DEEP LEARNING

ACCELERATED ATTENTION MECHANISMS

Master Degree in Data Science and Scientific Computing

Gabriele Ruggeri
2021/2022

Contents

1 Abstract

Since the Attention mechanism was introduced in 2015 (Bahdanau, Cho, & Bengio,) this technique has revolutioned the Deep Learning community leading to new architectures like the Transformer (Vaswani et al.,), BERT (Devlin, Chang, Lee, & Toutanova,), ViT (Dosovitskiy et al.,) which immidiately achieved state-of-art performances on tasks such as NLT, MLM, NSP, Image Classification, VQA and so forth.
The main contribution of the Attention Model was the ability to capture arbitrarily long dependencies within the input data without any loss of information, against previous state-of-art models in both NLP and CV, that mainly relied on RNNs and CNNs that can only keep information with fixed lenght memory. Attention allowed researchers to develop models that understand the semantic meaning and details within images and text, being able to project machine learning towards new horizons; also pushed by the great interpretability of the Attention that again is a new trait with respect to the aforementioned models. Although the great benefits the Attention mechanism still could be a bottleneck in many applications because of the amount of operations and memory needed to compute and store its weights.
In this project I will review some of the most recent and interesting revisitations of the Attention mechanism with a focus on the ones that enhance the time complexity, from the original which is quadratic in the input size. Starting from a brief review of the original ideas, both for Attention (Bahdanau et al.,) and Self-Attention (Vaswani et al.,), I will later analyze seven different approaches where the time complexity is reduced to either a linear or a polynomial function of order $o < 2$, and where this approaches either modify the initial idea by introducing an inductive bias reasonable for the type of data under consideration (Wang et al.,) (Yang et al.,) (Chu et al.,) (Liu et al.,) or revisit the Attention mechanism in its core design (Zhang, Xiong, & Su,) (Child, Gray, Radford, & Sutskever,) (Choromanski et al.,).

2 Attention and Self-Attention

2.1 Attention

(Bahdanau et al.,) introduced for the first time the Attention mechanism for the task of machine translation. Given $\mathbf{x} = \{x_1, x_2, \ldots, x_{T_x}\}$ and $\mathbf{y} = \{y_1, y_2, \ldots, y_{T_y}\}$ respectively input and output sentence, the translation was performed by a *RNN Encoder-Decoder* where the encoder reads the input sequence and returns a sequence of hidden states $\mathbf{h} = \{h_1, h_2, \ldots, h_{T_x}\}$ containing information about the context at each position. The decoder, when predicting the i-th term y_i looks at the best matches within the input sequence to be aligned with the output token. Formally

$$p(y_i|y_1, y_2, \ldots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i)$$

where g is a bidirectional-RNN, s_i is its hidden state at time i and c_i is the context vector for the i-th position, which again is a weighted average of the encoder's hidden states:

$$c_i = \sum_{j=1}^{T_x} a_{i,j} h_j$$

where

$$a_{i,j} = softmax(e_i)_j, \quad e_{i,j} = a(s_{i-1}, h_j)$$

provide the probabilities $(a_{i,j})$ that the i-th output token depends or is translated from the j-th input token. The alignment model a was designed as a *FNN*, which is learnt jointly with all the other components. We can notice that the time complexity of the attention layer is $\mathbf{O}(T_x T_y)$, basically quadratic for sentences that are expected to have a similar length.

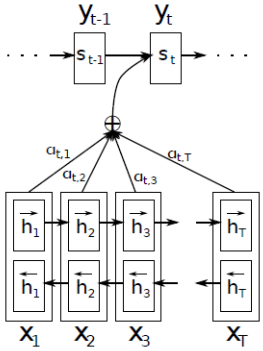

Figure 1: An illustration of how the attention mechanism is embedded in the architecture. From (Bahdanau et al.,)

2.2 Self-Attention

Employed in many modern architectures Self-Attention captures the intra-dependencies within input data, in order to decorate hidden representations of data with arbitrarily long dependencies.
Let $X \in \mathbf{R}^{N \times D}$ be an embedding for the input data, let $Q(X), K(X), V(X) \in \mathbf{R}^{\mathbf{N} \times \mathbf{d}_{\text{model}}}$ be respectively the *query*, *key* and *value* matrices; then:

$$\text{Attention}(Q, V, K) = softmax\Big(\frac{QK^T}{\sqrt{d_{\text{model}}}}\Big)V = AV,$$

which outputs $AV \in \mathbf{R}^{N \times d_{\text{model}}}$. Notice that $A \in \mathbf{R}^{N \times N}$, so that every row of A is the attention vector over all the tokens in the input data. Once again the complexity of the Self-Attention is $\text{O}(\text{d}_{\text{model}} N^2)$.

2.2.1 Multi-Head Self-Attention

MHSA is a variation of the SA that splits the computation of the Attention over h heads, where the matrices Q, K, V are either splitted over the heads, reducing their dimensionality, or we learn h different $\{Q^l, K^l, V^l\}_{l \in \{1,2,\dots,h\}}$. Then the h heads outputs $\{(AV)^l\}_{l \in \{1,2,\dots,h\}}$ are concatenated and eventually reprojected in $\mathbf{R}^{N \times d_{\text{model}}}$. The time complexity is not changed as a function of the input size.

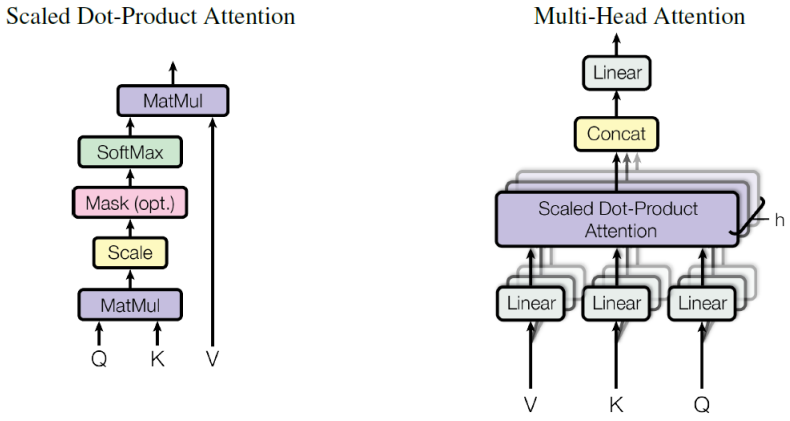

Figure 2: An illustration of both self-attention and multi headed self-attention. From (Vaswani et al.,)

3 Accelerated-Attentions designed for Vision Tasks

After the success of Transformers (Vaswani et al.,) for NLP, many attempts to bring the power of Attention into computer vision tasks were attempted (Dosovitskiy et al.,). Some of the reasons behind this will are that CNNs, which have been and still are the state-of-art for this tasks, need very deep designs to overcome the problem of the limited receiptive field, which is crucial to many dense vision tasks, and are less prone to properly capture long range dependencies. On the contrary the Attention mechanism not only has no "spacial limit" but it is also able to learn such relationships in few layers; at the cost of quadratic complexity. Time complexity is the main bottleneck in applying Self-Attention to image data since, differently from text, the size of an image is usually 10-100 times bigger the average sentence, leading to very poor performance; moreover images have a very different semantic with respect to text data since the way in which the information distributes in a 2-d space is different from the 1-d case of text, and on top of that usually information is not concentrated in a single pixel but mostly into chunks or strips of the whole image so that it may be useless to compute the Attention matrix at a pixel granularity. This brief introduction will hopefully clarify some of the design choices in the following revisitation of the Attention mechanism.

3.1 Swin

The *Shifted Window Transformer*, (Liu et al.,), was proposed as a new convolution-free backbone for neural networks in computer vision. Based on a hierarchical representation of data with a resolution that gets lowered as long as the network gets deeper, Swin replaces the usual encoder block of the Transformer, based on multi-head self-attention with a *Swin Transformer block* that uses windows based attention (W-MSA/SW-MSA). The idea behind W-MSA is to divide the input image into non overlapping sub-windows made of patches of size $\{2^i \times 2^i\}_{i\in\{2,3,4,5\}}$, and compute MSA locally to the windows. For an image of size $H \times W \times 3$ with windows containing $M \times M$ patches we have a time complexity of

$$\mathbf{O}(M^2hwC) = \mathbf{O}(hw) = \mathbf{O}(HW),$$

where $h = H/M$, $w = W/M$ and C is the inner dimensionality of the patches. The W-MSA achieves linear complexity because of the fixed window size ($M = 7$ in (Liu et al.,)). The drawback of W-MSA is that it lacks cross-window connections which reflects in limited modelling power; but to mitigate this issue the authors proposed a "shifted window partitioning approach which alternates between two partitioning configurations in consecutive layers". Considering a pair of subsequent layers where the first uses W-MSA on a 2×2 windows partition scheme with 4×4 pathes per window, starting from the top-left; then the second layer will use a shifted window multi-head self attention where all the windows are translated by the vector $(\lfloor \frac{M}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$. Although the SW-MSA let the model learn cross window dependencies it introduces some asymmetries with respect to W-MSA, since after the shift not only we have more windows than

in the case of W-MSA but they could also be rectangular, leading to windows with a different number of patches. The last introduction is the *relative position bias* $B \in \mathbf{R}^{M^2 \times M^2}$ introduced for each head; which substitutes the position embeddings in the input data.

$$Attention(Q, K, V) = softmax\Big(\frac{QK^T}{\sqrt{d}} + B\Big)V$$

In the end W-MSA and SW-MSA proved to be both effective and efficient, leading to new state-of-art scores in the tasks they were originally designed for, on the other side they represent a task specific, first try in revisiting the attention mechanism for improved performances.

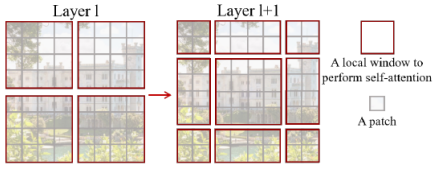

Figure 3: An illustration of how W-MSA and SW-MSA see the input image. From (Liu et al.,)

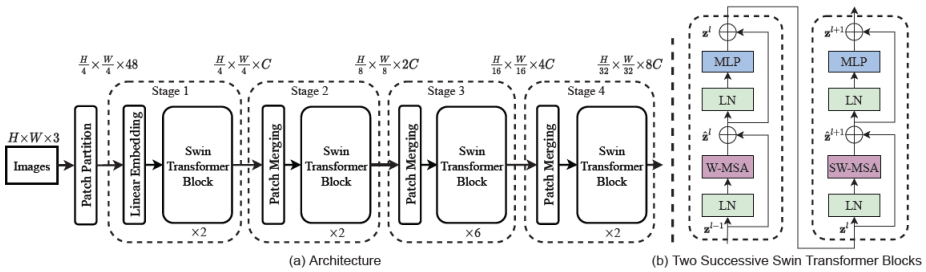

Figure 4: An illustration of (a) the Swin Transformer architecture and (b) two subsequent Swin layer which use respectively W-MSA and SW-MSA. From (Liu et al.,)

3.2 PVT

The *Pyramid Vision Transformer*, (Wang et al.,), belongs to the same framework of Swin, that is the convolution free hierarchical representation transformer based backbone for computer vision models. The architecture of PVT is divided in stages, each corresponding to a feature map with a given scale.
At the i-th stage the input image of size $H_{i-1} \times W_{i-1} \times C_{i-1}$ is divided in $\frac{HW}{2^{2(i+1)}}$ patches which are linearly projected in $\mathbf{R}^{\frac{HW}{2^{2(i+1)}} \times C_i}$, passed to the L_i transformer layers of this stage and then the output feature map F_i is reshaped into $\mathbf{R}^{\frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}} \times C_i}$. Each of the L_i transformer layers is composed of an attention layer and a *FFN* like in the original architecture (Vaswani et al.,), but MHSA is replaced with a spacial-reduction attention (SRA).
Let Q, V, K be the usual query, key and value matrices in $\mathbf{R}^{(H_i W_i) \times C_i}$ (with C_i channel dimension at stage i), the idea is to reduce the scale of K and V using the operator

$$SR(x) = Norm(Reshape(x, R_i)W^S)$$

where $x \in \mathbf{R}^{(H_i W_i) \times C_i}$ is the input sequence and R_i is the reduction ratio of the attention layer at stage i, so that:

$$Reshape(\cdot, R_i): \quad \mathbf{R}^{(H_i W_i) \times C_i} \to \mathbf{R}^{\frac{H_i W_i}{R_i^2} \times (R_i^2 C_i)}.$$

In the end $W^S \in \mathbf{R}^{(R_i^2 C_i) \times C_i}$ is a learnable linear projection that reduces the dimension back to C_i. The SRA at stage i can be formulated as:

$$SRA(Q, K, V) = \text{Concat}(head_0, head_1, head_2, \ldots, head_{N_i})W^O \quad \text{with}$$

$$head_j = Attention(QW_j^Q, SR(K)W_j^K, SR(V)W_j^V)$$

where N_i is the number of heads, d_{head} is the dimension of one head, clearly equal to $\frac{C_i}{N_i}$, $W_j^Q, W_j^K, W_j^V \in \mathbf{R}^{C_i \times d_{\text{head}}}$ and $W^O \in \mathbf{R}^{C_i \times C_i}$. The time complexity of SRA at the i-th stage is $\mathbf{O}(\frac{H_i^2 W_i^2}{R_i^2})$, still quadratic in the input size, but the complexity constant of SRA is smaller than the one for MHSA by a factor of R_i^2 that in practice, as shown in (Wang et al.,) and (Chu et al.,), leads to a great speed-up.

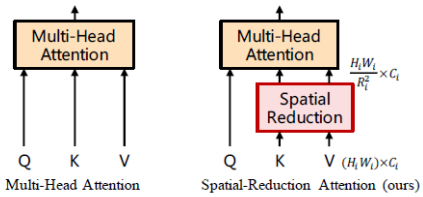

Figure 5: A comparison between MHSA and SRA. From (Wang et al.,)

3.3 Twin

The *Twin Transformer*, (Chu et al.,), was designed to deal with the time and space complexity of Self-Attention and proposed some revisitation starting from the work of (Liu et al.,) and (Wang et al.,), highlighting the limitations of W-MSA and SW-MSA proposed in Swin, because of the issues with the patch size and cross-window correlation present in such techniques. Twin proposed a Spatially Separable Self-Attention mechanism (SSSA) that in the same transformer encoder layer uses in a subsequent fashion a Locally Grouped Self-Attention (LSA) and a Global Sub-Sampled Attention (GSA) mechanism.

Locally Grouped Self-Attention: given a $H \times W$ input image we divide it into $m \times n$ non overlapping windows and locally compute self attention, which aside from the fact that the windows may be rectangular is the same approach as Swin. We have a time complexity of $\mathbf{O}(\frac{H^2W^2}{m^2n^2}d)$ for each window and $\mathbf{O}(\frac{H^2W^2}{mn}d)$ in total, where d is the inner dimensionality of the model; if we let $k_1 = H/m$ and $k_2 = W/n$ be the dimensions of each window then the time complexity is $\mathbf{O}(k_1k_2HWd)$, that for fixed k_1 and k_2 is linear in the input size.

Global Sub-Sampled Attention: at this point to embedd inter-window dependencies we use a standard Self-Attention where a single representative is used to summarize all the information for each window, which reduces the time complexity to $\mathbf{O}(mnHWd) = \mathbf{O}(\frac{H^2W^2}{k_1,k_2}d)$. The total complexity of SSSA is $\mathbf{O}(k_1k_2HWd + \frac{H^2W^2}{k_1,k_2}d)$, where I recall that k_1 and k_2 are the dimensions of each window. Since both k_1 and k_2 are hyperparameters of the architecture we can look for the best value of such pair.
Since

$$k_1k_2HWd + \frac{H^2W^2}{k_1,k_2}d \geq 2HWd\sqrt{HW},$$

the left hand side of the equation reaches its minimum for k_1, k_2 such that $k_1k_2 = \sqrt{HW}$, where we get a total time complexity of $\mathbf{O}(dHW\sqrt{HW})$. Since aside from the first layer H and W are controlled by design we can always manually set k_1 and k_2 to their best value, usually with $k_1 = k_2$; while for the first layer where H and W are taken from the input image, the authors went for $k_1 = k_2 = 15$, assuming $H = W = 224$.
The most interesting design choice in the Twin architecture was not to waive on using a form of global Attention; nevertheless the time complexity was improved by a sub-linear order of magnitude, which will reveal not to be an isolated case.

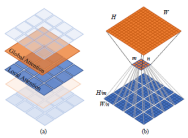

Figure 6: An illustration of the SSSA. From (Chu et al.,)

3.4 ScalableViT

ScalableViT, (Yang et al.,), was introduced to move the application of transformer's architectures for computer vision in the direction of inner hierarchical representation, in a similar way to (Wang et al.,). In the original paper the authors introduced two new attention based mechanisms designed to push performances in the direction of accuracy-cost improvements; such new techniques, which are organized alternately in each stage, are Scalable Self-Attention (SSA), meant to compute global Self-Attention introducing scale factors in all the dimensions, and Interactive Window-based Self-Attention (IWSA), meant to enlarge the receiptive field by aggregating information from a set of discrete tokens (this is conceptually analogous to GSA but implemented differently).
The architecture follows (Wang et al.,). At the i-th stage the input image of size $H_{i-1} \times W_{i-1} \times C_{i-1}$ is divided in $\frac{HW}{2^{2(i+1)}}$ patches which are linearly projected in $\mathbf{R}^{\frac{HW}{2^{2(i+1)}} \times 2^{i-1} C_i}$. ScalableViT uses a position encoding generator (PEG) to learn dynamic position embeddings.

Scalable Self-Attention: The main idea is that information in images is often shared among different tokens which could be instead aggregated leading to a performance improvement. Let r_N and r_C be two scaling factor for the spatial and channel dimension so that $N \to Nr_N$ and $C \to Cr_C$. Scalable Self-Attention can be defined as:

$$SSA(X) = A'(X)V'(X) = softmax\Big(\frac{Q'(X)^T K'(x)^T}{\sqrt{d'_k}}\Big)V'(X) \quad \text{where}$$

$$Q' = f_q(X) \in \mathbf{R}^{N \times Cr_c}, \quad K' = f_k(X) \in \mathbf{R}^{Nr_N \times Cr_c}, \quad V' = f_v(X) \in \mathbf{R}^{Nr_N \times C}$$

are the scaled query, key and value matrices; $A' \in \mathbf{R}^{N \times Nr_N}$ and f_q, f_k and f_v are the scaling operators, implemented with a convolution and a linear projection. The time complexity of SSA is $\mathbf{O}(NN_{r_N}C + NN_{r_N}C_{r_C})$, linear in N, the input size.

Interactive Window-based Self-Attention: Consider WSA which on a $H \times W \times C$ image divides it into windows of size $M \times M$ patches; let $\{Z_n\}_{n=1}^{\frac{H}{M} \times \frac{W}{M}}$ be the set of discrete values corresponding to the Self-Attention for each window and let merge them back to $Z \in \mathbf{R}^{H \times W \times C}$. Given a set of discrete values $\{V_n(X_n)\}_{n=1}^{\frac{H}{M} \times \frac{W}{M}}$, the local interactive module (LIM) reshapes them from $\mathbf{R}^{M^2 \times C}$ into $\mathbf{R}^{M \times M \times C}$ and merges them all into $V \in \mathbf{R}^{H \times W \times C}$. By means of a deepwise 3×3 convolution with zero padding (which can implicitly encode position information (Islam, Jia, & Bruce,)) we construct the operator $\mathbf{F}$, able "to establish marriages and connections between adjacent $V_n(X_n)s$" (Yang et al.,). Formally IWSA can be defined as:

$$Z' = Z + \mathbf{F}(V).$$

Since for $k = 3$ the cost of LIM is negligible and the complexity of WSA is linear it follows that the complexity of IWSA is $\mathbf{O}(N)$.

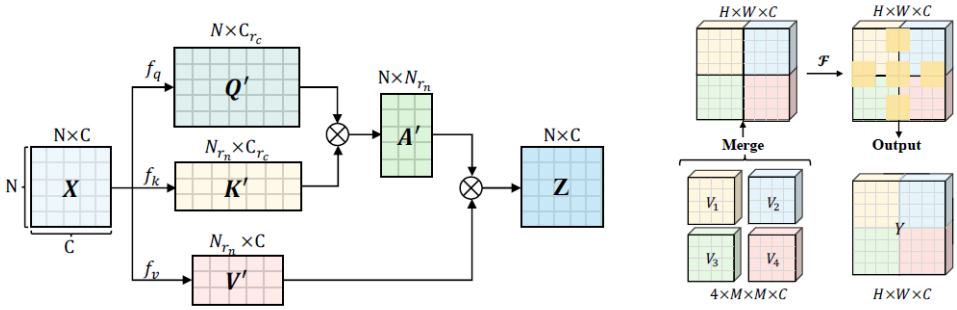

Figure 7: On the left an illustration of SSA while on the right of the LIM. From (Yang et al.,)

4 General Purpose Accelerated-Attentions Mechanisms

In the next section I will introduce some revisitations (Child et al.,), (Zhang et al.,) of the Attention mechanism proposed on top of general purpose architectures, whose modification to the Attention equations are not supported by an inductive bias induced by the specific data type but instead inspired by general approximation techniques which allow the time complexity to be reduced.

4.1 Sparse Transformer

The *Sparse Transformer*, (Child et al.,), introduced a new approach to tackle the complexity of Self-Attention based on a sparse factorization of the Attention itself (FSA). This is a general approach that is not specifically designed for any data type but was indeed tested in multiple scenarios, always with positive outcomes.
The paper considers a sequence generation task on the input sequence $\vec{x} = \{x_1, x_2, x_3, \dots, x_n\}$, where the probability for the whole sequence is modelled by a network with vector parameter $\vec{\theta}$ as:

$$p(\vec{x}) = \prod_{i=1}^{n} p(x_i | x_1, \dots, x_{i-1}, \vec{\theta}).$$

Inspired by considerations on textual and image data the authors implemented a set of sparse attention patterns which over different steps of Attention provide full connectivity among the inputs.

Factorized Self-Attention: Self-Attention maps the input embeddings X into a new representation parametrized by a connectivity pattern $S = \{S_1, S_2, \dots, S_n\}$ where S_i is the set of indices on which the i-th output vector depends on. Following the notation in the paper the output vector is formally:

$$Attend(X, S) = \{a(x_i, S_i)\}_{i=1}^{n} \quad \text{where}$$

$$a(x_i, S_i) = softmax\Big(\frac{(W_q x_i) K_{S_i}^T}{\sqrt{d}}\Big) V_{S_i},$$

where $K_{S_i} = (W_k x_j)_{j \in S_i}$, $V_{S_i} = (W_v x_j)_{j \in S_i}$ and W_q, W_k, W_v are the weight matrices that turn X into the query, key and value matrices. For Self-Attention $S_i = \{1, 2, \dots, n\}$, for SA on autoregressive models $S_i = \{j : j \leq i\}$. Since the cardinality of S_i is what determines the time complexity of SA the idea is to find a new candidate set of indices whose cardinality is not linear in the input size. FSA uses p heads $(A_i^{(m)})_{m=1}^{p} \subset \{j : j \leq i\}$ and sets $S_i = A_i^{(m)}$, where $|A_i^{(m)}| \propto n^{\frac{1}{p}}$. The time complexity of FSA, with such choice of S_i is $\mathbf{O}(n^{\frac{1+p}{p}}) \subset \mathbf{O}(n^2)$. To restrict the possible choices of the head design two conditions must be satisfied:

- All input positions must be connected to all output positions across the p steps of Attention.
- For every $j \leq i$, i can depend on j through a path with maximum length of $p + 1$.

This two criteria ensure the output to have full visibility on the input, which is equivalent to a global receiptive field as in the original Attention mechanism. In the original paper the authors provide an example of FSA with $p = 2$ by setting $A_i^1 = \{t, t+1, t+2, \dots, i | t = max(0, t - l)\}$ and
$A_i^2 = \{j : (i - j) \text{mod}(l) = 0\}$, where l is an hyperparameter, (Child et al.,).

4.2 Performer

The *Performer*, (Choromanski et al.,), was introduced with the idea to rethink the attention mechanism not by relying on a simpler mechanism but by approximating it through a proper estimator; which allowed for a general purpose technique with a linear time complexity for Attention, termed Fast Attention Via positive Orthogonal Random features (FAVOR+). This technique provides unbiased or nearly unbiased approximation of the Attention matrix with low variance (Choromanski et al., , Appendix).
FAVOR+ works on the Attention matrix $A \in \mathbf{R}^{L\times L}$ where $A_{i,j} = K(q_i, k_j^T)$ and K is a kernel such that $K(x,y) = \mathbf{E}[\phi(x)^T\phi(y)]$ and $\phi : \mathbf{R}^d \to \mathbf{R}_+^r$ is a random feature map. If Q, K, V are the query, key and value matrices then let $Q', K' \in \mathbf{R}^{L\times r}$ with rows given by $\phi(q_i), \phi(k_i)$. The the Attention is computed as:

$$\hat{\text{Att}}(Q,K,V) = \hat{D^{-1}}(Q'((K')^TV)), \quad \hat{D} = \text{diag}(Q'((K')^T1_L)),$$

which has a time complexity of $\mathbf{O}(Lrd)$, where d is the dimension of the transformer representation. To approximate the softmax in an efficient way we assume $l, m \in \mathbf{N}$, $f_1, f_2, \ldots, f_l : \mathbf{R} \to \mathbf{R}$, $h : \mathbf{R}^d \to \mathbf{R}$, $w_1, w_2, \ldots, w_m \sim \mathbf{D}$ where $D \in \mathbf{P}(\mathbf{R^d})$ and then

$$\phi(x) = \frac{h(x)}{\sqrt{m}}(f_1(w_1^Tx), \ldots, f_1(w_m^Tx), \ldots, f_l(w_1^Tx), \ldots, f_l(w_m^Tx)).$$

Since the softmax kernel can be approximated as

$$SM(x,y) = \mathbf{E}_{w\sim\mathbf{N}(0,I_d)}\Big[exp\Big(w^Tx - \frac{||x||^2}{2}\Big)exp\Big(w^Ty - \frac{||y||^2}{2}\Big)\Big]$$

$$= exp\Big(-\frac{||x||^2 + ||y||^2}{2}\Big)\mathbf{E}_{w\sim\mathbf{N}(0,I_d)}\Big[cosh(w^T(x+y)),\Big]$$

we can construct the feature map ϕ by taking: $h(x) = exp(-\frac{||x||^2}{2})$, $l = 1$, $f_1(x) = exp(x)$ and $\mathbf{D} = \mathbf{N}(0, I_d)$.
The last step suggested in the paper is to make the sampled w_is orthogonal by means of Gram-Schmidt, to reduce the variance of the estimator.

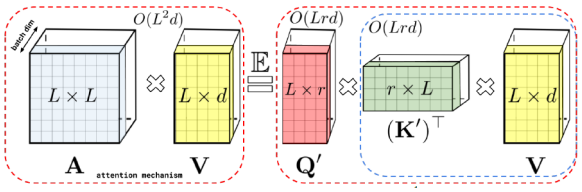

Figure 8: A comparison between classic SA and FAVOR+, where the dashed-block indicate the order of computation. From (Choromanski et al.,)

4.3 Average Attention

Average Attention (AA), (Zhang et al.,), is a mechanism that captures long range dependencies using a weighted average of the inputs, abandoning the quadratic Self-Attention. In the original paper the authors discussed the task of Neural Machine Translation, on textual data, but the approach can be generalized to any kind of data keeping its linear complexity.
Let $\vec{x} = \{x_1, x_2, \dots, x_m\}$ with $x_1 \in \mathbf{R}^d$ be the input sequence, then the cumulative weighted average is computed as:

$$g_j = FFN\Big(\frac{1}{j}\sum_{i=1}^{j} x_i\Big)$$

where a Feed-Forward Network is used to enhance the expressiveness of the model. The key idea in this approach is that all the previous inputs equally contribute (because of the shared $1/j$ factor) to the attention weight g_j and moreover each position depends on all the previous ones; combining this two properties we obtain a global mechanism to capture arbitrarily long dependencies, which substitutes the classic Attention. g_j is treated as a contextual representation for the j-th input. At this point the output is computed by means of a gating layer:

$$i_j, f_j = \sigma(W[x_j, g_j])$$
$$\widetilde{h_j} = i_j \cdot x_j + f_j \cdot g_j$$

where *sigma* is a gating layer, $[\cdot, \cdot]$ is the concatenation operator, W is a learnable weight matrix, $[i_j, f_j]$ is the input/forget gate and $\hat{h_j}$ is the hidden state at time j. In the end:

$$h_j = \text{LayerNorm}(\widetilde{h_j} + x_j).$$

The time complexity of AA is $\mathbf{O}(nd^2)$ which is again a linear algorithm. Notice that in the original paper the authors suggested to implement a masking trick to compute the cumulative weighted average in a fully parallelizable way, which at the cost of a quadratic complexity[1] allowed for a great speed-up.

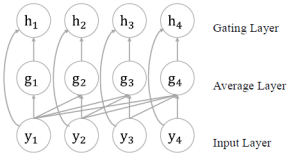

Figure 9: An illustration of the Average Attention mechanism. From (Zhang et al.,)

[1] In this case $\mathbf{O}(nd^2 + dn^2)$

References

Bahdanau, D., Cho, K., Bengio, Y. (2014). *Neural machine translation by jointly learning to align and translate.* arXiv. Retrieved from https://arxiv.org/abs/1409.0473 doi: 10.48550/ARXIV.1409.0473

Child, R., Gray, S., Radford, A., Sutskever, I. (2019). *Generating long sequences with sparse transformers.* arXiv. Retrieved from https://arxiv.org/abs/1904.10509 doi: 10.48550/ARXIV.1904.10509

Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., ... Weller, A. (2020). *Rethinking attention with performers.* arXiv. Retrieved from https://arxiv.org/abs/2009.14794 doi: 10.48550/ARXIV.2009.14794

Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., ... Shen, C. (2021). *Twins: Revisiting the design of spatial attention in vision transformers.* arXiv. Retrieved from https://arxiv.org/abs/2104.13840 doi: 10.48550/ARXIV.2104.13840

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. (2018). *Bert: Pre-training of deep bidirectional transformers for language understanding.* arXiv. Retrieved from https://arxiv.org/abs/1810.04805 doi: 10.48550/ARXIV.1810.04805

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... Houlsby, N. (2020). *An image is worth 16x16 words: Transformers for image recognition at scale.* arXiv. Retrieved from https://arxiv.org/abs/2010.11929 doi: 10.48550/ARXIV.2010.11929

Islam, M. A., Jia, S., Bruce, N. D. B. (2020). *How much position information do convolutional neural networks encode?* arXiv. Retrieved from https://arxiv.org/abs/2001.08248 doi: 10.48550/ARXIV.2001.08248

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... Guo, B. (2021). *Swin transformer: Hierarchical vision transformer using shifted windows.* arXiv. Retrieved from https://arxiv.org/abs/2103.14030 doi: 10.48550/ARXIV.2103.14030

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). *Attention is all you need.* arXiv. Retrieved from https://arxiv.org/abs/1706.03762 doi: 10.48550/ARXIV.1706.03762

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., ... Shao, L. (2021). *Pyramid vision transformer: A versatile backbone for dense prediction without convolutions.* arXiv. Retrieved from https://arxiv.org/abs/2102.12122 doi: 10.48550/ARXIV.2102.12122

Yang, R., Ma, H., Wu, J., Tang, Y., Xiao, X., Zheng, M., Li, X. (2022). *Scalablevit: Rethinking the context-oriented generalization of vision transformer.* arXiv. Retrieved from https://arxiv.org/abs/2203.10790 doi: 10.48550/ARXIV.2203.10790

Zhang, B., Xiong, D., Su, J. (2018). *Accelerating neural transformer via an average attention network.* arXiv. Retrieved from https://arxiv.org/

abs/1805.00631 doi: 10.48550/ARXIV.1805.00631