TENSOR DECOMPOSITION TECHNIQUES FOR TRANSFORMERS PROJECT REPORT

Guntas Singh Saran[*]**& Hrriday V. Ruparel** [†]**& Sumeet Sawale**[‡]
Department of Computer Science and Engineering
Indian Institute of Technology Gandhinagar
Palaj, GJ 382355, India

Prof. Anirban Dasgupta
Department of Computer Science and Engineering
Indian Institute of Technology Gandhinagar
Palaj, GJ 382355, India

1 INTRODUCTION

Transformer models [Vaswani et al. (2023)] have achieved state-of-the-art results across a diverse range of domains, including natural language, conversation, images, and even music. The core block of every Transformer architecture is the attention module, which computes similarity scores for all pairs of positions in an input sequence. This however, scales poorly with the length of the input sequence, requiring quadratic computation time to produce all similarity scores, as well as quadratic memory size to construct a matrix to store these scores Choromanski et al. (2022).

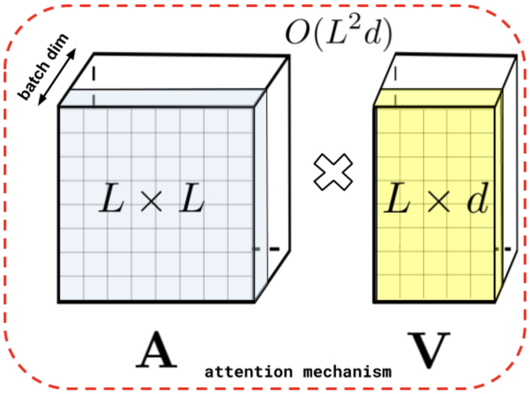


Figure 1: Standard attention module computation, where the final desired result is computed by performing a matrix multiplication with the attention matrix A and value tensor V. Source [Choromanski et al. (2022)]

Several papers have been proposed targeting ***Linear Attention***, and then we will look at the flaws and redundancies introduced in the introduction of tensors into transformer architecture for the Attention Module.

[*] Guntas Singh Saran - 22110089 - guntassingh.saran@iitgn.ac.in
[†] Hrriday V. Ruparel 22110099 - hrriday.ruparel@iitgn.ac.in
[‡] Sumeet Sawale 22110234 - sumeet.sawale@iitgn.ac.in
*Order based on Roll Numbers Only.

2 Linear Attention using Kernels

2.1 Random Features for Large-Scale Kernel Machines [Rahimi & Recht (2007)]

The kernel trick is a simple way to generate features for algorithms that depend only on the inner product between pairs of input points. It relies on the observation that any positive definite function $k(\mathbf{x},\mathbf{y})$ with $\mathbf{x},\mathbf{y} \in \mathbb{R}^d$ defines an inner product and a lifting ϕ so that the inner product between lifted datapoints can be quickly computed as $\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle = k(\mathbf{x},\mathbf{y})$. The cost of this convenience is that the algorithm accesses the data only through evaluations of $k(\mathbf{x},\mathbf{y})$, or through the kernel matrix consisting of k applied to all pairs of datapoints. As a result, large training sets incur large computational and storage costs. Instead of relying on the implicit lifting provided by the kernel trick, we propose explicitly mapping the data to a low-dimensional Euclidean inner product space using a randomized feature map $\mathbf{z} : \mathbb{R}^d \to \mathbb{R}^D$ so that the inner product between a pair of transformed points approximates their kernel evaluation:

$$\boxed{k(\mathbf{x},\mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle \approx \mathbf{z}(\mathbf{x})'\mathbf{z}(\mathbf{y})}$$

Random Fourier Features. Each component of the feature map $\mathbf{z}(\mathbf{x})$ projects $\mathbf{x}$ onto a random direction $\boldsymbol{\omega}$ drawn from the Fourier transform $p(\boldsymbol{\omega})$ of $k(\Delta)$, and wraps this line onto the unit circle in $\mathbb{R}^2$. After transforming two points $\mathbf{x}$ and $\mathbf{y}$ in this way, their inner product is an unbiased estimator of $k(\mathbf{x},\mathbf{y})$.

To obtain a real-valued random feature for k, note that both the probability distribution $p(\boldsymbol{\omega})$ and the kernel $k(\Delta)$ are real, so the integrand $e^{j\boldsymbol{\omega}'(\mathbf{x}-\mathbf{y})}$ may be replaced with $\cos(\boldsymbol{\omega}'(\mathbf{x}-\mathbf{y}))$. Defining $\mathbf{z}_{\boldsymbol{\omega}}(\mathbf{x}) = [\cos(\mathbf{x}) \ \sin(\mathbf{x})]'$ gives a real-valued mapping that satisfies the condition $\mathbb{E}[\mathbf{z}_{\boldsymbol{\omega}}(\mathbf{x})'\mathbf{z}_{\boldsymbol{\omega}}(\mathbf{y})] = k(\mathbf{x},\mathbf{y})$, since $\mathbf{z}_{\boldsymbol{\omega}}(\mathbf{x})'\mathbf{z}_{\boldsymbol{\omega}}(\mathbf{y}) = \cos(\boldsymbol{\omega}'(\mathbf{x}-\mathbf{y}))$.

Other mappings such as $\mathbf{z}_{\boldsymbol{\omega}}(\mathbf{x}) = \sqrt{2}\cos(\boldsymbol{\omega}'\mathbf{x} + b)$, where $\boldsymbol{\omega}$ is drawn from $p(\boldsymbol{\omega})$ and b is drawn uniformly from $[0, 2\pi]$, also satisfy the condition $\mathbb{E}[\mathbf{z}_{\boldsymbol{\omega}}(\mathbf{x})'\mathbf{z}_{\boldsymbol{\omega}}(\mathbf{y})] = k(\mathbf{x},\mathbf{y})$. We can lower the variance of $\mathbf{z}_{\boldsymbol{\omega}}(\mathbf{x})'\mathbf{z}_{\boldsymbol{\omega}}(\mathbf{y})$ by concatenating D randomly chosen $\mathbf{z}_{\boldsymbol{\omega}}$ into a column vector $\mathbf{z}$ and normalizing each component by $\sqrt{D}$.

Let $\phi : \mathbb{R}^d \to \mathbb{R}^{2D}$ be a nonlinear transformation:

$$\phi(\mathbf{x}) = \frac{1}{\sqrt{D}}[\sin(\mathbf{w}_1 \cdot \mathbf{x}), \ldots, \sin(\mathbf{w}_D \cdot \mathbf{x}), \cos(\mathbf{w}_1 \cdot \mathbf{x}), \ldots, \cos(\mathbf{w}_D \cdot \mathbf{x})].$$

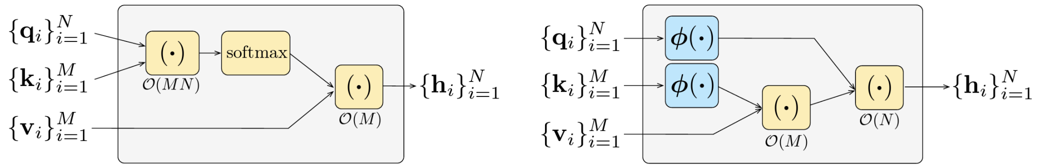
When d-dimensional random vectors $\mathbf{w}_i$ are independently sampled from $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_d)$,

$$\mathbb{E}_{\mathbf{w}_i}[\phi(\mathbf{x}) \cdot \phi(\mathbf{y})] = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right).$$

2.2 Random Feature Attention [Peng et al. (2021)]

To extend the idea of Rahimi & Recht (2007), the authors of this paper noticed that SOFTMAX function $\sigma(x_i) = \frac{\exp(x_i)}{\sum_k \exp(x_k)}$ in itself encapsulates the exp kernel. Let $\{\mathbf{q}_t\}_{t=1}^N$ denote a sequence of N query vectors, that attend to sequences of M key and value vectors. At each timestep, the attention linearly combines the values weighted by the outputs of a softmax:

$$\text{attn}(\mathbf{q}_t, \{\mathbf{k}_i\}, \{\mathbf{v}_i\}) = \frac{\exp(\mathbf{q}_t \cdot \mathbf{k}_i/\tau)}{\sum_j \exp(\mathbf{q}_t \cdot \mathbf{k}_j/\tau)}\mathbf{v}_i.$$


$\{\mathbf{q}_i\}_{i=1}^N$
$\{\mathbf{k}_i\}_{i=1}^M$
$\{\mathbf{v}_i\}_{i=1}^M$
$(\cdot)$
$\mathcal{O}(MN)$
softmax
$(\cdot)$
$\mathcal{O}(M)$
$\{\mathbf{h}_i\}_{i=1}^N$
$\{\mathbf{q}_i\}_{i=1}^N$
$\{\mathbf{k}_i\}_{i=1}^M$
$\{\mathbf{v}_i\}_{i=1}^M$
$\phi(\cdot)$
$\phi(\cdot)$
$(\cdot)$
$\mathcal{O}(M)$
$(\cdot)$
$\mathcal{O}(N)$
$\{\mathbf{h}_i\}_{i=1}^N$

(a) Softmax attention.

(b) Random feature attention.

Figure 2: Computation graphs for softmax attention (left) and random feature attention (right). Here, we assume cross attention with source length M and target length N. Source [Peng et al. (2021)]

RFA builds on an unbiased estimate of $\exp(\langle\cdot,\cdot\rangle)$ from Theorem 1, which we begin with:

$$\begin{aligned}\exp\left(\frac{\mathbf{x}\cdot\mathbf{y}}{\sigma^2}\right) &= \exp\left(\frac{\|\mathbf{x}\|^2}{2\sigma^2}+\frac{\|\mathbf{y}\|^2}{2\sigma^2}\right)\exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right)\\ &\approx \exp\left(\frac{\|\mathbf{x}\|^2}{2\sigma^2}+\frac{\|\mathbf{y}\|^2}{2\sigma^2}\right)\phi(\mathbf{x})\cdot\phi(\mathbf{y}).\end{aligned}$$

The last line does not have any nonlinear interaction between $\phi(\mathbf{x})$ and $\phi(\mathbf{y})$, allowing for a linear time/space approximation to attention. For clarity, we assume the query and keys are unit vectors. Thus:

$$\begin{aligned}\operatorname{attn}(\mathbf{q}_t,\{\mathbf{k}_i\},\{\mathbf{v}_i\}) &= \sum_i \frac{\exp\left(\mathbf{q}_t\cdot\mathbf{k}_i/\sigma^2\right)}{\sum_j \exp\left(\mathbf{q}_t\cdot\mathbf{k}_j/\sigma^2\right)}\mathbf{v}_i^T\\ &\approx \sum_i \frac{\phi(\mathbf{q}_t)^T\phi(\mathbf{k}_i)\mathbf{v}_i^T}{\sum_j \phi(\mathbf{q}_t)\cdot\phi(\mathbf{k}_j)}\\ &= \frac{\phi(\mathbf{q}_t)^T\sum_i\phi(\mathbf{k}_i)\otimes\mathbf{v}_i}{\phi(\mathbf{q}_t)\cdot\sum_j\phi(\mathbf{k}_j)} = \operatorname{RFA}(\mathbf{q}_t,\{\mathbf{k}_i\},\{\mathbf{v}_i\}).\end{aligned}$$

Here, $\otimes$ denotes the outer product between vectors, and σ^2 corresponds to the temperature term τ.

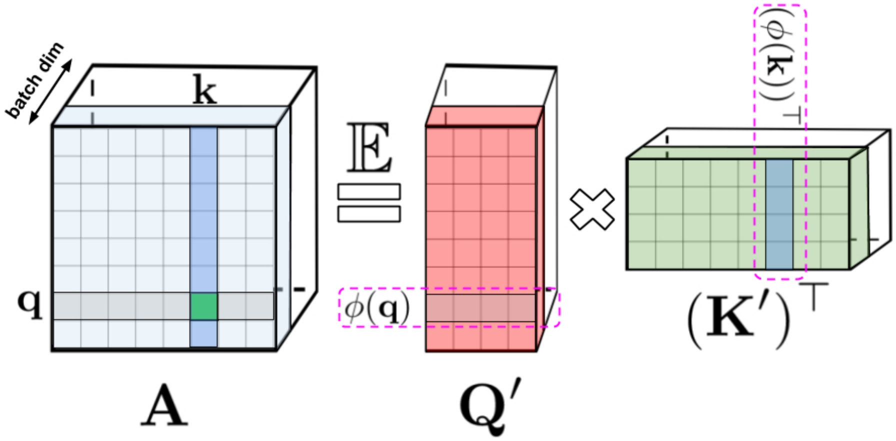


Figure 3: The standard attention matrix can be approximated via lower-rank randomized matrices Q' and K' with rows encoding potentially randomized nonlinear functions of the original queries/keys. For the regular softmax-attention, the transformation is very compact and involves an exponential function as well as random Gaussian projections. Source [Choromanski et al. (2022)]

2.3 Rethinking Attention with Performers [Choromanski et al. (2022)]

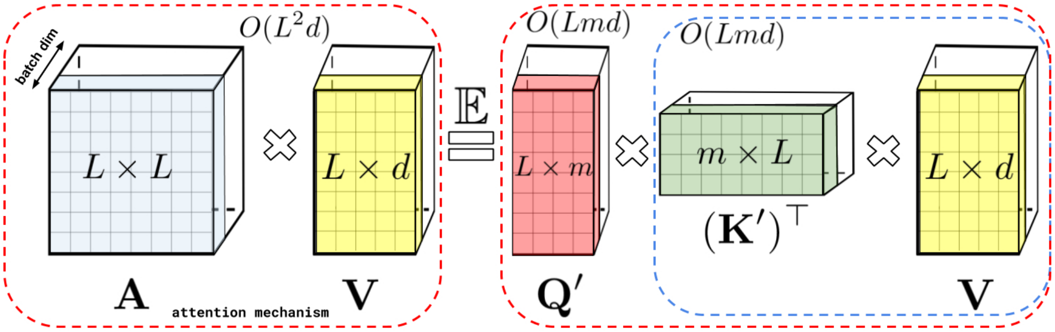


Figure 4: Standard attention module computation, where the final desired result is computed by performing a matrix multiplication with the attention matrix $\mathbf{A}$ and value tensor $\mathbf{V}$. *Right:* By decoupling matrices $\mathbf{Q}'$ and $\mathbf{K}'$ used in lower rank decomposition of $\mathbf{A}$ and conducting matrix multiplications in the order indicated by dashed-boxes, we obtain a linear attention mechanism, never explicitly constructing $\mathbf{A}$ or its approximation. Source [Choromanski et al. (2022)]

It turns out that by taking ϕ of the following form for functions $f_1, \ldots, f_l : \mathbb{R} \to \mathbb{R}$, a function $g : \mathbb{R}^d \to \mathbb{R}$, and deterministic vectors $\boldsymbol{\omega}_i$ or $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_m$ i.i.d. $\sim D$ for some distribution $D \in \mathcal{P}(\mathbb{R}^d)$:

$$\phi(\mathbf{x}) = \frac{h(\mathbf{x})}{\sqrt{m}}\big(f_1(\boldsymbol{\omega}_1 \cdot \mathbf{x}), \ldots, f_1(\boldsymbol{\omega}_m \cdot \mathbf{x}), \ldots, f_l(\boldsymbol{\omega}_1 \cdot \mathbf{x}), \ldots, f_l(\boldsymbol{\omega}_m \cdot \mathbf{x})\big),$$

we can model most kernels used in practice. Furthermore, in most cases D is isotropic (i.e., with a pdf function constant on a sphere), usually Gaussian. By taking $h(\mathbf{x}) = 1$, $l = 1$, and $D = \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, we obtain estimators of the so-called PNG-kernels (Choromanski et al., 2017), e.g., $f_1 = \text{sgn}$ corresponds to the angular kernel. - Configurations $h(\mathbf{x}) = 1$, $l = 2$, $f_1 = \sin$, $f_2 = \cos$ correspond to shift-invariant kernels. In particular, $D = \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ leads to the Gaussian kernel K_{gauss}.

The authors gave their own version of the SOFTMAX kernel with strong theoretical guarantees: unbiased or nearly-unbiased estimation of the attention matrix, uniform convergence and low estimation variance.

2.4 A Tensorized Transformer for Language Modeling [Ma et al. (2019)]

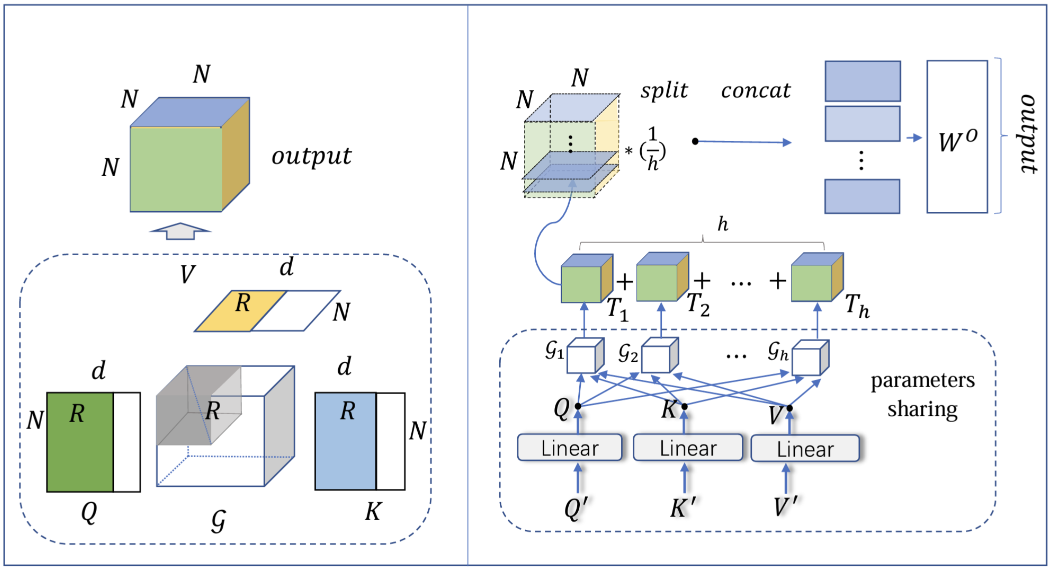


Figure 5: (left) Single-block attention using Tucker decomposition. (right) Multi-linear attention based on Block-Term tensor decomposition. Source [Ma et al. (2019)]

2.4.1 Theorem (As Stated in their Supplementary Material)

Let $Q, K, V \in \mathbb{R}^{n \times d}$ be the query, key, and value matrices, respectively. Assume that the query and key matrices Q and K are formed using a set of orthonormal basis vectors $\{e_1, e_2, \ldots, e_n\}$:

$$Q = E \cdot \alpha, \quad K = E \cdot \beta, \quad V = E \cdot \xi$$

where $E = [e_1, e_2, \ldots, e_n] \in \mathbb{R}^{n \times n}$ is the matrix of orthonormal basis vectors, and $\alpha, \beta, \xi \in \mathbb{R}^{n \times d}$ are coefficient matrices.

The scaled dot-product attention is given by:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right) V$$

The supplementary material claims that the following equality holds for a unitary matrix U:

$$\text{softmax}\left(EAE^\top\right) = E\,\text{softmax}(A)\,E^\top$$

where $A = \alpha \cdot \beta^\top$ represents the product of the coefficient matrices of Q and K. As a result, the attention representation can be rewritten as:

$$\text{Attention}(Q, K, V) = (e_1, e_2, \ldots, e_n) \cdot \text{softmax}\left(\frac{A}{\sqrt{d}}\right) \cdot (\xi_1, \xi_2, \ldots, \xi_d)^\top$$

2.4.2 Why the Theorem is Incorrect

The claim that:

$$\text{softmax}\left(UAU^\top\right) = U\,\text{softmax}(A)\,U^\top$$

is incorrect because the **softmax function is non-linear**, and non-linear functions do not generally commute with matrix multiplications or unitary transformations. Specifically:

2.4.3 Non-linearity of the Softmax Function

The softmax function is defined as:

$$\text{softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

where z_i is the i-th component of a vector $\mathbf{z}$. Applying softmax to a matrix involves exponentiation and normalization of each row, which makes it a non-linear operation.

Unitary transformations U preserve norms and inner products but do not preserve non-linear operations like softmax. Therefore, the softmax of a transformed matrix is **not equivalent** to applying the softmax to the matrix first and then transforming it.

2.4.4 Counterexample

Consider a random matrix $QK^\top \in \mathbb{R}^{n\times n}$. Perform a singular value decomposition (SVD):

$$QK^\top = U\Sigma V^\top$$

where U and V are unitary matrices, and Σ is a diagonal matrix of singular values.

The claim implies:

$$\text{softmax}(QK^\top) = U\,\text{softmax}(\Sigma)\,V^\top$$

This is false because softmax operates element-wise, and the rows of $QK^\top$ do not retain the same structure as Σ after applying the softmax function. This discrepancy can be experimentally verified.

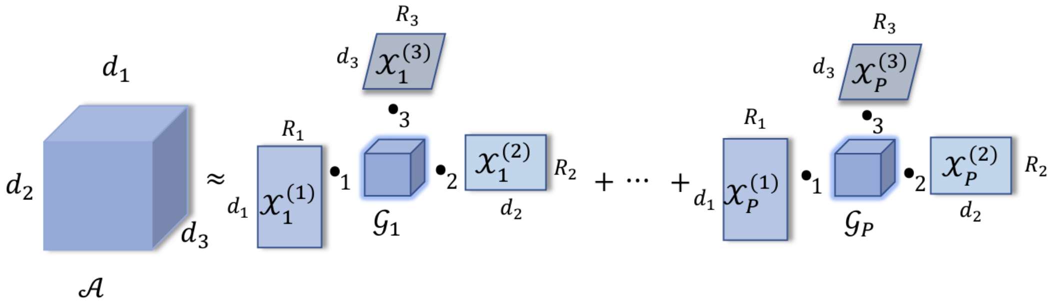


Figure 6: The representation of Block-Term tensor decomposition for a 3-order tensor. $\mathcal{A} \in \mathbb{R}^{d_1\times d_2\times d_3}$ is a 3-order tensor, and can be approximated by P Tucker decomposition. P is the CP rank, and R_1, R_2, R_3 are the Tucker rank, respectively. In the paper, authors assume that $R = R_1 = R_2 = R_3$. Source [Ma et al. (2019)]

All this demonstrates that their claim is wrong:

$$\text{softmax}(QK^\top) \neq U\,\text{softmax}(\Sigma)\,V^\top$$

2.4.5 Experimental Verification (Python Example)

```
import numpy as np

def custom_softmax(A, B, d):
    return np.exp(A @ B.T / np.sqrt(d)) / np.exp(A @ B.T /
    np.sqrt(d)).sum(axis=1)[:, None]

d, n = 3, 5 # Model Dimension and Number of Basis Vectors

E = np.random.randn(n, n)
Q, R = np.linalg.qr(E) # QR Decomposition to get Orthonormal Matrix Q

alpha = np.random.randn(n, d)
beta = np.random.randn(n, d)
gamma = np.random.randn(n, d)

Queries = Q @ alpha
Keys = Q @ beta
Values = Q @ gamma
softmax = custom_softmax(Queries, Keys, d)

softmax_alpha_beta = custom_softmax(Q @ alpha, Q @ beta, d)

softmax_proposed = custom_softmax(alpha, beta, d)
softmax_proposed = Q @ softmax_proposed @ Q.T

res = np.allclose(softmax_alpha_beta, softmax_proposed)
```

```
True True

QK^T
[[ 0.25277615 -0.0098578   0.3692536   0.2330351  -0.06745566]
 [ 0.33297648  0.5575941   0.321273    0.32699524  0.03367966]
 [-0.01159045 -0.51220186 -0.71230766 -0.56545564 -0.18237266]
 [-0.05710365 -0.30886214  0.32759332  0.14063831 -0.02289952]
 [-0.56656211 -0.47856675  0.00313731 -0.10354725  0.10493576]]

Softmax(QK^T)
[[0.2174391  0.16721575 0.24429983 0.21318872 0.1578566 ]
 [0.20098141 0.25159736 0.19864294 0.19978288 0.14899542]
 [0.28408135 0.17219873 0.14096948 0.1632684  0.23948204]
 [0.18181669 0.1413502  0.26711945 0.22157052 0.18814315]
 [0.13503035 0.14745086 0.23869796 0.21454391 0.26427691]]

Softmax(QK^T) = Softmax(E @ Alpha @ Beta^T @ E^T)
[[0.2174391  0.16721575 0.24429983 0.21318872 0.1578566 ]
 [0.20098141 0.25159736 0.19864294 0.19978288 0.14899542]
 [0.28408135 0.17219873 0.14096948 0.1632684  0.23948204]
 [0.18181669 0.1413502  0.26711945 0.22157052 0.18814315]
 [0.13503035 0.14745086 0.23869796 0.21454391 0.26427691]]

Proposed = E @ Softmax(Alpha, Beta) @ E^T
[[ 0.09204761  0.07530701 -0.09638922 -0.08063838  0.08131486]
 [ 0.11262331  0.20274226 -0.17649771 -0.11554653  0.12477391]
 [-0.14262831 -0.34549666  0.43647629  0.31476903 -0.33152286]
 [-0.07793599 -0.18608544  0.37641874  0.25305831 -0.13666747]
 [-0.04682374  0.0198849  -0.25761946 -0.19885507  0.13074815]]

Are the two softmaxes equal? False
```

2.4.6 Analysis of their Tensor Decomposition Approach

They state that under the same conditions as in their previous **flawed theorem** and the value of N is equal to the value of d, the Single-block attention representation can reconstruct the Scaled Dot-Product attention by the summing over the tensor according to the **second index**. It holds that:

$$\text{Attention}\left(Q, K, V\right)_{i,m} = \sum_{j=1}^{N} \text{Attn}_{TD}\left(\mathcal{G}; Q, K, V\right)_{i,j,m}$$

where i, j, and m are the indices of the Single-block attention output (i.e., a 3-order tensor). $\text{Attn}_{TD}(\cdot)$ is the function of the Single-block attention based on Tucker decomposition. i and m are the output indices.

Their Proof. Represent the self-attention function in by the form as follows:

$$\text{Attention}\left(Q, K, V\right) = \Theta Q K^T V$$

where Θ is a normalization factor matrix, which can be used to replace the use of a `softmax` function. They assume that Θ contains all the non-zero elements of the core tensor $\mathcal{G}$. The self-attention in Eq. 2 (in the paper) can be re-written as follows:

$$X_{i,m} = \sum_{k=1}^{N} \sum_{r=1}^{R} \Theta_{i,m} Q_{i,r} K_{k,r} V_{k,m}$$

where N is the length of a sentence, $X_{i,m} = \text{Attention}\left(Q, K, V\right)_{i,m}$ is the entry of the output from the self-attention, and R is equal to d. Here the core tensor is $\mathcal{G}$. Then, the Single-block attention (a 3-order tensor) can be represented as follows:

$$\mathcal{A}_{i,j,m} = \sum_{p}^{R} \sum_{q}^{R} \sum_{r}^{R} \mathcal{G}_{p,q,r} Q_{i,p} K_{j,p} V_{m,r}$$

where $\mathcal{A}$ is a 3-order tensor, which is equal to $\text{Attn}_{TD}\left(\mathcal{G}; Q, K, V\right)$. Consequently, $\mathcal{A}_{i,j,m}$ is an entry in the tensor $\mathcal{A}$ and is equal to $\text{Attention}_{TD_{i,j,m}}$. Next, since the core tensor $\mathcal{G}$ is a special tensor (i.e., diagonal tensor), they wrote the above equation as:

$$\mathcal{A}_{i,j,m} = \sum_{r=1}^{R} g_{r,r,r} Q_{i,r} K_{j,r} V_{m,r}$$

$$X_{i,m} = \sum_{r=1}^{R} \sum_{j=1}^{N} \mathcal{G}_{rrr} Q_{i,r} K_{j,r} V_{m,r}$$

After that, they computed the attention representation by summing over the second index, j.

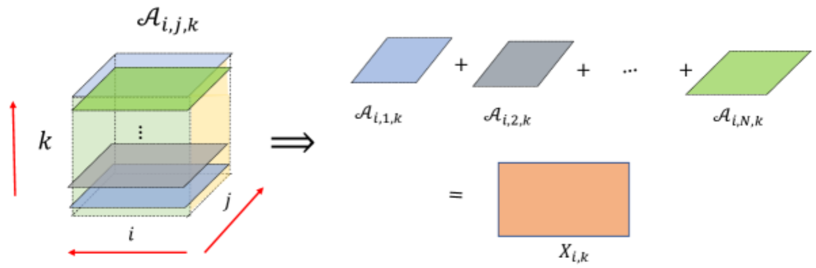


Figure 7: Their Tensor Representation of the Attention Tensor in the Supplement. **Discrepancy in diagram depicted and theorem proposed where summation is over second index,** j. Source [Ma et al. (2019)]

Find their NeurIPS reviews at https://proceedings.neurips.cc/paper/2019/file/dc960c46c38bd16e953d97cdeefdbc68-Reviews.html and the supplementary at https://ar5iv.labs.arxiv.org/html/1906.09777.

2.5 Discussion

Although the paper introduced a step in new direction for approximating attention mechanism in Transformers using tensor decomposition, **the reported theorems and figures were flawed, raising uncertainty regarding the reported implementations and experimental results in language modeling tasks.**

3 Conclusion

In this project, we first looked at the state-of-the-art matrix factorization based approximations to attention mechanism in Transformers, especially modeling the problem as a softmax kernel approximation using random features. While the theory for these matrix-factorization based techniques is quite well-established and rigorously researched, we explored potential use case of tensor decomposition applied to approximate the SOFTMAX kernel. In doing so, we came across a paper that proposed Block Tensor Decomposition based approximation of Single-Head and Multi-Head Attention Modules. By delving deeper into the theoretical results of the paper, we were able to **experimentally disprove claims** proposed in the paper [Ma et al. (2019)].

References

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers, 2022. URL https://arxiv.org/abs/2009.14794.

Xindian Ma, Peng Zhang, Shuai Zhang, Nan Duan, Yuexian Hou, Dawei Song, and Ming Zhou. A tensorized transformer for language modeling, 2019. URL https://arxiv.org/abs/1906.09777.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A. Smith, and Lingpeng Kong. Random feature attention, 2021. URL https://arxiv.org/abs/2103.02143.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. Platt, D. Koller, Y. Singer, and S. Roweis (eds.), *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.