Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/47/20241291

An analysis of attention mechanisms and its variance in transformer

Yuzhong Chen[1], Hongren Pu[2], and Yang Qu[3,4,5]

[1]School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu, 215028, China
[2]College of Mathematics, Taiyuan University of Technology, Jinzhong, Shanxi, 030024, China
[3]College of Physics, Sichuan University, Chengdu, Sichuan, 610064, China

[4]quyang@stu.scu.edu.cn
[5]corresponding author

Abstract. Transformer is a machine learning model based on attention mechanism, which is widely used. When the Transformer model was first proposed, it gradually developed many variants and was promoted and applied in many fields, becoming an important research part in the areas of deep learning. However, the critical attention mechanism of Transformers has issues such as square complexity that affect computational speed and data processing efficiency. In order to meet the needs of data processing and related computing, there have been endless efforts to improve the attention mechanism in Transformers in different work areas. This article mainly provides a brief overview of the recent research progress on the attention mechanism in Transformers. Select representative studies from several directions of attention improvement work to introduce, in order to explore the latest research trends in its improvement work and lay a foundation for pointing out potential research directions for future research work and further improving the performance of Transformers.

Keywords: Transformer, Attention Mechanism, Deep Learning.

1. Introduction

Classical neural networks are mainly used to deal with the data classification issue, and their effectiveness is very significant when analyzing and calculating discrete data with low correlation. But when considering processing sequential data, traditional feed-forward networks (FNN) can't satisfy the requirements. That's because when dealing with sequential data, such as a sentence or a passage, classical neural networks need to calculate every word and consider the relationship between them. Recurrent neural networks (RNN) were usually used to resolve this problem. But there are some limits in this model:

(1) RNN only recurrently deal with sequential data, this model can't perform parallel calculations in procession and training.

(2) RNN aren't used to process long sequential data, this model has the "vanishing gradient" problem.

(3) Due to the property of recurrent neural networks, this model only can process the relationship between one data with the former data. Therefore, this model can only be used for limited data sets.

© 2024 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/47/20241291

Long short-term memory (LSMT) [1] can solve “vanishing gradient” problem and handle long data. However, it still cannot perform parallel calculations, and its running speed is slower compared to classical RNN. Afterward, some models were improved on LSMT to improve its running speed, but still have some limitations in data processing.

For the need of processing sequential data, and further improving the efficiency, Transformer [2] was built for this. This model is based on attention mechanism to improve running and training speed. Transformer can realize parallel calculations and track the relationship of different data. Therefore, it is possible to process long and complex sequential data. Thus, once proposed, this model has become one of the important models in the field of natural language processing (NLP), then has gradually been extended to computer vision (CV), authentic intelligence model enhancement and other fields.

Transformer is a very flexible neural network model that can be improved according to different needs to meet different work needs [3]. Therefore, once proposed, this model has gradually become one of the research hotspots of neural network models. Because attention is the key mechanism of Transformer, in this paper, combining many previous improvements survey to Transformer [3-5] and recent research works, this work provides a brief overview of the latest improvements to Transformer's attention mechanism. Starting from aspects such as computation and performance improvement, this work has introduced some prominent advances in attention improvement that have not been detailed in previous literature.

The organization of the paper is as follows. In section 2, we briefly introduce Transformer model, especially attention mechanism in Transformer, self-attention and muti-head attention, and the problems. The latest progress in improving the attention in Transformer is introduced in section 3 and 4. In section 5, some discussions and conclusions are given.

2. Preliminary Knowledge of Attention in Transformer

Transformer, which is usually called the vanilla Transformer, consists of two model blocks: encoder and decoder [2]. There are several small modules in the encoder, and the number of modules in the decoder equal to the encoder. Each small module concludes the multi-head self-attention layer and FFN layer. After the input passes through above two layers, it then passes through the linear and softmax layer. The self-attention mechanism is the critical structure of vanilla Transformers and a critical factor that sets Transformers ahead of classical neural network models.

2.1. Self-Attention module

The relationship of certain data in the dataset can be dealt with due to the self-attention mechanism, and encode and decode related data. It is shown in the following figure.

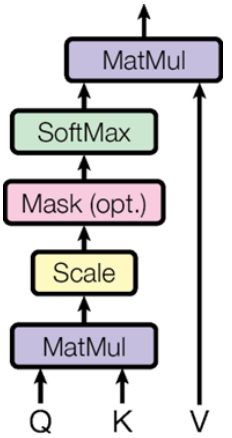

Figure 1. Scaled Dot-Product Attention [2].

DOI: 10.54254/2755-2721/47/20241291

Self-attention layer contains three matrix representations: query **Q**, key **K**, and value **V**, and through the calculations between **Q**, **K**, and **V** to realize the data process. Input $\mathbf{X} \in \mathbb{R}^{T\times d}$ first performs dot product with $\mathbf{W}_Q \in \mathbb{R}^{d\times d}$, $\mathbf{W}_K \in \mathbb{R}^{d\times d}$, and $\mathbf{W}_V \in \mathbb{R}^{d\times d}$ respectively to obtain **Q**, **K**, and $\mathbf{V} \in \mathbb{R}^{T\times d}$, d indicates the hidden dimension, and T indicates the input sequence length. And then calculate **Q** and **K** first to obtain similarity, and weight the result on the **V** yields the output. The process of self-attention can be shown as following formula,

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = (\text{softmax}(\frac{\mathbf{QK}^T}{\sqrt{d_k}}))\mathbf{V} = \mathbf{AV} \quad (1)$$

, where **A** is attention matrix, d_k is the dimensions of **K** (here $d_k = d$), the factor $\sqrt{d_k}$ is used to solve "vanishing gradient" problem caused by softmax.

2.2. Multi-Head Attention

In order to further optimize the feature representation, multi-head attention mechanism separately projects original **Q**, **K** and **V** onto a lower dimensional space through a linear layer for h times. Its structure is shown in Figure 2. H different outputs generated by performing Eq. (1) on each set are then combined and additionally projected to the ultimate value. The process can be represented as follows:

$$\text{Att}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Concat}(\mathbf{H}_1,\ldots,\mathbf{H}_h)\mathbf{W}^O, \text{where } H_i = \text{Attention}(\mathbf{QW}_i^Q, \mathbf{KW}_i^K, \mathbf{VW}_i^V) \quad (2)$$

This mechanism has similar advancement as convolutional neural network (CNN) as it provides multiple output channels to improve the feature extraction. Therefore, multi-head self-attention enables attention with various sub-expressions and strengthens the model capability of focusing on different positions.

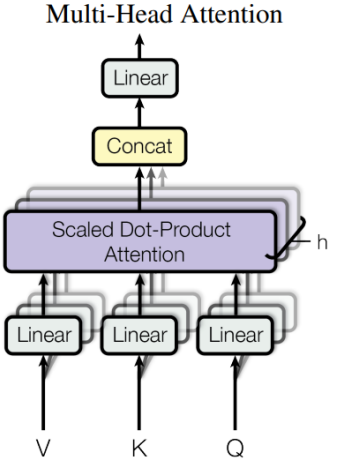

Figure 2. Multi-Head Attention [2].

2.3. Limitation Analysis

Although attention mechanism has many excellent traits and solves some problems that occur in the RNN, it isn't a perfect mechanism, and it has some defects and limits [3], which are: (1) This mechanism has a quadratic complexity related to the input sequence $\mathcal{O}(T^2d)$. Therefore, when Transformer processes large datasets such as long text or long sequences, it will cost a lot of time. (2) As attention heads increases, the model may begin to learn some redundant feature representations because it can obtain similar information from multiple heads. This may reduce the generalization ability of the model. (3) Transformer has no structural bias towards input data, and the order of output data largely depends on the training data, which can easily lead to overfitting.

DOI: 10.54254/2755-2721/47/20241291

These issues have also prompted researchers to continuously improve attention mechanism in Transformer.

3. Attention Optimization for Improving Computational Efficiency

In the self-attention mechanism, other tokens need to be weighted with each token. Although this process has many benefits, such as connecting sequence elements and contextual information, the complexity problem can lead to significant consumption of computing resources. Due to some flaws in the calculation process of self-attention layer, such as its quadratic complexity, one of the goals of researchers is to solve computational defects such as complexity problems or the ability to process tokens by optimizing its calculation steps, thereby improving computational efficiency. There are many efforts to reduce the complexity, and two most commonly used improvement methods are linearized complexity and sparse attention.

3.1. Linear Complexity

The complexity of the quadratic power is one of the main problems that attention mechanism faces in the calculation process. Simplifying complexity to linear is one of the main directions for improving computational speed and efficiency. There are many tasks that simplify complexity, and the general method is to linearize attention. The linearized attention method utilizes mapping to adjust the calculation order of **Q**, **K**, and **V**, thereby reducing the quadratic complexity to linear. There are many works that utilize this method to reduce complexity to a linear [6-8]. In addition, there are many recent works, and here a few prominent works are introduced.

External attention uses external memory to calculate attention matrix [9]. To be specific, in self-attention, to gain the output, key and query first calculate to obtain the attention map, and then weight attention map with value. But external attention introduces external learned key memory in linear layers to calculate the attention matrix and corresponding outputs. For input future map $\mathbf{F} \in \mathbb{R}^{\mathrm{T}\times d}$, the main process can be expressed as

$$\mathbf{A} = \mathrm{Norm}(\mathbf{F}\mathbf{M}_{\mathrm{k}}^{\mathrm{T}}) \tag{3}$$

$$\mathbf{F}_{\mathrm{out}} = \mathbf{A}\mathbf{M}_{\mathrm{v}} \tag{4}$$

, where $\mathbf{M}_k$ and $\mathbf{M}_{\mathrm{v}} \in \mathbb{R}^{S\times d}$ are memory units that represent key and value respectively, and $\mathbf{A} \in \mathbb{R}^{T\times T}$ is attention matrix, Norm is normalization process. Its structure is demonstrated in Figure 3.

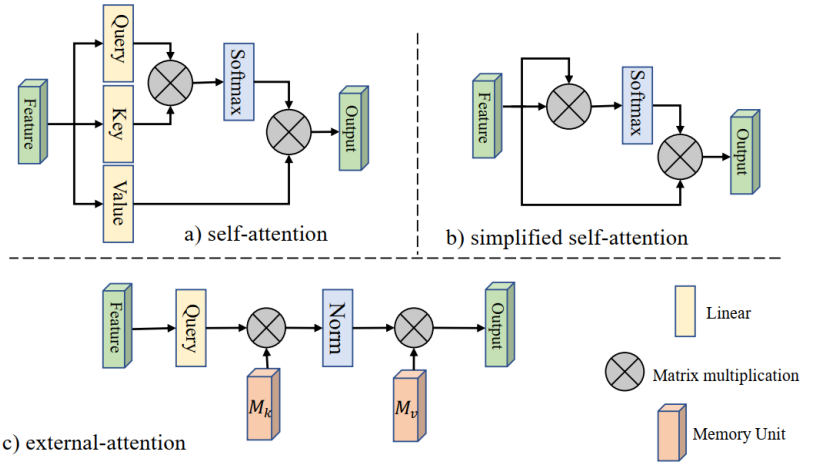

Figure 3. Compare of self-attention and external attention [9].

The complexity of this process is $\mathcal{O}(\mathrm{dST})$. Due to the simplification of complexity to linear by external attention, its computational speed and efficiency should be correspondingly improved

DOI: 10.54254/2755-2721/47/20241291

compared to the previous model. Comparative experiments have shown external attention requires only half of the parameters required by standard self-attention (SA) [2], and its speed is about twice that of the best variant of SA.

QuadTree attention reduces complexity from quadratic to linear by introducing quadtree as shown in Figure 4 [10]. In image processing tasks, most images can be divided into many unrelated regions. Therefore, inspired by this idea, the concept of quadtree can be borrowed for processing. Quadtrees are typically used to recursively subdivide two-dimensional space into four quadrants or regions. QuadTree attention constructed a token pyramid, which is divided into different levels. At each level, the top K patches with the highest score are selected through calculation. In the next level, only the areas related to these patches are calculated and evaluated, thus achieving a coarse to fine calculation of attention. In QuadTree attention, complexity is related to the number of tokens and exhibits a linear relationship.

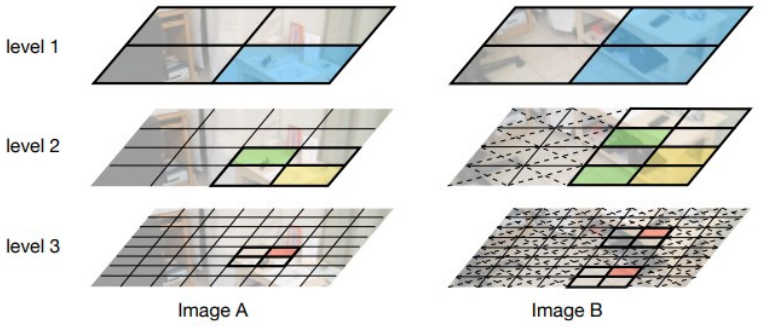

Figure 4. Illustration of QuadTree attention, here K=2 and level=3 [10].

The efficiency of QuadTree attention has been significantly improved. The QuadTree attention model reduces FLOP by about 50% in stereo matching, the classification tasks on image classification and ImageNet show that the model has fewer FLOPs, in COCO object detection dataset, its FLOPs have been reduced by about 35% in experiments[10].

Fast linear attention with a single head (FLASH) model uses the Gated Attention Unit (GAU) mechanism to replace self-attention and FFN in the Transformer, and simplify complexity to linear inspried by linearized attention idea [11]. Its structure is shown in Figure 5. GAU is formed by combining Gated Linear Unit (GLU) with attention, which firstly calculates the intermediate states $\mathbf{U} \in \mathbb{R}^{T\times e}$ and $\mathbf{V} \in \mathbb{R}^{T\times e}$ for input, e denotes the expanded intermediate size, then the corresponding output is

$$\mathbf{O} = (\mathbf{U} \odot \mathbf{AV})\mathbf{W_o} \tag{5}$$

, where $\mathbf{W_o} \in \mathbb{R}^{e\times d}$, $\odot$ is element-wise product, $\mathbf{A} \in \mathbb{R}^{T\times T}$ is attention matrix. If $\mathbf{A}$ is an identity matrix, then GAU degenerates into GLU. If $\mathbf{A}$ is a all-one matrix, then GAU degenerates into self-attention.

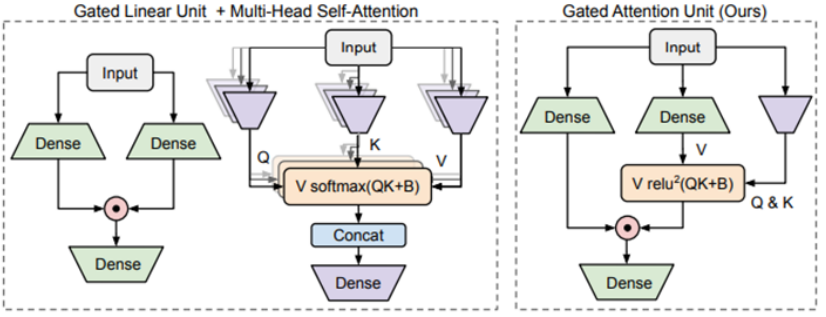

Figure 5. GLU and Multi-Head self-attention versus GAU [11].

DOI: 10.54254/2755-2721/47/20241291

Then, the complexity of GAU is reduced through Mixed Chunk Attention. Divide the sequence size C into non overlapping G segments, and take intermediate states as the g-th block, through four simple affine transformations to obtain $\mathbf{Q}_g^{quad}$, $\mathbf{K}_g^{quad}$, $\mathbf{Q}_g^{lin}$, and $\mathbf{K}_g^{lin}$. Then calculate the local attention per chunk and global attention across chunks respectively to obtain $\widehat{\mathbf{V}}_g^{quad}$and $\widehat{\mathbf{V}}_g^{lin}$. Finally, the two results are integrated to obtain the calculation formula for linear GAU

$$\mathbf{O}_g = (\mathbf{U}_g \odot (\widehat{\mathbf{V}}_g^{quad} + \widehat{\mathbf{V}}_g^{lin}))\mathbf{W}_o \tag{6}$$

This is the FLASH model, which simplifies the complexity from GAU to linear. Subsequently, the author conducted a series of experiments, particularly comparing the running time about models. The comparing result of the running time of the FLASH model with other Transformer models shows that the FLASH model performs better and is faster than the Transformer, especially when processing long text.

Attention free Transformer (AFT) is a model that replaces the self-attention layer [12]. From the structure, AFT also has **Q**, **K**, and **V** like dot product self-attention, but the difference is the combining form. The value and key calculate with learned position biases to compose the weight, then the calculation result does the element-wise product with query. Given the input **X**, first obtain **Q**, **K**, and **V** through linear transformation, and then performs the following operations

$$\mathbf{Y} = f(\mathbf{X}),\ Y_t = \rho_q(\mathbf{Q}_t) \odot \frac{\sum_{t'=1}^{T} \exp(\mathbf{K}_{t'} + \mathbf{w}_{t,t'}) \odot \mathbf{V}_{t'}}{\sum_{t'=1}^{T} \exp(\mathbf{K}_{t'} + \mathbf{w}_{t,t'})} \tag{7}$$

, where ρ_q is a nonlinear function that defaults to sigmoid, $\mathbf{w} \in \mathbb{R}^{T \times T}$ is a learned paired position biases. For each target position t, the weighted average of AFT execution values is multiplied by the query element by element. This model has linear complexity and can handle large data inputs. The experiment results of image autoregressive modeling, character level language modeling, and image classification show that AFT has good performance and higher efficiency, and its running speed has been significantly improved compared to standard Transformers and Image Transformers.

3.2. Sparse Attention

The emergence of sparse attention mechanism has effectively reduced the computation complexity of attention in models, especially in dealing with long sequence situations. Specifically, this mechanism can be seen as a sparse layer compare to the standard self-attention mechanism. The standard attention pattern and sparse attention pattern can be represented as follows:

$$\mathbf{M} = \mathbf{Q}\mathbf{K}^{\mathbf{T}} \tag{8}$$

$$\mathbf{M} = \begin{cases} q_i k_i^T & if\ token\ q_i\ attends\ to\ token\ k_i \\ -\infty & if\ token\ q_i\ not\ attends\ to\ token\ k_i \end{cases} \tag{9}$$

where **M** represents the attention matrix. Sparse structures can be observed in standard attention patterns, making it possible to introduce sparse attention to reduce the weights that each token needs to calculate. The key to sparse attention mechanism lies in exploring appropriate methods of sparsity while maintaining Transformer's performance, thus enabling it to accept inputs of more tokens. The latest works in this area are primarily based on combining ideas from different algorithms or other improvement directions for sparsity. Several recent prominent works are as follows.

Unlimiformer is a sparse attention model based on the kNN algorithm as displayed in Figure 6 [13]. For a given long input sequence, Unlimiformer constructs a kNN index based on hidden states of all input tokens. Then, in each standard cross-attention head at every decoder layer, Unlimiformer uses kNN dot-product as attention weights, while only focusing on the top-k input tokens.

DOI: 10.54254/2755-2721/47/20241291

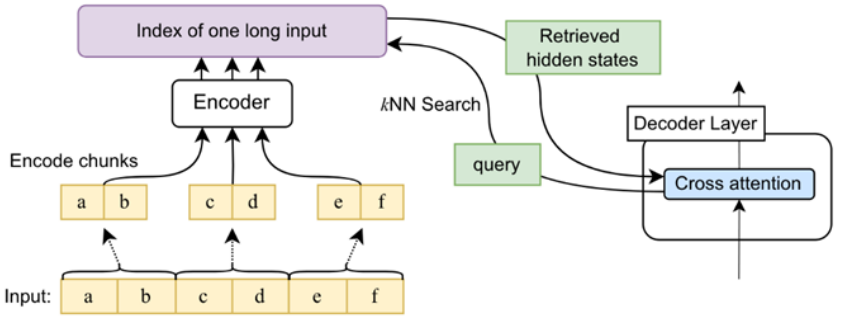

Figure 6. A model with Unlimiformer encoder has a maximum input length of 2 tokens [13].

Unlimiformer focuses attention on the top-k tokens, allowing it to extract information from the entire input sequence. This approach significantly reduces complexity since Unlimiformer ignores tokens with lower weights for each computed token. Moreover, since the softmax function is typically dominated by the maximum value, Unlimiformer theoretically preserves the majority of performance. The summarization experiments on 3 datasets with average input sequence lengths ranging from 10k to 143k tokens showed significant improvements for different training approaches based on Unlimiformer.

LongNet utilizes a form of sparse attention called "Dilated Attention" [14]. This model applies different dilated attention processing to individual attention heads, enabling LongNet to capture attention variations between different heads. Therefore the model can simultaneously capture both long-range and short-range information. As illustrated in Figure 7, both the segment length and dilated rate of dilated attention in different heads of LongNet can be adjusted according to the sequence length.

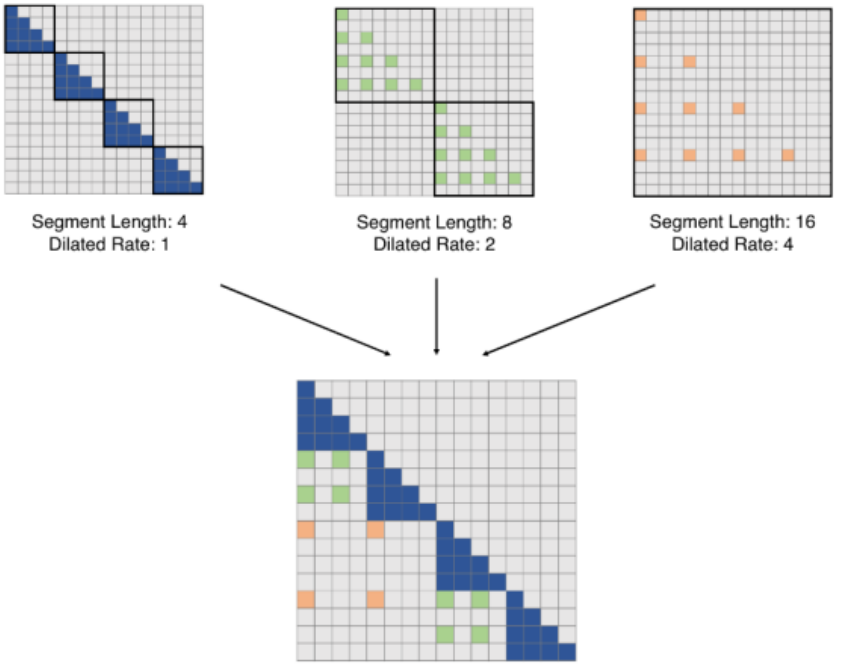

Figure 7. The combination of different sparse patterns in different heads [14].

From another perspective, this sparse attention pattern is similar to dilated convolutions, introducing certain gaps or skips in the attention computation process, thereby reducing computational complexity. Moreover, because there is still sparse global attention in some heads, this theoretically maintains the ability to capture long-range attention. Experiment in Figure 8 demonstrates variations in runtime

DOI: 10.54254/2755-2721/47/20241291

between Dilated Attention and Regular Attention as sequence length increases, highlighting the superiority of Dilated Attention.

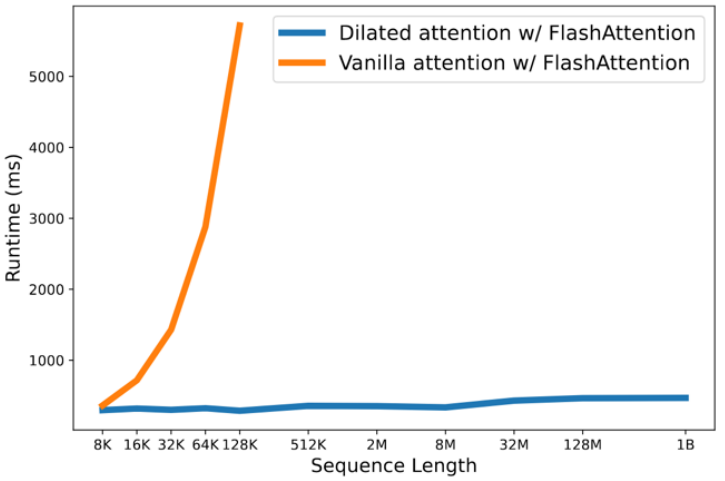

Figure 8. Comparison of runtime between Dilated Attention and regular attention [14].

The RMT (Recursive Memory Transformer) model integrates block sparse attention and a recurrent memory mechanism [15]. RMT segments lengthy sequences, constraining attention to operate within each segment in order to reduce complexity. Prior to attention computation, the model presets the memory state obtained from the previous segment. This sparse pattern resembles RNN. The recurrent memory mechanism also facilitates information transfer between segments, theoretically preserving the model's long-text handling capabilities. The trained model that applied the RMT optimization method was tested on various tasks and demonstrated improved generalization when processing longer texts.

4. Attention Optimization for Improving Performance

For attention mechanism, there are various efforts to improve the performance that mainly reflect on improving the structure of the attention mechanism to improve performance or adapt to more complex tasks. Improving the multi-head mechanism and utilizing the low-rank attribute of attention are two main ways to realize this purpose.

4.1. Multi-Head Mechanism Improvement

For specific work, improving the multi-head mechanism is often a major direction for improving model performance. The specific methods can be categorized by different stages of the multi-head mechanism where the optimization is performed, including the prior processing on **Q**, **K** or **V**. Besides, limitations and constraints can be placed on the projection or attention span. Furthermore, the combination of the h outputs can be complicated with certain method to make full use of the diversity that multi-head mechanism offers. Here several recent progressive works on improving the multi-head mechanism to improve performance are introduced.

Shatter model improved on the multi-head model, in which the multi-head has been simplified into single head to achieve performance improvement [16]. Concretely speaking, Shatter introduces a relatively soft partition of sequence with a partition mask $\mathbf{N}$, an $n \times l \times l$ tensor and defines $\mathbf{K} = \mathbf{X}$ to raise the operation efficiency. Due to the switch from multi-head to single head, the softmax layer needs adjustment accordingly. It has been determined that sigmoid presents better performance than softmax in terms of valid loss, which suggests the idea of introducing L2-Normalized sigmoid to replace the previous operation. Therefore, the attention matrix can be expressed by

DOI: 10.54254/2755-2721/47/20241291

$$\mathbf{A} = \text{L2_Normalize}\left(\sigma\left(\frac{\mathbf{Q}\mathbf{X}^{\mathrm{T}}}{\sqrt{\mathrm{d}}}\right)\right)\odot\mathbf{N} \tag{10}$$

Experiment pretrains both BERT and Shatter under various conditions. In each set of them Shatter always has significantly less parameters and consumes less time, which demonstrates that the simplification and reconstruction of multi-head mechanism contribute to the better performance of Shatter compared to other models.

Another variant called ConvFormer is designed to estimate monocular 3D human poses as shown in Figure 9 [17]. This model has dramatically reduced the parameter amount for nearly 60% due to the novel dynamic multi-head convolutional self-attention. Compared to regular multi-head structure, convolutional scaled self-attention applies special and same generation on **Q**, **K** and **V** in $\mathbb{R}^{\mathrm{T}\times d}$. H heads are generating separately and simultaneously, and h attention scores will be performed with concatenation to output the result of dynamic multi-head convolutional self-attention (DMHCSA) layer. Thus, initial Eq. (2) will be written as

$$DMHCSA(\mathbf{Q},\mathbf{K},\mathbf{V}) = Concat(\mathbf{H}_1,\ldots,\mathbf{H}_h) \tag{11}$$

Massive image samples experiment showed that ConvFormer accomplished the reduction of parameters and lower error rate in the meantime compared to the former model named State of the Art (SOTA), even on some particularly complex and swift actions.

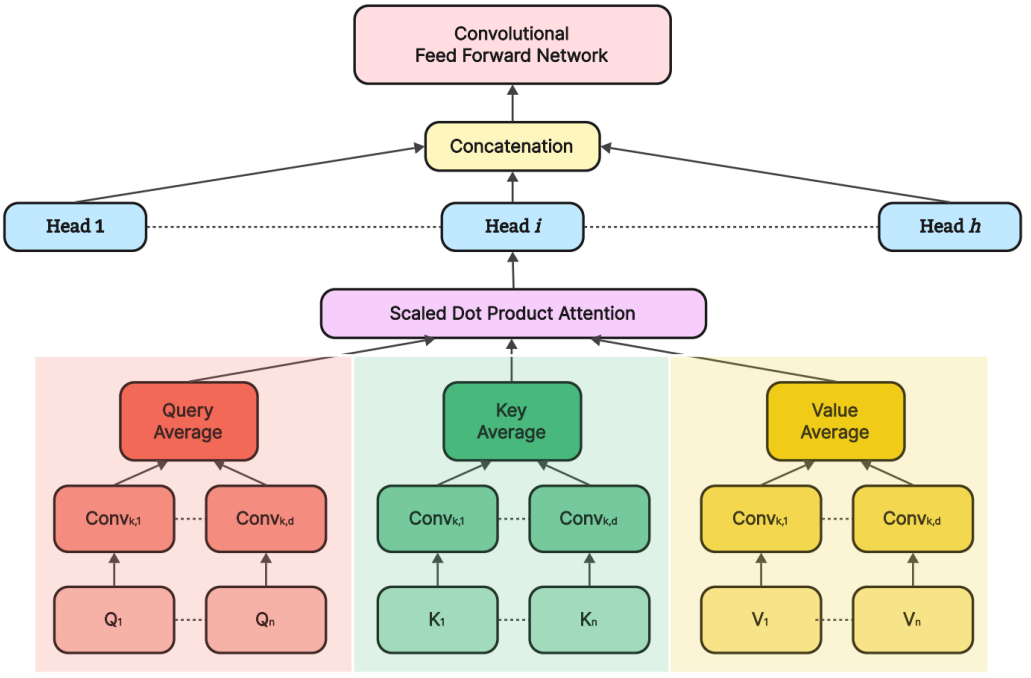

Figure 9. Graphical representation of the ConvFormer process [17].

The third model is quite different from the previous ones, as it focuses on pruning heads of little contribution to the results [18]. The model aims at neural machine translation, while during the process small numbers of heads play a crucial role. The core idea is to eliminate less important heads to achieve performance improvement. The model applies layer-wise relevance propagation (LRP) to compare the correlation between each head's output and the best prediction of the model, and head with high relevance is considered as valuable head. As for the process of pruning, the model defines a scalar gate g_i, a parameter determined by head rather than input. Then multiply g_i with Eq. (2) as the novel multi-head self-attention operation. Besides, g_i is further performed with regularization function to strongly drive the model to put more attention on significant heads rather than unconsidered ones. By applying

DOI: 10.54254/2755-2721/47/20241291

this model, the quantity of heads has notably decreased, while performance remains basically stable with tiny loss of approximately 0.25 BLEU score according to relevant experiment. Therefore, the capability and efficiency of processing text get improved in the meantime.

4.2. Low-Rank

Another direction that can be used to improve attention performance is to consider the low-rank attribute, which means that the self-attention has the low-rank feature. The feature can be considered from two aspects. Firstly, when input length T is larger than the rank of $\mathbf{A}$, low-rank features can be considered to reduce dimension and thus reduce the number of parameters [19]. Alternatively, consider using a low-rank matrix instead of an attention matrix to optimize self-attention and improve efficiency. This method is inspired by kernel approximation, and the key is to use low rank approximation to simplify the attention matrix. Several recent works on improving the attention mechanism using low-rank method will be introduced.

The low-rank Transformer (LRT) [20] uses the low-rank approximation method to simplify the attention matrix, replacing the high-rank matrix with the low-rank matrix, thereby reducing the matrix dimension. This model introduces linear encoder-decoder units to replace the feed-forward unit, and approximately decomposes the $\mathbf{W} \in \mathbb{R}^{a\times b}$ matrix in the feed-forward unit into two parts $\mathbf{E} \in \mathbb{R}^{a\times c}$ and $\mathbf{D} \in \mathbb{R}^{c\times b}$

$$\mathbf{W} \approx \mathbf{E} \times \mathbf{D} \tag{12}$$

Therefore, through the approximate decomposition of this matrix, the LED unit can be combined with the multi-head attention mechanism

$$\text{head}_i = \text{Attention}\left(\mathbf{Q}\mathbf{E}_i^Q\mathbf{D}_i^Q, \mathbf{K}\mathbf{E}_i^K\mathbf{D}_i^K, \mathbf{V}\mathbf{E}_i^V\mathbf{D}_i^V\right) \tag{13}$$

$$f(\mathbf{K},\mathbf{Q},\mathbf{V}) = \text{Concat}(\text{h}_1,\ldots,\text{h}_\text{H})\mathbf{E}^\text{O}\mathbf{D}^\text{O} + \mathbf{Q} \tag{14}$$

, where f is the attention function. Low-rank multi-head attention can be depicted by Figure 10.

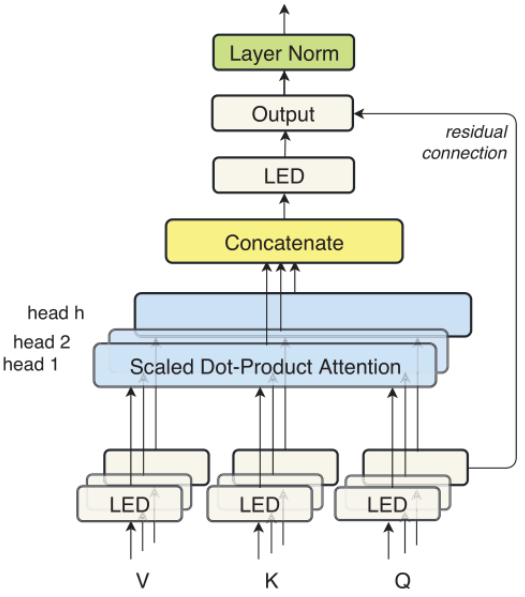

Figure 10. Illustration of Low-rank multi-head attention [20].

Compared with Transformers, the performance of this model has significantly improved. The experiment expresses that this model has fewer errors and stronger generalization ability. Compared to the baseline Transformer model, this model reduces parameters by over 50%.

DOI: 10.54254/2755-2721/47/20241291

Another Transformer variant named LightSANs [21] aims to eliminate the influence by over-parameterization and enhance performance on next-item recommendation. The core idea is to introduce low-rank decomposed self-attention layer into the model. Concretely speaking, the model defines a mapping relation from $\mathbf{H} \in \mathbb{R}^{\mathrm{T}\times\mathrm{d}}$ to $\widetilde{\mathbf{H}} \in \mathbb{R}^{\mathrm{k}\times\mathrm{d}}$ to yield a low-rank matrix, where k stands for a relatively small number of latent interests compared to T. The function is represented as follows:

$$\widetilde{\mathbf{H}} = f(\mathbf{H}) = (softmax(\mathbf{H} \cdot \mathbf{\Theta}^{\mathrm{T}}))^{\mathrm{T}} \cdot \mathbf{H} \tag{15}$$

, where $\mathbf{\Theta}$ is a learnable parameter in $\mathbb{R}^{\mathrm{k}\times\mathrm{d}}$. Then apply the above equation on $\mathbf{K}$ and $\mathbf{V}$ and replace them with low-rank matrices $\widetilde{\mathbf{K}}$ and $\widetilde{\mathbf{V}}$ respectively in Eq. (2). Reduction in matrix size leads to higher efficiency, while higher accuracy in attention weights of latent interest results in higher effectiveness and strengthens the capability of interest representation and prediction. Related experiments have shown that the model performance will get greatly decreased if the process of low-rank mapping is skipped, which proves the significance of the generation in LightSANs from the opposite perspective.

Thus far most research primarily applied low-rank methods independently to the Transformer architecture. FLuRKA [22] achieves efficient approximate self-attention computation by merging low-rank and kernel methods. This model incorporates kernel methods to substitute the softmax function, resulting in accelerating and ultimately improving performance. The fusion principle is shown in Figure 11.

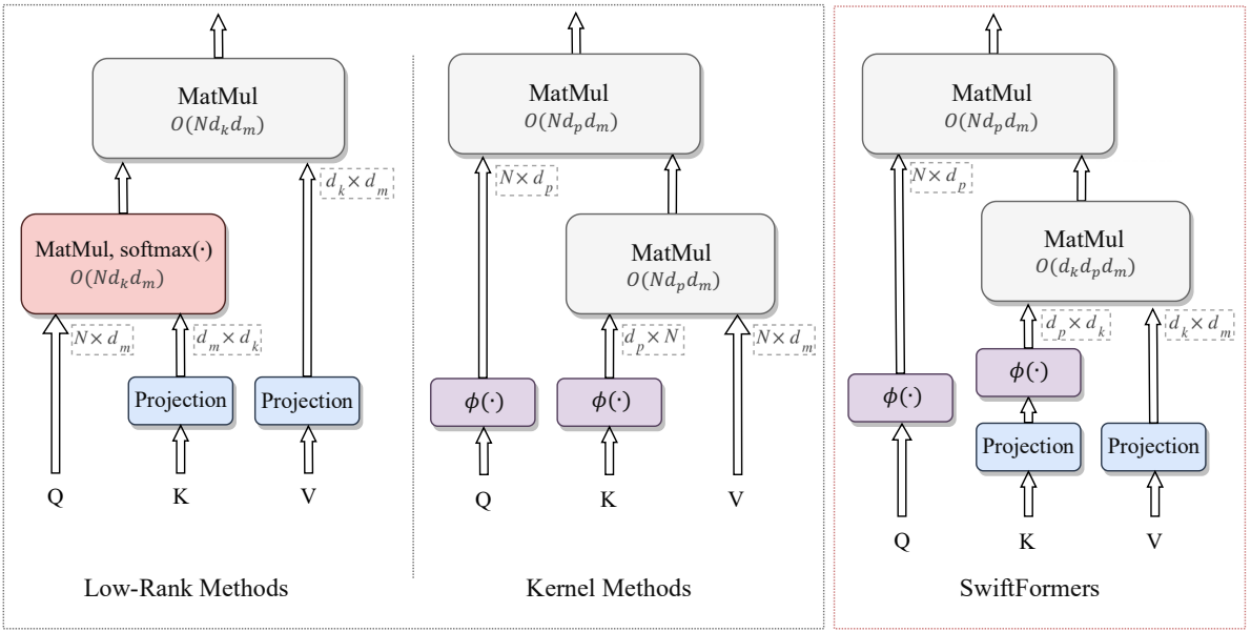

Figure 11. The fusion of low-rank and kernel methods results in Swiftformer [22].

This model leverages existing efficient low-rank methods (linformer) and various kernel methods, leading to speed improvements ranging from 30% to 200% compared to using the corresponding low-rank or kernel methods individually.

5. Conclusion

This paper mainly summarizes the recent optimization work of attention mechanism in Transformer and introduces the latest landmark progress. The article mainly focuses on two aspects. Firstly, starting with the calculation process of attention, this work introduces some of the latest works on dealing with the problem of excessive computation consumption when processing long sequences caused by quadratic complexity in the attention mechanism through methods such as linearization complexity and sparsity, thereby greatly improving computational and running speed. On the other hand, this work introduces

DOI: 10.54254/2755-2721/47/20241291

relevant work to improve the performance of attention, including restructing the attention structure and adapting to more complex tasks by improving multi-head mechanisms and low-rank methods, to improve the performance of problem-solving and model operation. Improvement from a single perspective has limitations, while combining multiple aspects of improvement is a necessary trend for adapting to computing and problem-solving. The optimization of multifaceted and comprehensive attention mechanisms has gradually become one of the main directions for future research.

Authors contribution

All authors contributed equally to this research, and their names are listed in alphabetical order.

References

[1] Graves, A., & Graves, A. (2012). Long short-term memory. Supervised sequence labelling with recurrent neural networks, 37-45.
[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. Advances in neural information processing systems, 30, 1-8.
[3] Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. AI Open, 111-132.
[4] Guo, M. H., Xu, T. X., Liu, J. J., Liu, Z. N., Jiang, P. T., et al. (2022). Attention mechanisms in computer vision: A survey. Computational visual media, 8(3), 331-368.
[5] Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in vision: A survey. ACM computing surveys, 54(10s), 1-41.
[6] Katharopoulos, A., Vyas, A., Pappas, N., & Fleuret, F. (2020). Transformers are rnns: Fast autoregressive transformers with linear attention. In International conference on machine learning, 5156-5165.
[7] Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., et al. (2020). Rethinking attention with performers. arXiv preprint arXiv:2009.14794.
[8] Schlag, I., Irie, K., & Schmidhuber, J. (2021). Linear transformers are secretly fast weight programmers. In International Conference on Machine Learning, 9355-9366.
[9] Guo, M. H., Liu, Z. N., Mu, T. J., & Hu, S. M. (2022). Beyond self-attention: External attention using two linear layers for visual tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(5), 5436-5447.
[10] Tang, S., Zhang, J., Zhu, S., & Tan, P. (2022). Quadtree attention for vision transformers. arXiv preprint arXiv:2201.02767.
[11] Hua, W., Dai, Z., Liu, H., & Le, Q. (2022). Transformer quality in linear time. In International Conference on Machine Learning, 9099-9117.
[12] Zhai, S., Talbott, W., Srivastava, N., Huang, C., Goh, H., Zhang, R., & Susskind, J. (2021). An attention free transformer. arXiv preprint arXiv:2105.14103.
[13] Bertsch, A., Alon, U., Neubig, G., & Gormley, M. R. (2023). Unlimiformer: Long-range transformers with unlimited length input. arXiv preprint arXiv:2305.01625.
[14] Ding, J., Ma, S., Dong, L., Zhang, X., Huang, S., Wang, W., & Wei, F. (2023). Longnet: Scaling transformers to 1,000,000,000 tokens. arXiv preprint arXiv:2307.02486.
[15] Bulatov, A., Kuratov, Y., & Burtsev, M. S. (2023). Scaling Transformer to 1M tokens and beyond with RMT. arXiv preprint arXiv:2304.11062.
[16] Tian, R., Maynez, J., & Parikh, A. P. (2021). Shatter: An efficient transformer encoder with single-headed self-attention and relative sequence partitioning. arXiv preprint arXiv:2108.13032.
[17] Diaz-Arias, A., & Shin, D. (2023). ConvFormer: Parameter Reduction in Transformer Models for 3D Human Pose Estimation by Leveraging Dynamic Multi-Headed Convolutional Attention. arXiv preprint arXiv:2304.02147.
[18] Voita, E., Talbot, D., Moiseev, F., Sennrich, R., & Titov, I. (2019). Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. arXiv preprint arXiv:1905.09418.

DOI: 10.54254/2755-2721/47/20241291

[19] Guo, Q., Qiu, X., Xue, X., & Zhang, Z. (2019). Low-rank and locality constrained self-attention for sequence modeling. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 27(12), 2213-2222.

[20] Winata, G. I., Cahyawijaya, S., Lin, Z., Liu, Z., & Fung, P. (2020). Lightweight and efficient end-to-end speech recognition using low-rank transformer. In IEEE International Conference on Acoustics, Speech and Signal Processing, 6144-6148.

[21] Fan, X., Liu, Z., Lian, J., Zhao, W. X., Xie, X., & Wen, J. R. (2021). Lighter and better: low-rank decomposed self-attention networks for next-item recommendation. In Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval, 1733-1737.

[22] Gupta, A., Yuan, Y., Zhou, Y., & Mendis, C. (2023). FLuRKA: Fast fused Low-Rank & Kernel Attention. arXiv preprint arXiv:2306.15799.