参考1

参考2

0. base

- global attention: dot product

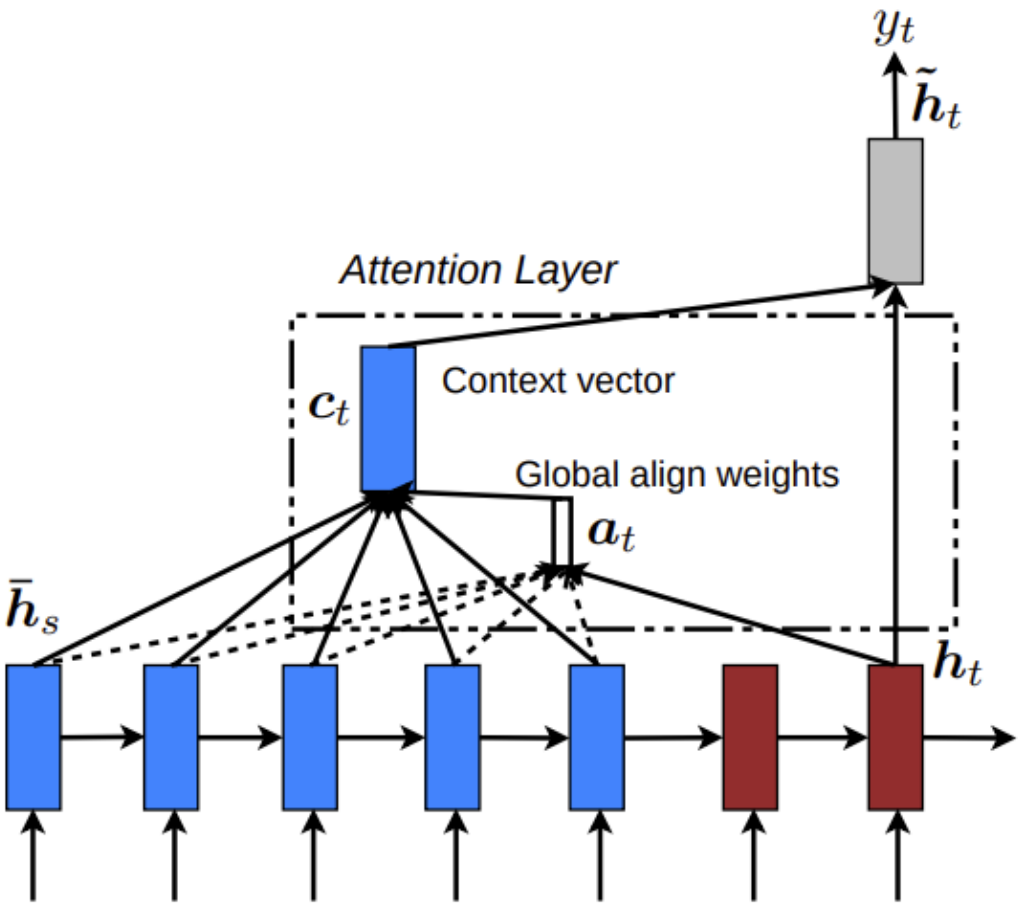


Figure 2: **Global attentional model** – at each time step t, the model infers a *variable-length* alignment weight vector $\boldsymbol{a}_t$ based on the current target state $\boldsymbol{h}_t$ and all source states $\bar{\boldsymbol{h}}_s$. A global context vector $\boldsymbol{c}_t$ is then computed as the weighted average, according to $\boldsymbol{a}_t$, over all the source states.

- local attention：general,人工经验设定的参数D去选择一个以 p**tpt为中心，$[p_t - D, p_t + D]$为窗口的区域,不随输入序列长度变化而变化，它的 维度固定为2D+1

$$at(s) = align(ht, h'_s)exp(-(s - pt)^2/2\sigma^2)$$

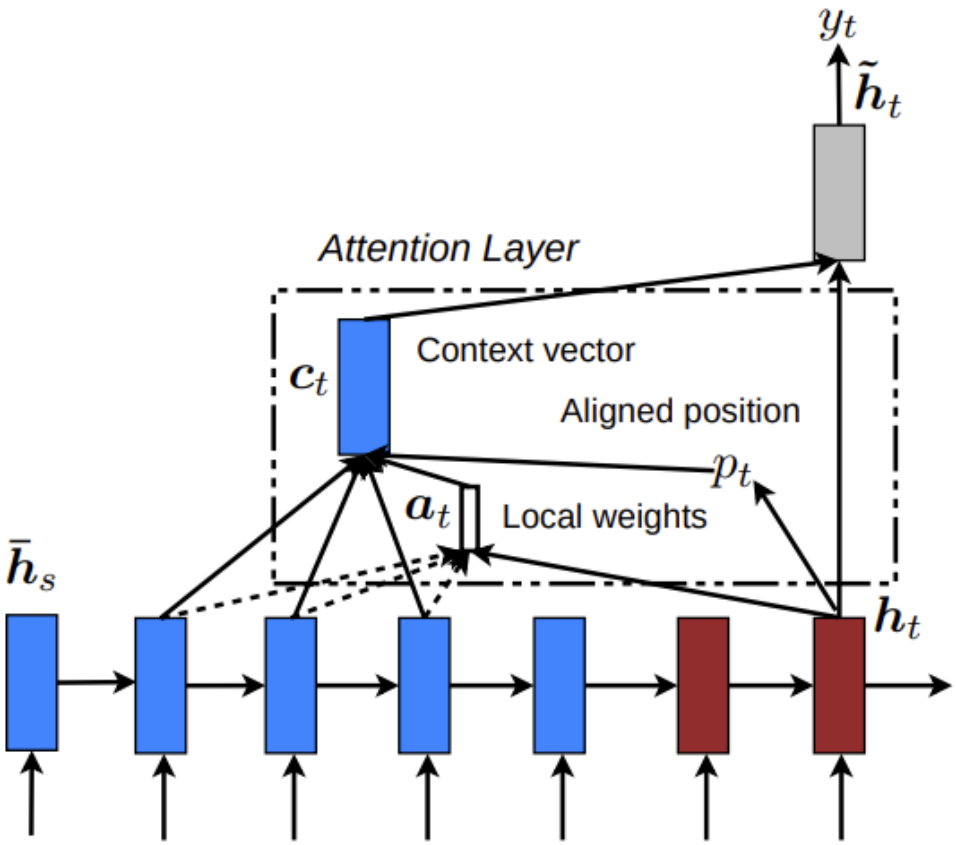
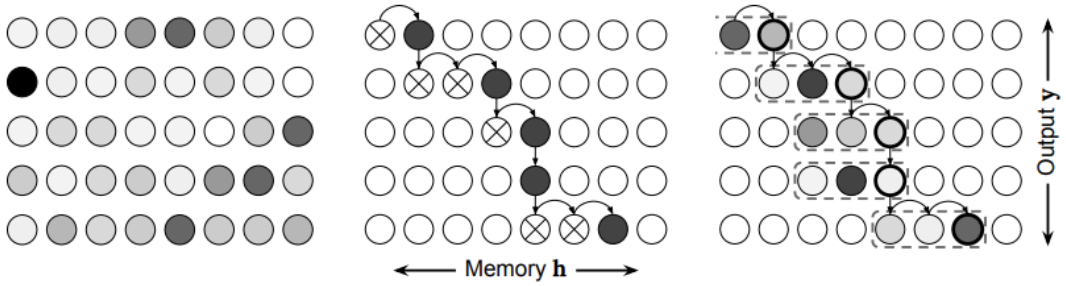


Figure 3: **Local attention model** – the model first predicts a single aligned position p_t for the current target word. A window centered around the source position p_t is then used to compute a context vector $\boldsymbol{c}_t$, a weighted average of the source hidden states in the window. The weights $\boldsymbol{a_t}$ are inferred from the current target state $\boldsymbol{h}_t$ and those source states $\bar{\boldsymbol{h}}_s$ in the window.

- non-local attention
- hard attention:不可微分，用jenson+蒙特卡洛采样
- soft attention:可微



(a) Soft attention. (b) Hard monotonic attention. (c) Monotonic chunkwise attention.

- self attention(scaled dot-product) --> multi-head self attention

> [Effective Approaches to Attention-based Neural Machine Translation,2015]
>
> [Show, Attend and Tell: Neural Image Caption Generation with Visual Attention,2015]
>
> [attention is all you need,2017]
>
> [参考1]
>
> [参考2]
>
> [参考3]
>
> [参考4]
>
> [参考5]
>
> [参考6]

> - attention：基于encoder-decoder model中，输入的source和target不一致，Attention权值的计算不仅需要Encoder中的隐状态而且还需要Decoder 中的隐状态。
> - Self Attention，Source内部元素之间或者Target内部元素之间发生的Attention机制。在Transformer中在计算权重参数时将文字向量转成对应的KQV，只需要在Source处进行对应的矩阵操作，用不到Target中的信息。

1. attention in CV

	model	idea	paper/code
♠	SASA	CNN的卷积 --> pure self-attention(local attention)	Stand-Alone Self-Attention in Vision Models,2019 code more more
♠	HaloNet	光韵attention：blocked local attention and attention downsampling vs CNN	Scaling Local Self-Attention for Parameter Efficient Visual Backbones,2021 code
	Set Transformer	set：顺序无关的任务	Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks,2019 code more more
♠	CCNet	十字交叉：Criss-Cross attention去获取criss-cross路径上的上下文	CCNet: Criss-Cross Attention for Semantic Segmentation,2020 code more more more
	Efficient Attention	线性复杂度	Efficient Attention: Attention with Linear Complexities,2020 code more
♠	Sparse Self Attention	稀疏Attention：适用于文本、图像和语音的稀疏Transformer	Generating Long Sequences with Sparse Transformers,2020 code more
			paper code

	Convolution Attention	convolution+attention	CBAM: Convolutional Block Attention Module,2018
	Pyramid Attention	源于Pyramid Pooling技巧	Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition,2015
	Pyramid Pooling	上面的补充	Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition,2018
	Co Attention	成对的输入，Question Answering，Video Object Segmentation	See More, Know More: Unsupervised Video Object Segmentation With Co-Attention Siamese Networks,2019
	Cross Attention		Cross Attention Network for Few-shot Classification,2019
	Compositional Attention	多种形式的Attention，CoDA的核心就是在于构造相似度	Compositional De-Attention Networks,2019
	Dual Attention	多种形式的attention	Dual Attention Network for Scene Segmentation

> transformer in cv

> 参考1
>
> code

1.1 SASA_model

- CNN的卷积(卷积核权重共享) --> pure self-attention(local)
- 提出代替空间卷积的操作--self attention，以弥补CNN无法捕获长距离信息的问题。解决用全局注意力层作为卷积的附加模块带来的大开销；代替卷积，提出独立自注意力层和空间感知独立自注意力层
- 位置编码上：相对位置信息 > 绝对位置信息 > 没有位置的信息
- global attention的计算效率高，但是实际运行时间长。自注意力层定义了三个概念query、key和value，自注意力的运算是局部的，所以不用限制输出的大小。自注意力层的参数个数与感受野大大小无关，卷积的参数个数与感受野的大小成平方关系；运算量的增长也比卷积的缓慢。
- self attention平等对待中心像素邻近的其他像素点，没有利用位置信息，因此文中进一步通过用嵌入向量来表示相对位置，把位置信息也添加到了自注意力操作中。

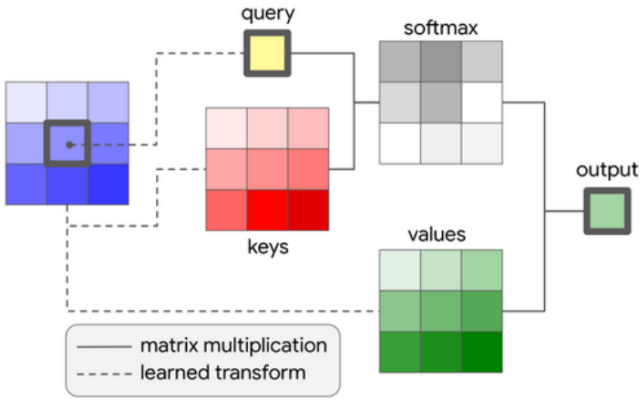


Figure 3: An example of a local attention layer over spatial extent of $k = 3$.

-1, -1	-1, 0	-1, 1	-1, 2
0, -1	0, 0	0, 1	0, 2
1, -1	1, 0	1, 1	1, 2
2, -1	2, 0	2, 1	2, 2

Figure 4: An example of relative distance computation. The relative distances are computed with respect to the position of the highlighted pixel. The format of distances is *row offset*, *column offset*.

Similar to a convolution, given a pixel $x_{ij} \in \mathbb{R}^{d_{in}}$, we first extract a local region of pixels in positions $ab \in \mathcal{N}_k(i, j)$ with spatial extent k centered around x_{ij}, which we call the *memory block*. This form of local attention differs from prior work exploring attention in vision which have performed global (i.e., all-to-all) attention between all pixels [32, 33]. Global attention can only be used after significant spatial downsampling has been applied to the input because it is computationally expensive, which prevents its usage across all layers in a fully attentional model.

Single-headed attention for computing the pixel output $y_{ij} \in \mathbb{R}^{d_{out}}$ is then computed as follows (see Figure 3):

$$y_{ij} = \sum_{a,b \in \mathcal{N}_k(i,j)} \texttt{softmax}_{ab}\left(q_{ij}^{\top} k_{ab}\right) v_{ab} \tag{2}$$

where the *queries* $q_{ij} = W_Q x_{ij}$, *keys* $k_{ab} = W_K x_{ab}$, and *values* $v_{ab} = W_V x_{ab}$ are linear transformations of the pixel in position ij and the neighborhood pixels. $\texttt{softmax}_{ab}$ denotes a softmax applied to all logits computed in the neighborhood of ij. $W_Q, W_K, W_V \in \mathbb{R}^{d_{out} \times d_{in}}$ are all learned transforms. While local self-attention aggregates spatial information over neighborhoods similar to convolutions (Equation 1), the aggregation is done with a convex combination of value vectors with mixing weights ($\texttt{softmax}_{ab}(\cdot)$) parametrized by content interactions. This computation is repeated for every pixel ij. In practice, multiple attention *heads* are used to learn multiple distinct representations of the input. It works by partitioning the pixel features x_{ij} depthwise into N groups $x_{ij}^n \in \mathbb{R}^{d_{in}/N}$, computing single-headed attention on each group separately as above with different transforms $W_Q^n, W_K^n, W_V^n \in \mathbb{R}^{d_{out}/N \times d_{in}/N}$ per head, and then concatenating the output representations into the final output $y_{ij} \in \mathbb{R}^{d_{out}}$.

1.2 HaloNet

- local attention更快速、更少memory、更高精度
- translational equivariance,To increase expressivity, multi-headed attention is used

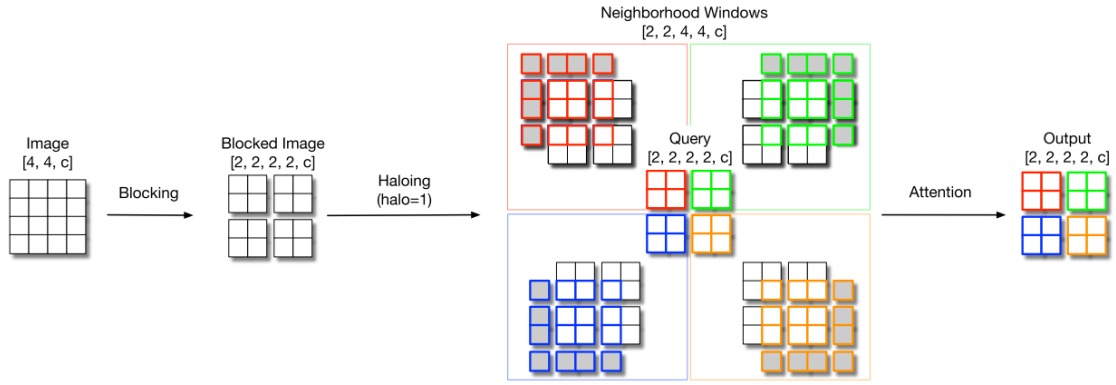


Figure 1. **HaloNet local self-attention architecture: The different stages of blocked local attention for a** $[4, 4, c]$ **image, block size** $b = 2$**, and halo** $h = 1$. The image is first blocked into non-overlapping $[2, 2, c]$ images from which the queries are computed. The subsequent haloing step then extracts a $[4, 4, c]$ memory around each of the blocks which linearly transform to keys and values. The spatial dimensions after attention are the same as the queries.

$$f(i,j,a,b)^{conv} = W_{a-i,b-j} \tag{1}$$

$$f(i,j,a,b)^{self-att} = \texttt{softmax}_{ab}\Big((W_Q x_{ij})^\top W_K x_{ab} + (W_Q x_{ij})^\top r_{a-i,b-j}\Big) W_V \tag{2}$$

$$= p^{ij}_{a-i,b-j} W_v \tag{3}$$

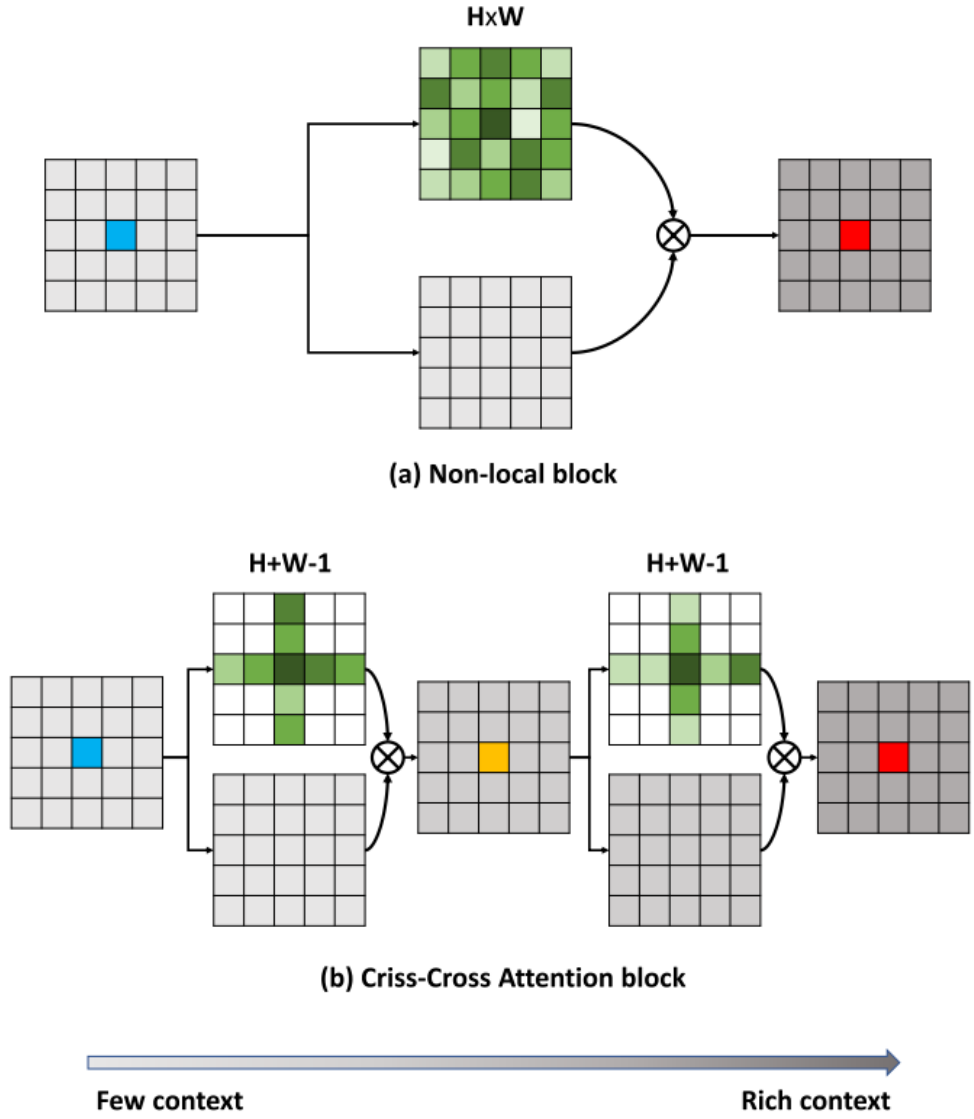
Figure 1 depicts the different steps involved in executing blocked local self-attention for an image with height $H = 4$, width $W = 4$, and c channels with stride 1. Blocking chops up the image into a $\frac{H}{b}, \frac{W}{b}$ tensor of *non-overlapping* (b, b) blocks. Each block behaves as a group of query pixels and a *haloing* operation combines a band of h pixels around them (with padding at boundaries) to obtain the corresponding *shared neighborhood* block of shape $(\frac{H}{b}, \frac{W}{b}, b + 2h, b + 2h, c)$ from which the keys and values are computed. $\frac{H}{b} \times \frac{W}{b}$ attention operations then run in parallel for each of the query blocks and their corresponding neighborhoods, illustrated with different colors in Figure 1. SASA [43] used the same blocking strategy[3], setting $h = \lfloor \frac{k}{2} \rfloor$ and uses attention masks to emulate pixel-centered neighborhood windows of size

[2]To illustrate this, on a 128×128 resolution with 64 channels, global self-attention would incur about 28 times more FLOPs than a 3×3 convolution with 64 input and output channels

[3]Code for both SASA and HaloNet will be made available, along with the checkpoints for HaloNet

1.3 CCNet

- 从水平和竖直方向聚合long-range上下文信息，对一个像素捕捉到全局的contextual information
- 计算量少



convolution attention

2. attention in NLP

	model	idea	paper/code
♠	transformer		code
	memory attention		paper code
			paper code
			paper code
			paper code
			paper code
			paper code
			paper code
			paper code
			paper code
			paper code
			paper code
			paper code
			paper code
			paper code
		multi-head is all you need?	Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned,2019
	ABCNN	CNN+attention 多种组合	ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs,2018 more

multi-head is all you need?

- Positional Head：这 个Head计算的权值通常指向临近的词，规则是这个Head在90%的情况下都会把最大的权值分配给左边或者右边的一个词。
- Syntactic Head：这个Head计算的权值通常会将词语之间的关系联系起来，比如名词和动词的指向关系。
- Rare Head：这个Head通常会把大的权值分配给稀有词。

然后文章还分析了如何去精简Heads，优化的方法如下（给各个Head加个权值，相当于门）：

$$MultiHead = concat\left(g_1 * head_1, g_2 * head_2, \cdots, g_n * head_n\right)$$

然后去优化 g_i ，加入 $L0$ 稀疏正则化损失：

$$L0(g) = \sum_{i=1}^{n}(1 - I[g_i = 0])$$

最后为了近似 $I[g_i = 0] = p(g_i = 0|\phi_i)$，其中 ϕ_i 的优化可以采用Gumbel Softmax技巧，因此每次训练的时候相当于使用了Sampling的技巧从多个Head里面选出来一部分进行运算，而Sampling的依据（参数 ϕ_i ）则可以通过Gumbel Reparametrization的技巧来实现。

> bert_review

3. attention in ASR/TTS

CTC

		paper code

4. attention in other

https://github.com/lucidrains/alphafold2