Paper Implementation Report

Attention Is All You Need

Ⅰ. Implementation of Transformer's key contributions

The key contribution of this paper is that they implemented NMT with better performance than existing RNN and CNN methods **with novel attention methods** only and yielded **a more interpretable model by utilizing self-attention masks**. To demonstrate these contributions, I conducted two major experiments. The first was to **reproduce the results** of **(A)** and **(C)** as described *Table 3* in the paper, which differ in the number of multi-head-attention racers and the encoder and decoder layers, respectively. The next one was **to visualize each attention matrix to verify its interpretability.** I tried to **royally follow the implementation details** in this paper as far as possible. As noted in the paper, I applied **warmup learning rate scheduling and label smoothing**. I applied warmup scheduling up to *4000 steps* and set the epsilon of label smoothing to *0.1*. In addition, I added **gradient clipping**, which improved BLEU by about ***2%***. Due to time and resource limitations, we used *Multi30K* data instead of *WMT14* applying 10K train steps based on *64* batch sizes.

Ⅱ. Experiment results

ⅰ. Reproduce performance of transformer using Multi30K dataset.

	N	d_{model}	d_{ff}	h	d_k	d_v	P_{drop}	ϵ_{ls}	*train steps*	BLEU (test)	params $\times 10^6$
base	6	512	2048	8	64	64	0.1	0.1	10K	45.06	54
(A)				1	512	512				45.20	
				4	128	128				45.09	
				16	32	32				45.37	
				32	16	16				45.61	
(C)	2									45.94	24
	4									44.37	39
	8									44.80	68
		256			32	32				42.75	22
		1024			128	128				**49.50**	146
			1024							46.00	41
			4096							44.65	79
big	6	1024	4096	16			0.3		30K	24.54	196

Table 1. The performance reproduction of variations on the Transformer architecture for (A), (C) and big

As can be seen from Table 1, the overall BLEU scores of (A) and (C) were similar, but the results changed most sensitively when changing the $\boldsymbol{d_{model}}$. In more detail, as the number of headers increases, a slight performance improvement can be seen. In contrast, as the number of layers increases, the performance decreases. The performance was best when only the $\boldsymbol{d_{model}}$ was increased, and the performance was fatally reduced for the big model.

ⅱ. Visualize each attention matrix.

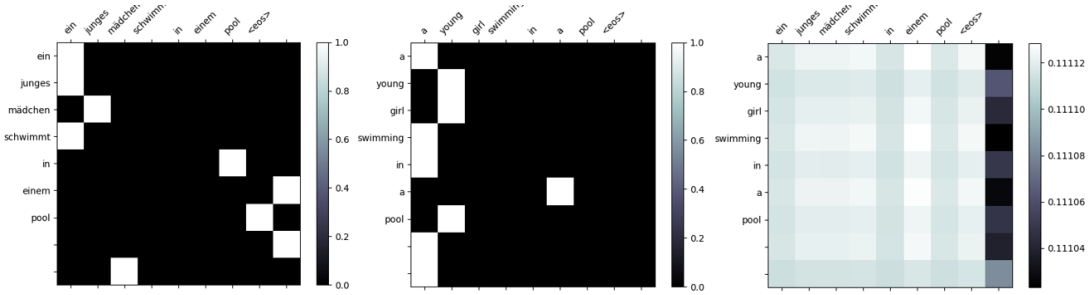

Figure 1. Visualization of each attention matrix.
(left: encoder self-attention, middle: decoder self-attention, right: encoder-decoder attention)

Figure 1 shows the visualization of the attention matrix of the first header of the first layer. In detail, the self-attention matrix tends to weight itself or the next word, and articles such as 'a' affect all words. In encoder-decoder attention, it can be seen that the source word affects the target word evenly, and the unique word has a relatively large attention value.

Ⅲ. Discussion

Through the above experiments, we can confirm that NMT can be implemented by attention alone, and it leaves more room for interpretation than existing models. However, since I was experimenting with Multi30K rather than WMT14, the results were somewhat different from the paper. Especially the most interesting result was the decrease of performance for big model. For the big model, I trained with 3 times more train steps, expecting the best performance because it has same $\boldsymbol{d_{model}}$ with the best performance model, but the performance dropped by nearly twice. It seems that the model has been overfitted due to excessive training, since Multi30K is not that large dataset.