Evaluating Linformer's performance on the WMT14 EN-DE machine translation task

Marco Pampaloni

Department of Computer Science

October 7, 2024

Contents

1 Introduction

The Transformer architecture, since its introduction in 2017 [1], has revolutionized the field of natural language processing (NLP), reaching state of the art in various downstream tasks. Despite its massive parallelism capabilities, the Transformer struggles with longer sequence lengths due to its attention mechanism, which scales quadratically with the number of input tokens. This is a problem at both training and inference time.

For this reason, there has been a huge research effort in these years to develop faster attention mechanisms, either exploiting the IO properties of the hardware these models run on [2], or by approximating the result of scaled dot product attention (SDPA).

The *Linformer* architecture [3] is an example of the latter approach. The authors first empirically show that the attention matrix is often low-rank, meaning that it could be approximated with an SVD decomposition by only keeping the singular values with larger magnitude. This would of course introduce additional asymptotical complexity to the method, so the authors propose the adoption of linear projections on the keys and values matrices K, V to reduce their dimensionality and drive the computational complexity of the attention mechanism from $O(n^2)$ to $O(kn)$. The authors further show that the choice of k does not depend on the sequence length n, so that the scaling can be considered linear in n.

The standard multi head attention (MHA) introduced by [1] is computed as follows

$$\begin{aligned}\text{MultiHead}(Q,K,V) &= \text{Concat}(\text{head}_1,\ldots,\text{head}_h)W^O \\ \text{where head}_i &= \text{Attention}(QW^Q, KW^K, VW^V) \\ &= \underbrace{\text{softmax}\left(\frac{QW^Q(KW^K)^T}{\sqrt{d_{model}}}\right)}_{P_i} VW^V\end{aligned} \tag{1}$$

The Linformer attention first projects the key and value matrices into a dimension of space $\mathbb{R}^{k\times d}$ with projection matrices $E, F \in \mathbb{R}^{k\times n}$ and then computes MHA as before:

$$\overline{\text{head}}_i = \text{Attention}\left(Q, E_i K W_i^K, F_i V W_i^V\right) \tag{2}$$

This produces an attention matrix $\bar{P}_i \in \mathbb{R}^{n\times k}$, which is computed in time linear with n. Since the projection matrices E, F are fixed in size before training, an obvious donwside of this approach is that the maximum sequence length n has to be known beforehand and the model cannot then scale to inputs with more tokens than this value. One workaround is to set the maximum sequence length n to a large number and handle shorter inputs by slicing the projection matrices along their columns before multiplying them with the inputs K and V.

1.1 Masking

1.1.1 Causal masking

2 Prior work

3 Data

4 Architecture

5 Experiments

6 Results and Analysis

6.1 Model performance

Model	PPL (dev)	BLEU (dev)
Transformer	**3.59**	**26.7**
Linformer (k=32)	-	-
Linformer (k=64)	-	-

6.2 Training time

6.3 Inference time

7 Conclusions

Bibliography

References

[1] Ashish Vaswani et al. *Attention Is All You Need*. Dec. 5, 2017. arXiv: `1706.03762 [cs]`. URL: `http://arxiv.org/abs/1706.03762` (visited on 05/21/2023). Prepublished.

[2] Tri Dao et al. *FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness*. June 23, 2022. DOI: `10.48550/arXiv.2205.14135`. arXiv: `2205.14135 [cs]`. URL: `http://arxiv.org/abs/2205.14135` (visited on 07/06/2023). Prepublished.

[3] Sinong Wang et al. *Linformer: Self-Attention with Linear Complexity*. June 14, 2020. DOI: `10.48550/arXiv.2006.04768`. arXiv: `2006.04768 [cs, stat]`. URL: `http://arxiv.org/abs/2006.04768` (visited on 09/15/2024). Pre-published.