Skoltech

Final Project of Numerical Linear Algebra course

Comparison of dimension reduction using Kronecker Product Approximation and Shapeshifter

Shapeshifters Team:
Bogdan Aleksandrov
Vasily Tesalin
Vasiliy Viskov
Oleg Shepelin
Anastasia Grigoreva

Problem Statement

Finding the most effective way of low dimensional representation of high dimensional data.

Importance/Relevance

For example natural language models involve large number of parameters. A single encoder-decoder Transformer in its base variant has about **44 million parameters**, not counting the word embedding matrix, which adds another **10 million or more**, depending on the chosen vocabulary or tokenization scheme. This explosion in the model size has led to increased interest in approaches for reducing the number of parameters in the model.

Using low-dimensional representations eases the learning task for machine learning algorithms without sacrificing too much model performance.

Available Solutions

Principal Component Analysis (PCA) - a popular technique for analyzing large datasets containing a high number of dimensions/features per observation, increasing the interpretability of data while preserving the maximum amount of information, and enabling the visualization of multidimensional data.

DeLighT (DEEP AND LIGHT-WEIGHT TRANSFORMER) -
DeLighT transformation maps input vector into a high dimensional space (expansion) and then reduces it down to a do dimensional output vector (reduction). During these expansion and reduction phases, DeLighT transformation uses group linear transformations (GLTs). To learn global representations, the DeLighT transformation shares information between different groups in the group linear transformation using feature shuffling.

Parameterized Hypercomplex Multiplication (PHM)
Generalize quaternions arithmetic to parametrise Fully-connected layers and use sum of r Kronecker products. Each Kronecker product involves a small $r \times r$ matrix and a larger $n/r \times m/r$ matrix, which together have $r^3 + nm/r$ trainable parameters.

Kronecker Product Approximation. Theory

$$\|A - \sum_{k=1}^{r} A_k \otimes B_k\| \to \min$$

Kronecker Product Approximation. Theory

If $B \in \mathbb{R}^{m_1 \times n_1}$ and $C \in \mathbb{R}^{m_2 \times n_2}$, then their *Kronecker product* $B \otimes C$ is an $m_1 \times n_1$ block matrix whose (i, j) block is the $m_2 \times n_2$ matrix $b_{ij}C$. Thus,

$$\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \otimes \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} = \left[\begin{array}{ccc|ccc} b_{11}c_{11} & b_{11}c_{12} & b_{11}c_{13} & b_{12}c_{11} & b_{12}c_{12} & b_{12}c_{13} \\ b_{11}c_{21} & b_{11}c_{22} & b_{11}c_{23} & b_{12}c_{21} & b_{12}c_{22} & b_{12}c_{23} \\ b_{11}c_{31} & b_{11}c_{32} & b_{11}c_{33} & b_{12}c_{31} & b_{12}c_{32} & b_{12}c_{33} \\ \hline b_{21}c_{11} & b_{21}c_{12} & b_{21}c_{13} & b_{22}c_{11} & b_{22}c_{12} & b_{22}c_{13} \\ b_{21}c_{21} & b_{21}c_{22} & b_{21}c_{23} & b_{22}c_{21} & b_{22}c_{22} & b_{22}c_{23} \\ b_{21}c_{31} & b_{21}c_{32} & b_{21}c_{33} & b_{22}c_{31} & b_{22}c_{32} & b_{22}c_{33} \end{array}\right].$$

Kronecker Product Approximation. Theory

$$\phi(B,C) = ||A - B \otimes C||_F$$

$$\phi(B,C) = \left\| \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ \hline a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \\ \hline a_{51} & a_{52} & a_{53} & a_{54} \\ a_{61} & a_{62} & a_{63} & a_{64} \end{array}\right] - \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} \otimes \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \right\|_F$$

$$= \left\| \begin{bmatrix} a_{11} & a_{21} & a_{12} & a_{22} \\ a_{31} & a_{41} & a_{32} & a_{42} \\ a_{51} & a_{61} & a_{52} & a_{62} \\ a_{13} & a_{23} & a_{14} & a_{24} \\ a_{33} & a_{43} & a_{34} & a_{44} \\ a_{53} & a_{63} & a_{54} & a_{64} \end{bmatrix} - \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \\ b_{12} \\ b_{22} \\ b_{32} \end{bmatrix} \begin{bmatrix} c_{11} & c_{21} & c_{12} & c_{22} \end{bmatrix} \right\|_F .$$

$$\mathscr{R}(A) = \begin{bmatrix} \mathrm{vec}(A_{11})^{\mathrm{T}} \\ \mathrm{vec}(A_{21})^{\mathrm{T}} \\ \mathrm{vec}(A_{31})^{\mathrm{T}} \\ \mathrm{vec}(A_{12})^{\mathrm{T}} \\ \mathrm{vec}(A_{22})^{\mathrm{T}} \\ \mathrm{vec}(A_{32})^{\mathrm{T}} \end{bmatrix}$$

$$\phi(B,C) = ||\mathscr{R}(A) - \mathrm{vec}(B)\mathrm{vec}(C)^{\mathrm{T}}||_F$$

Kronecker Product Approximation. Theory

$$\phi(B, C) = ||\mathscr{R}(A) - \text{vec}(B)\text{vec}(C)^{\text{T}}||_F$$

minimizing $\phi \Leftrightarrow$ finding nearest rank-1 matrix to $\mathscr{R}(A)$
solution - SVD : $\mathscr{R}(A) = U\Sigma V^T$

$$vec(B_{opt}) = \sqrt{\sigma_1}U(:,1), \quad vec(C_{opt}) = \sqrt{\sigma_1}V(:,1)$$

if $\mathscr{R}(A)$ has rank $\tilde{r}$: $\quad A = \sum_{k=1}^{\tilde{r}} \sigma_k U_k \otimes V_k$ (KP SVD).

$$A_r = \sum_{k=1}^{r} \sigma_k U_k \otimes V_k \quad r \leqslant \tilde{r}$$

is the closest matrix to A that is the sum of r Kronecker products

Kronecker Product Approximation. Advantages

- Matvec costs $O((m+n)r)$ flops where $m = m1*m2$, $n = n1*n2$ and r - rank
- Memory usage is $O(r(m1*n1 + m2*n2))$ ($O(r\sqrt{mn})$ when $n1 = n2$, $m1 = m2$)

Kronecker Product Approximation. Theory

Consider two vectors, $x_A, x_B \in \mathbb{R}^m$. Kronecker product of $x = x_A \otimes x_B$ of these two vectors is a vector $x \in \mathbb{R}^n$ with entries defined through products

$$x[k] = x_A[\kappa]x_B[\lambda] \quad \text{for} \quad k = (\kappa - 1)m + \lambda,$$
$$x[q] = x_A[\pi]x_B[\rho] \quad \text{for} \quad q = (\pi - 1)m + \rho$$

for $\kappa, \lambda, \pi, \rho \in [m]$.

A matrix $A \otimes B$ acts on vectors $x = x_A \otimes x_B$ as $(A \otimes B)x = (Ax_A) \otimes (Bx_B)$.

Shapeshifter. Theory

Suppose we have matrix U that acts only on 2 dimensions {k, q}. Then we can make it's tensor decomposition: $U = A_1 \otimes B_1 + A_2 \otimes B_2$

1) Act on {p, q}:

   remember: $(A \otimes B)x = (Ax_A) \otimes (Bx_B)$

Let's construct B that would act on these 2 dimensions.

Consider special case for p, q: $\kappa = \pi = \nu$

That is:

$$x[k] = x_A[\underline{\nu}]x_B[\lambda] \quad \text{for } k = (\underline{\nu} - 1)m + \lambda$$
$$x[q] = x_A[\underline{\nu}]x_B[\rho] \quad \text{for } q = (\underline{\nu} - 1)m + \rho$$

Shapeshifter. Theory

We need B that acts on lambda and rho the same way U acts on p and q: B acts on λ, ρ in the same way as U acts on k, q: $B[\{\lambda, \rho\}] = U[\{k, q\}]$

$$\begin{bmatrix} 0 & & \\ & 1 & \\ & & 0 \end{bmatrix} \otimes \begin{bmatrix} \alpha & & -\beta \\ & 1 & \\ \beta & & \alpha \end{bmatrix} = \begin{bmatrix} 0 & & & & & \\ & \ddots & & & & \\ & & \alpha & & -\beta & \\ & & & 1 & & \\ & & \beta & & \alpha & \\ & & & & & \ddots \\ & & & & & & 0 \end{bmatrix}.$$

$$U^{\{k,q\}} = \mathbb{1}_\nu \otimes B^{\{\lambda,\rho\}} + (I - \mathbb{1}_\nu) \otimes I$$

Shapeshifter. Theory

Orthogonal transition to Perfect case: $\kappa \neq \pi$ and $\lambda \neq \rho$

But what if p, k are such that: $\kappa \neq \pi$ and $\lambda \neq \rho$. Then we would transform the problem to perfect case:

$$U^{\{k,q\}} = V^T \left(\mathbb{1}_\pi \otimes B^{\{\lambda,\rho\}} + (I - \mathbb{1}_\nu) \otimes I \right) V$$

V need to map κ into π and have tensor decomposition.

Set V as: $V = P_{\kappa \to \pi} \otimes \mathbb{1}_\lambda + I \otimes (I - \mathbb{1}_\lambda)$

Then we can see how V acts on x: $x' = Vx = P_{\kappa \to \pi} x_A \otimes \mathbb{1}_\lambda x_B + I x_A \otimes (I - \mathbb{1}_\lambda) x_B$

This V rotates changes dimension κ into π, thus changing task to perfect case.

Shapeshifter. Theory

We decomposed 2-variant matrix U into: $U = \prod_{i=1}^{3n^2/2} \sum_{j=1}^{2} A_{ij} \otimes B_{ij}$

By moving one pair of dimensions at a time we can decompose any matrix of size n into:

Moreover, we can decompose matrices into *r*-variant matrices

Example for *r* = 3: Perfect case (κ = π = τ): $U^{\{k,q,s\}} = \mathbb{1}_\nu \otimes B^{\{\lambda,\rho,\upsilon\}} + (I - \mathbb{1}_\nu) \otimes I$

Projection to perfect case:

$$x' = Vx = P_{\kappa\to\tau} x_A \otimes \mathbb{1}_\lambda x_B + P_{\pi\to\tau} x_A \otimes \mathbb{1}_\rho x_B + I x_A \otimes (I - \mathbb{1}_\lambda - \mathbb{1}_\rho) x_B$$

Shapeshifter. Theory

We can represent any square orthogonal matrix by:

$$U = \prod_{i=1}^{3} \sum_{j=1}^{r} Ai, j \otimes B_{i,j}$$

We would need at most $L = \mathcal{O}(3n^2/r)$ layers, each involving a Kronecker product of rank at most *r*.

But what about non-square case?

Shapeshifter. Theory

Reshaping to square:

Square matrix multiplications are efficient in the sense that all elements of a column of A are multiplied with each element of a row of B.

First step of Shapeshifter is to reshape W (n x m) into ($\sqrt{mn}$ x $\sqrt{mn}$)

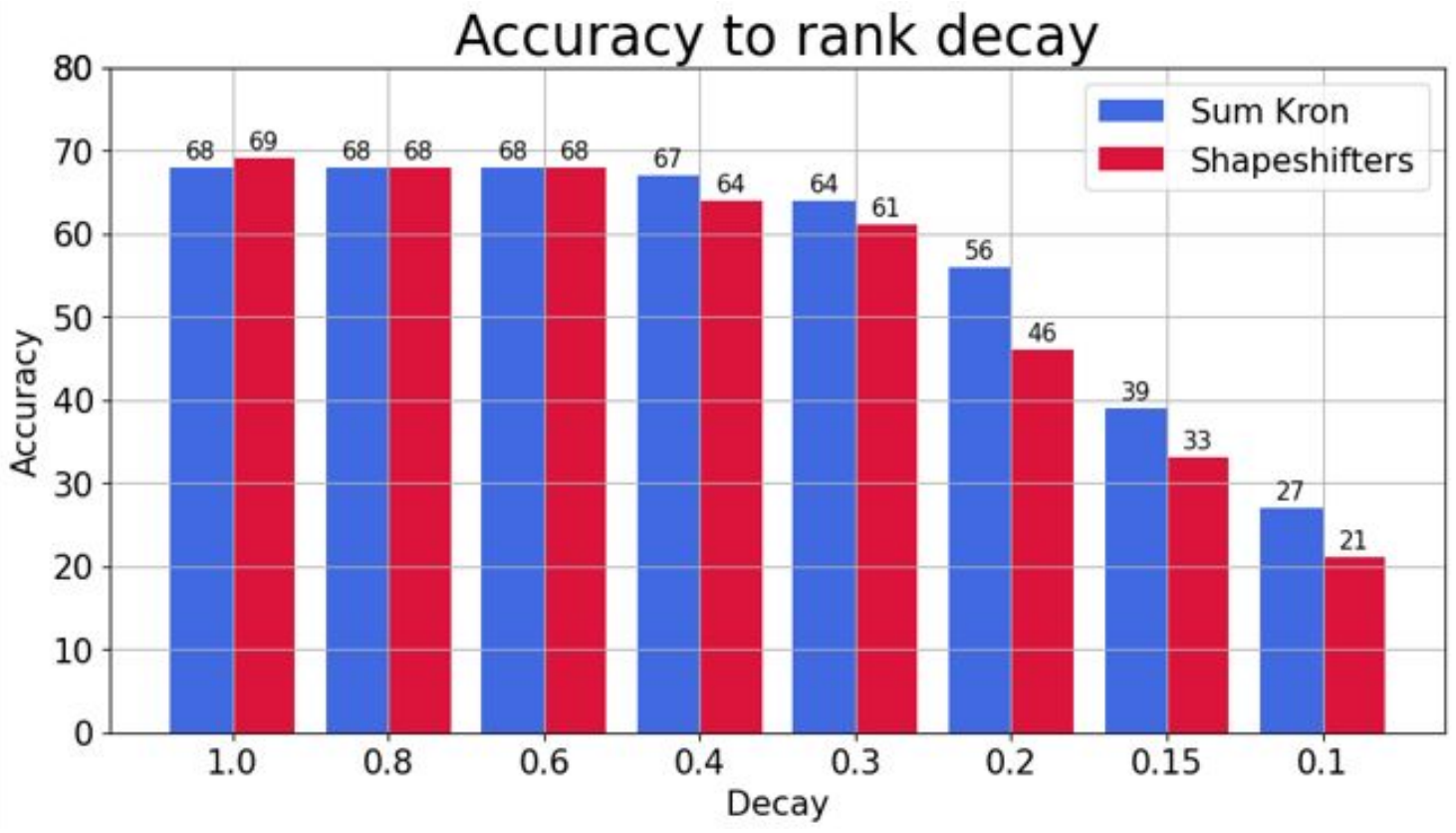
Then we use efficient multiplications:

$$\left(\sqrt{nm} \times r\right)\left(r \times \sqrt{nm}\right) \xrightarrow{\text{multiply}} \sqrt{nm} \times \sqrt{nm} \xrightarrow{\text{reshape}} \sqrt{n} \times \sqrt{m} \times \sqrt{n} \times \sqrt{m}$$
$$\xrightarrow{\text{transpose}} \sqrt{n} \times \sqrt{n} \times \sqrt{m} \times \sqrt{m} \xrightarrow{\text{reshape}} n \times m$$

This representation only costs $2r\sqrt{mn}$ parameters instead of mn.

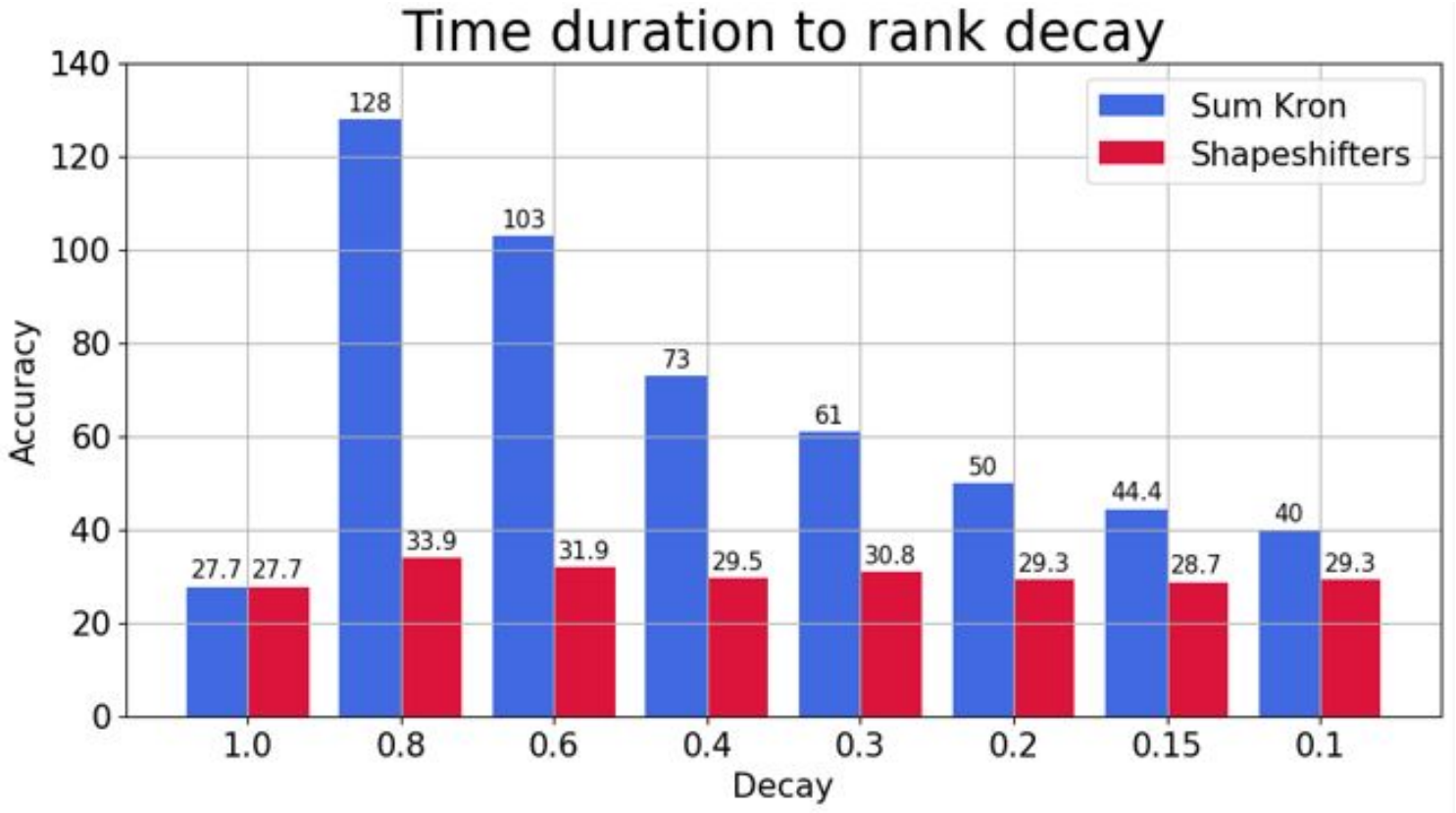
Skoltech

Dependence of accuracy on the number of terms for the two approaches



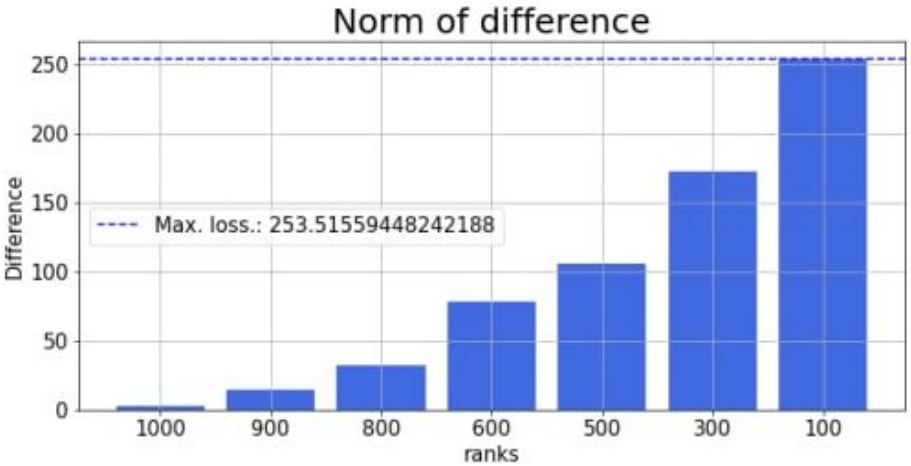
Skoltech

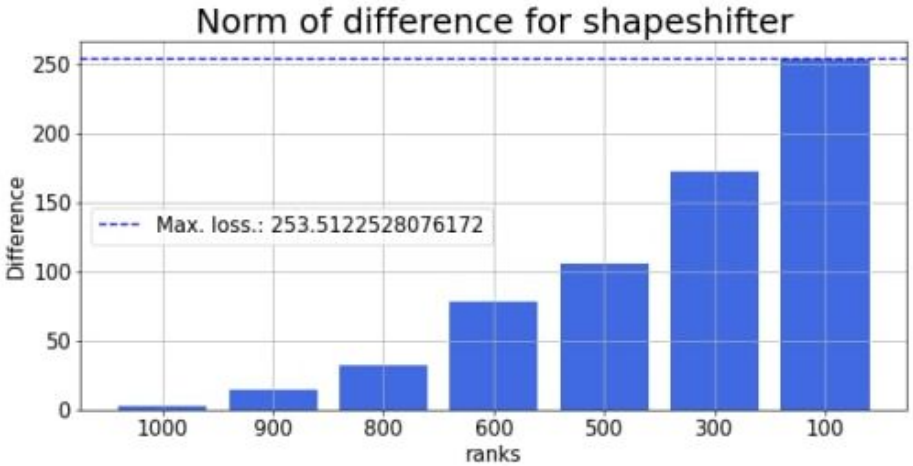
Dependence of execution time on the number of terms for the two approaches



Skoltech

Dependence of norm of difference on rank





matrix size 1050x1050

Thanks for attention!

GitHub