DCT-Former: Efficient self-attention with Discrete Cosine Transform

Carmelo Scribano, Giorgia Franchini, Marco Prato, Marko Bertogna

Department of Physics, Informatics and Mathematics,
University of Modena and Reggio Emilia, Modena (Italy)

UNIMORE – Università degli Studi di Modena e Reggio Emilia
High Performance Real Time Lab

➢ Transformers Self-Attention

- Self-Attention [2] maps a set of tokens $X \in E^{n\times d}$ to a similar set $\overline{X} \in E^{n\times d}$ where each rows of $\overline{X}$ is obtained as a weighted combination of all the rows of X.
- The Attention's weights (Energy, E) represents the affinity degree between pairs of tokens. To obtain the energy score, X is projected into Queries (Q) Keys (K) and Values (V):

$$Q = XW_Q, \qquad K = XW_K, \qquad V = XW_V$$

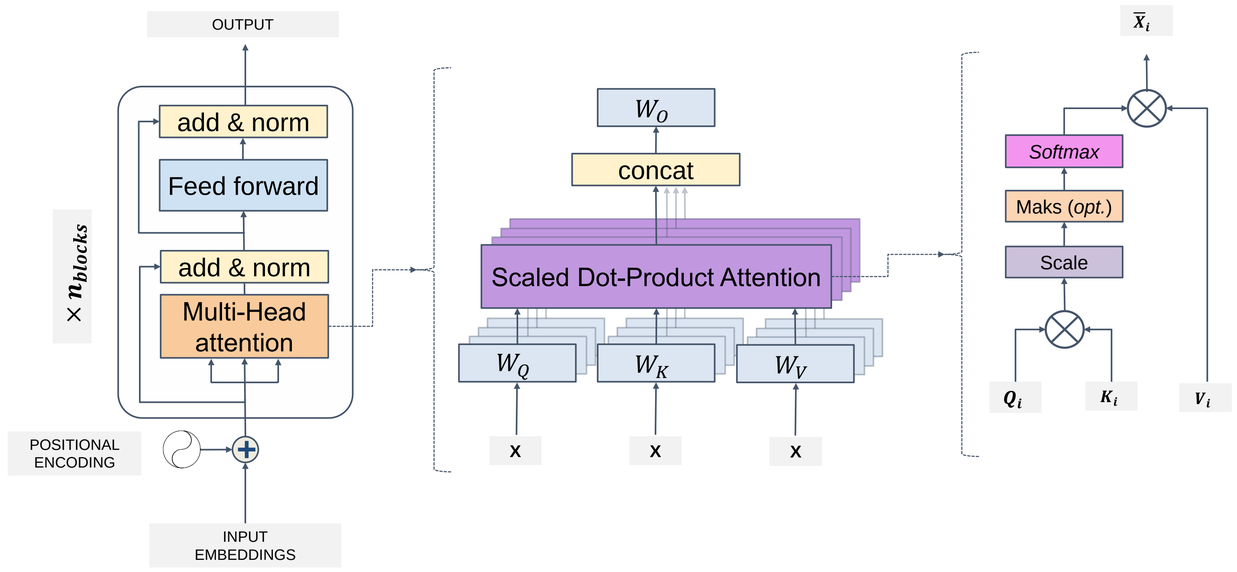
The Energy is then obtained as the normalized dot-product of Q and K^T, with a row-wise *softmax* operation to force the result in (0,1).

$$E(Q,K) = softmax\left(\frac{QK^T}{\sqrt{d}}\right) \quad (1)$$

Finally, the Attention's output is obtained (for a single head) by multiplying by V:

$$\overline{X} = Atn(X) = E(Q,K)V \quad (2)$$

Limitation: $\boldsymbol{E} \in \mathbb{R}^{\boldsymbol{n\times n}}$ as such, the attention cost grows quadratically with the input sequence length!



➢ Discrete Cosine Transform

- Type-II DCT is widely used in **data compression** (JPEG, MPEG etc....).
- Given a finite sequence of N real-valued elements $\hat{x} \in \mathbb{R}^{N\times 1}$ the DCT is defined as:

$$\hat{X}_K = a_k \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi(2n+1)k}{2N}\right) \quad for\ k = 0,1,\dots,N-1 \quad (3)$$

$$\text{where} \quad a_k = \begin{cases} \sqrt{1/N} & \text{if } k = 0 \\ \sqrt{2/N} & \text{if } k \neq 0 \end{cases}$$

- Being a *linear* transformation the DCT can be expressed as dot-product between the sequence $\hat{x}$ and a transformation matrix $\boldsymbol{D} \in \mathbb{R}^{\boldsymbol{N\times N}}$(with D orthogonal thanks to the normalization term a_k).
- A simple compression algorithm can be implemented by storing only $\overline{\boldsymbol{N}} < \boldsymbol{N}$ DCT coefficients, a *lossy* reconstruction of $\hat{x}$ can be computed using the inverse transform.
- The 2-dimensional DCT of a sequence $\hat{x}_{2d} \in \mathbb{R}^{N\times N}$ can be conveniently expressed as a row-wise and a column-wise DCT:

$$\hat{X}_{2d} = D\hat{x}_{2d}D^T \quad (4)$$

➢ Results

- Inference results are expressed in terms of GPU's peak memory occupation (MB) and inference speed (ms). The results are reported for different batch sizes.

Attention Head	Sequence length (N) – Batch size (BS)							
	128-256		512-32		1024-16		4096-1	
	MB	ms	MB	Ms	MB	ms	MB	ms
Vanilla [2]	21.25	0.45	171.67	2.1	397.9	5.3	6451.7	45.7
DCT-0.25	**15.0**	**0.37**	**178.7**	**1.45**	**261.4**	**3.0**	**2957.0**	**16.89**
Linformer-0.125	18.5	0.433	233.85	1.71	382.56	3.7	4740.0	20.36
Nyström-0.125	20.1	0.468	270.5	1.9	524.38	4.45	8008.6	51.53
Performer-0.125	20.17	0.48	259.8	1.98	448.1	4.4	6192.3	30.7

- For pre-training the Accuracy on the MLM task is reported, **Normalized** refers to the accuracy score normalized by the (average) inference time.
- For fine-tuning binary classification metrics are reported.
- In the fine-tuning stage the DCT is computed online using the Makhoul's algorithm, thus not relying on the matrix formulation. This allow to effectively work with inputs of variable length without padding.

Attention	Pretraining			Finetuning ↑		
	Loss ↓	Accuracy ↑	Normalized ↑	Precision	Recall	F1-Score
Vanilla [2]	2.07	59.7	1.32	0.9	0.9	0.9
DCT-16	2.58	51.6	1.46	-	-	-
DCT-32	2.36	54.7	**1.48**	0.87	0.87	0.87
DCT-48	2.28	56.0	1.35	0.86	0.85	0.85
DCT-64	2.24	56.6	1.30	0.85	0.85	0.85
Linformer-16 [4]	2.29	56.2	1.3	0.80	0.80	0.80
Linformer-32	2.17	57.9	1.22	0.82	0.82	0.82
Linformer-48	2.13	58.5	1.23	0.83	0.83	0.83
Nyström-16 [5]	2.25	56.6	1.2	0.88	0.87	0.87
Nyström-32	2.13	58.8	1.11	0.88	0.8	0.88

➢ Efficient Attention with DCT

- **Key idea:** leverage a DCT based approximation to compute a compressed version of (1).
- Q, K and V from (1) are replaced by their first $\bar{n} \ll n$ DCT coefficients:

$$\bar{Q} = \bar{D}Q, \qquad \bar{K} = \bar{D}K, \qquad \bar{V} = \bar{D}V \quad \text{with } D \in \mathbb{R}^{n\times\bar{n}}$$

- Substituting in (1), recalling (4):

$$\bar{E}(\bar{Q},\bar{K}) = softmax\big((\bar{D}Q)(K^T\bar{D}^T)\big) = softmax\big(DCT_{2D}(QK^T)\big)$$

(The normalization term $\sqrt{d}$ is omitted for clearness)

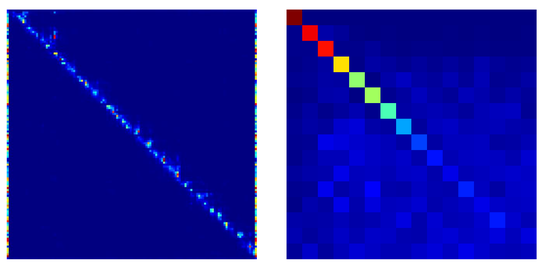
- Since $\bar{E} \in \mathbb{R}^{\bar{n}\times\bar{n}}$ and QK^T is never explicitly computed the quadratic bottleneck is avoided!
- The (lossy) attention's output is obtained from (2) using $\bar{V}$ and computing a final inverse DCT:

$$\tilde{X} = \bar{D}^T[\bar{E}(\bar{Q},\bar{K})\bar{V}]$$

- An important **relaxation** is leveraged, since $softmax\big(DCT(x)\big) \neq DCT\big(softmax(x)\big)$

Algorithm 1 Efficient Attention with DCT

Input $X \in \mathbb{R}^{n\times d}$
Output $\tilde{X} \approx Atn(X)$
Require: $\overline{D} \in \mathbb{R}^{\bar{n}\times n}$
1: $\overline{X} = DCT(X) = \overline{D}X$
2: $\overline{Q} = \overline{X}W_Q, \quad \overline{K} = \overline{X}W_K, \quad \overline{V} = \overline{X}W_V$
3: $Atn(\overline{Q},\overline{K},\overline{V}) = E(\overline{Q},\overline{K})\overline{V} = softmax\left(\frac{\overline{QK}^T}{\sqrt{d}}\right)\overline{V}$
4: $\tilde{X} = \overline{D}^T\left[Atn(\overline{Q},\overline{K},\overline{V})\right]$
5: **return** $\tilde{X}$



➢ (left) Attention's Energy matrix: **128x128** (right) DCT Attention's Energy: **16x16**

➢ Experimental Setup

- Our methodology is evaluated in a common **NLP** scenario, building a BERT-*like* [3] model. To maintain the training cost reasonable a smaller transformer model is employed ($n_{blocks} = 4,\ n_{heads} = 8$).
- The **pre-training** stage optimize a self-supervised task of Masked-Language-Modelling (MLM) on English Wikipedia text:

> **Masked sentence:** *"[CLS] How are [MSK] doing today? [SEP]"*
> **MLM target:** "*[CLS] How are you doing today? [SEP]*"

- The model is then **fine-tuned** on the downstream task of sentiment classification on the IMDb movies reviews dataset:

> **Sentence:** *"[CLS] I love this movie like no other [SEP]"* → 👍
> **Sentence:** "*[CLS] Unbelievably disappointing! [SEP]"* → 👎

➢ References

1. Scribano, C., Franchini, G., Prato, M., Bertogna, M.,: DCT-Former**: Efficient Self-Attention with Discrete Cosine Transform**. arXiv preprint arXiv:2203.01178 (2022).
2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: **Attention is all you need.** In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017).
3. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: **Bert**: Pre-training of deep bidirectional transformers for language understanding. NAACL-HLT 2019: 4171-4186.
4. Wang, Sinong, et al. "Linformer: Self-attention with linear complexity." arXiv preprint arXiv:2006.04768 (2020).
5. Xiong, Y., Zeng, Z., Chakraborty, R., Tan, M., Fung, G., Li, Y., & Singh, V. (2021). Nyströmformer: A Nyström-based Algorithm for Approximating Self-Attention. AAAI 2021