Assignment 6: Transformers

1 Part 1 Encoder-Decoder

1. Q1: *How do you think the architecture in Figure1 will perform on long sequences, and why? Consider the amount of information the decoder gets to see about the input sequence*

   I think this architecture will perform bad on long sentence. Because in this architecture we will include the meaning of the entire sentence in a single vector.

2. Q2: What modification can we make to this architecture to improve its performance on long sequences? Explain how this modification facilitates more accurate translation on long sequences.

   We can use attention as a remedy to this architecture, specifically, for the decoder, we can use the attention weights to generate the context vector by taking a weighted sum of the encoder hidden vectors, and concatenate the context vector to the decoder hidden vector. This can get more accurate results on long sentence because the context vector can get a distribution over the vocabulary on the target language size for the prediction.

2 Part 2 Transformers vs RNNs

1. *Q3: Transformers have allowed the NLP community to train on larger datasets than was once possible, what aspect of the Transformer makes this possible in contrast to RNNs?*

   Unlike RNNs that are inherently sequential, Transformers are based entirely on attention, which can process all tokens at the same time can calculating attention weights. This can enable parallelization for training on larger datasets. Also there are only matrix multiplications in transformers. These features make training Transformers more efficient than RNNs.

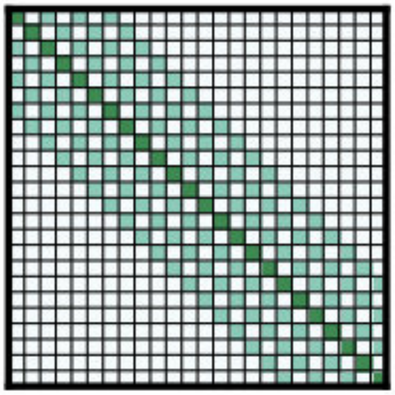
3 Part 3 Self-Attention

2. *Q4: Under what conditions might the performance of window attention be at a strong disadvantage in comparison to full self-attention? You can reason using specific examples, or you can provide your argument in more abstract terms. ?*

   I think in the conditions that that there are less compution to do (e.g. the length of the query is relatively small, the depth of layers is relatively shallow), the performance of window attention is at a strong disadvantage in comparison to full self-attention. This is because using window attention is the balance between efficiency and performance, if there are less compution to do, window attention lacks much representation power compared to full self-attention.

3. *Q5: Discuss another self-attention pattern (other than full-self attention and window attention). This can be your own novel pattern, we encourage you to think of your own, but you can also consult the literature for help. Explain one strength and one potential limitation of your chosen attention pattern. You may choose to show a visual similar to Figure 2 to illustrate your attention pattern*

Here I discuss another self-attention pattern called dilated sliding window attention (reference Beltazy et al, 2020). Compared to sliding window attention that taking consecutive neighbors in the fixed size window, dilated sliding window attention taking elements with fixed size of gaps, which is shown in the following graph(the graph comes from the literature).



(c) Dilated sliding window

One strength for this is that compared to sliding window attention, this method can increase receptive field without increasing computation or memory, therefore we can cover larger queries with fewer layers. One limitation for this is that using entirely dilated sliding window attention may lose local context/information in shallow layers, therefore the author in the literature explored a method to use both local and global attention by combined attention pattern.

4 Part 4 Attention Functions

In Part 3, we used dot product attention. Name an alternative attention function, and explain its strengths and weaknesses with respect to dot product attention.

Here I name addtive attention(reference Vaswani et al, 2017), which is shown as below:

$$e_i = v^T tanh(W_1 h_i + W_2 s)$$

Compared to dot product attention, the strength is that when the dimension of the queries and keys is very large, additive attention outperforms dot product attention without scaling; the weaknesses is that using highly optimized matrix multiplication code, dot-product attention is much faster and more space-efficient than additive attention.

5 Reference

[1] Beltagy, Iz, Matthew E. Peters, and Arman Cohan. "Longformer: The long-document transformer." *arXiv preprint arXiv:2004.05150* (2020).

[2] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).