https://arxiv.org/abs/1606.08415.
[14] He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition[C]//2016 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*). Las Vegas, NV, USA, 2016: 770 - 778. DOI: 10.1109/CVPR.2016.90.
[15] Burkhardt F, Paeschke A, Rolfes M, et al. A database of German emotional speech[C]//*Interspeech* 2005. Lisbon, Portugal, 2005: 1517 - 1520.
[16] Latif S, Qayyum A, Usman M, et al. Cross lingual speech emotion recognition: Urdu vs. western languages[C]//2018 *International Conference on Frontiers of Information Technology* (*FIT*). Islamabad, Pakistan, 2018: 88 - 93. DOI: 10.1109/FIT.2018.00023.
[17] Nediyanchath A, Paramasivam P, Yenigalla P. Multi-head attention for speech emotion recognition with auxiliary learning of gender recognition[C]//2020 *IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*). Barcelona, Spain, 2020: 7179 - 7183. DOI: 10.1109/ICASSP40776.2020.9054073.
[18] Chavan V M, Gohokar V V. Speech emotion recognition by using SVM-classifier[J]. *International Journal of Engineering & Advanced Technology*, 2012(5): 11 - 15.
[19] Xi Y X, Li P C, Song Y, et al. Speaker to emotion: Domain adaptation for speech emotion recognition with residual adapters[C]//2019 *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*. Lanzhou, China, 2019: 513 - 518. DOI: 10.1109/APSIPAASC47483.2019.9023339.

基于线性注意力和类 Transformer 模型的语音情感识别

杜 静[1] 唐曼婷[2] 赵 力[1]

([1] 东南大学信息科学与工程学院, 南京 210096)
([2] 金陵科技学院计算机工程学院, 南京 211169)

摘要:鉴于 Transformer 模型在自然语言处理等序列任务中的优异性能, 提出了一种适用于语音情感识别任务的改进的类 Transformer 模型. 为了减小 Transformer 模型中多头注意力单元内部由 softmax 运算引起的巨大时间消耗与内存开销, 提出了一种新的线性自注意力计算方法, 通过使用泰勒级数展开公式代替原来的指数函数, 并根据矩阵乘积的关联性将 softmax 运算相对于输入序列长度的时间复杂度和空间复杂度从 $O(N^2)$ 降至 $O(N)$, 其中 N 为序列长度. 在 2 个不同语言的情感语料库上进行实验. 结果表明:所提出的线性注意力算法可获得与原始缩放点积注意力相近的性能, 而模型训练过程中的时间和内存开销大幅降低;与原始的 Transformer 模型相比, 改进后的模型具有更鲁棒的语音情感识别性能.

关键词:Transformer;注意力机制;语音情感识别;快速 softmax
中图分类号:TN912.3;TP18