Sliding Window Attention

Lukas Bierling

October 8, 2024

1 Introduction

To overcome the limitations imposed by the quadratic complexity of standard attention mechanisms, it is essential to sparsify the attention matrix. This entails to design a scheme whereby each token x_i for $i \in 0, 1, \ldots, n$ attends selectively to a subset of tokens indexed by $S_i \subseteq \{0, 1, \ldots, n\}$. Such an approach ensures that not every token is attended to by every other, thereby reducing the computational burden of each attention operation. This reduction introduces a critical trade-off: the smaller the size of S_i, the lower the computational cost, yet with the increasing risk of omitting interactions crucial for capturing the intended semantics of the input.
The selection of the subset S_i need not be static but can vary depending on the token at position i. Our objective is to find a set of positions S_i for each token such that the resulting sparsified attention matrix closely approximates the full attention matrix in the best possible way.

2 Sliding window Attention

One way to sparsify the attention matrix is to use sliding windows as fixed pattern around the current token x_i. For a fixed window size w each token attends to $\frac{w}{2}$ tokens on each side (Beltagy, Peters, and Cohan [1]). The set S_i, representing the indices of tokens within the window of x_i, can be formalized as:

$$S_i = \{i - \frac{w}{2}, \ldots, i, \ldots, i + \frac{w}{2}\} \tag{1}$$

The authors of the paper which introduced the concept explained: "Using multiple stacked layers of such windowed attention results in a large receptive field, where top layers have access to all input locations and have the capacity to build representations that incorporate information across the entire input, similar to CNNs"(Beltagy, Peters, and Cohan [1]). This can also be seen as a giving the model an inductive bias where human assumptions are coded into the model's structure and architecture. By doing this the model is forced to learn in a specific way, here by limiting its attention to a local neighborhood. The window size w may also vary across different layers and as such is a tunable

hyperparameter.
We adjust the current computation of the attention matrix $\mathbf{S}$ with:

$$\mathbf{S}_i^{\mathrm{w}}(q_i, k_{S_i}) = softmax(\frac{q_i {k_{S_i}}^{\mathrm{T}}}{\sqrt{d_k}}) \tag{2}$$

for each token x_i. To compute the final attention output $\mathbf{A}^{\mathrm{w}}$ we need to consider the local context of the attention matrix $\mathbf{S_i}^{\mathrm{w}}$ and therefore start by calculating $\mathbf{A_i}^{\mathrm{w}}$ as:

$$\mathbf{A_i}^{\mathrm{w}}(q_i, k_{S_i}, v_{S_i}) = \mathbf{S_i}^{\mathrm{w}}(q_i, k_{S_i}) v_{S_i} \tag{3}$$

To finalize the attention output, each token-wise attention output $\mathbf{A_i}^{\mathrm{w}}$ is concatenated row-wise:

$$\mathbf{A}^{\mathrm{w}}(Q, K) = Concat_{row}(\mathbf{A}_0^{\mathrm{w}}(q_0, k_{S_0}, v_{S_0}), \ldots, \mathbf{A}_n^{\mathrm{w}}(q_n, k_{S_n}, v_{S_n})) \tag{4}$$

where $q_i \in R^{1 \times \mathrm{d_{model}}}$ is the query vector for token x_i and $k_{S_i} \in R^{\mathrm{w} \times \mathrm{d_{model}}}$ is a subset of the key matrix K that corresponds to the window S_i containing only the keys that token x_i is allowed to attend to and finally, $v_{S_i} \in R^{\mathrm{w} \times \mathrm{d_{model}}}$ the subset of the value matrix V. The sliding-window attention matrix is of shape $n \times w$ instead of $n \times n$ and the final attention output is again of shape $n \times d_{model}$

Since each token attends to w other tokens the complexity of the sliding window attention mechanism can formally described as:

$$\mathbf{S}^{\mathrm{w}} = \mathcal{O}(nw) \tag{5}$$

and thus scales linearly with the input sequence length n. To ease the understanding in the following section, a set H is defined which contains all the additional hyperparameters that have to be considered using the novel approaches. Currently H can be described as: $\mathrm{H} = \{\mathrm{W}\}$ where W contains the window sizes w_l for each layer l. Refer to Figure 13 to understand sliding window attention visually.

3 Dilated sliding window attention

Similar to dilated (or atrous) CNNs (Chen et al. [2]) where some values are skipped with a fixed rate d when computing the convolution, the concept of dilation can also be applied to the attention mechanism. The sliding-window approach of the previous section is constrained on short distances since the model only is able to attend to tokens inside the window size w making it impossible to capture long-range dependencies. To overcome this limitation dilated sliding window attention was introduces. Instead of using a sliding window of size w the window is extended to size dw but with gaps of size $d-1$ between tokens (Beltagy, Peters, and Cohan [1]). This means each token x_i still attends to a total number of w tokens but the the Set S_i changes to

$$S_i = \{i - \frac{dw}{2}, i - \frac{dw}{2} + d, \ldots, i, \ldots i + \frac{dw}{2} - d, i + \frac{dw}{2}\} \tag{6}$$

The range to which tokens x_i increases from $\frac{w}{2}$ in both direction to $\frac{dw}{2}$ in both directions giving the model a larger context to work with and learn from. This making it suitable for very long input sequences without a significant increase in computational costs.

Since each token still only attends to w other tokens the complexity of the dilated sliding window attention mechanism can formally be described also as:

$$\mathbf{S}^{\mathrm{w}} = \mathcal{O}(nw) \tag{7}$$

The computation for $\mathbf{A}^{\mathrm{w}}$ and everything else stays the same. The only thing that is changed is the set S_i. Consequently, H is adjusted to: $\mathrm{H} = \{\mathrm{W}, \mathrm{D}\}$ where D contains the dilation rates d_l for each layer l. Refer to Figure 14 for a visualized explanation.

4 Global attention connection

For masked LMs the model uses the context around the masked token to predict it. In BERT-style models an additional [CLS]-token is added at the beginning which represents the whole input sequence for further tasks like classification or clustering. The goal is to encode the the information of the input sequence inside the [CLS]-token embedding of the last attention block. When using (dilated) sliding window attention, the attention computed is not flexible enough to learn 'good' task-specific representations. Therefore global attention is added. Global attention in this context refers to the concept of adding full attention computations on few pre-selected input locations. This is a symmetric operation: A token with global attention attends to all tokens across the sequence and all tokens in the sequence attend to it (Beltagy, Peters, and Cohan [1]).

In our case, the windowed and dilated attention are not flexible enough to learn task-specific representations. Accordingly, we add "global attention" on few pre-selected input locations. Importantly, we make this attention operation symmetric: that is, a token with a global attention attends to all tokens across the sequence, and all tokens in the sequence attend to it (Beltagy, Peters, and Cohan [1]). The set L contains all the positions of token with global attention. That is for all $i \in G$, the following is stated: $\mathrm{S}_i = S$ which means that the local attention subset equals the full attention set. Additionally, since this is a symmetric operation (Beltagy, Peters, and Cohan [1]) each element of G has to be added to each S_i stating: $\mathrm{S}_i = S_i \cup L$ The size of L, i.e. the number of global attention tokens is small relative to the independent tokens n the complexity of the combined local and global attention computation is still $\mathcal{O}(nw)$ (Beltagy, Peters, and Cohan [1]).
The hyperparameter set H is adjusted accordingly: $\mathrm{H} = \{\mathrm{W,D,L}\}$ Figure 15 provides a visualized explanation of the global attention.

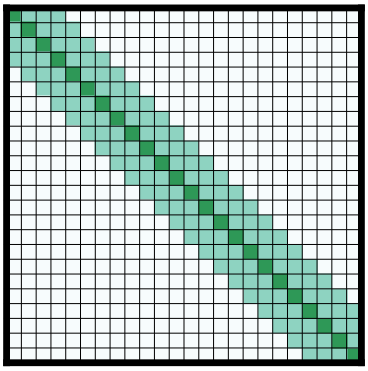

Figure 1: Sliding window attention. Each tokenx_i only attends to its local neighborhood. In this case $w = 6$ (Beltagy, Peters, and Cohan [1]).

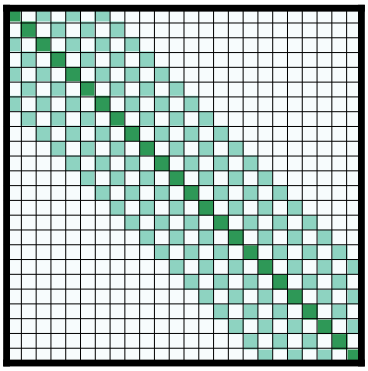

Figure 2: Dilated sliding window attention with $w = 6$ and $d = 2$. There is a gap of $d - 1 = 1$ between each token in the local neighborhood (Beltagy, Peters, and Cohan [1]).

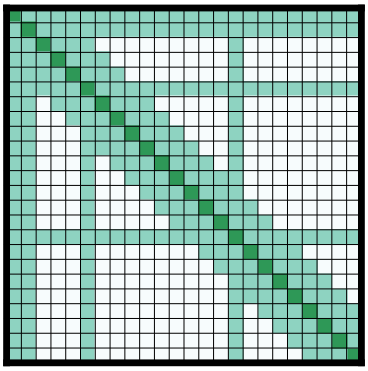

Figure 3: Global attention combined with sliding-window attention for $w = 6$, $L = \{0, 1, 5, 15\}$ and $d = 1$. The first row is the corresponding [CLS]-token which uses global attention to capture the information of the whole input sequence (Beltagy, Peters, and Cohan [1])

.