Exploring Low-Rank and Sparse Approximations for Transformers

Ashwin De Silva
Biomedical Engineering
Johns Hopkins University
Baltimore, MD, USA
ldesilv2@jhu.edu

Esther Whang
Biomedical Engineering
Johns Hopkins University
Baltimore, MD, USA
ewhang2@jh.edu

***Abstract*—Transformers have become the workhorse of modern deep learning models, enabling advances in natural language processing and computer vision. However, the computation burden of these models - mainly due to their large size - restricts their use in resource-constrained environments. Recent work attempts to tackle the large size of transformer-based models by applying model compression techniques used on neural networks. This work examines the unique challenges of compressing transformers, specifically with low-rank and sparse approximations. We explore ways to apply these assumptions to components of the attention mechanism and Furthermore, we look beyond the simple singular value decomposition for low-rank approximation and examine the effect of including assumptions about sparsity using algorithms like robust PCA and GreGoDec. Our code is available at https://github.com/Laknath1996/cs-project.**

I. Introduction

The transformer architecuture has been instrumental in advancing the state-of-the-art in natural language processing and computer vision. Much of its success is due to the unique structure of the attention mechanism, which allows transformers to model much longer ranges of dependencies in a given input sequences compared to its predecessors. However, despite their many capabilities, transformer-based models have disadvantages that limit their implementation. Mainly, they impose a heavy computational burden due to their large number of parameters, making them difficult to utilize in resource-constrained environments. This conflict between performance and practicality motivates study into methods of reducing the number of parameters in transformer-based models while maintaining their outstanding performance.

One approach to reduce this computational burden is model compression, where components of the model are removed or approximated. The underlying assumption of model compression is that these large models are over-parameterized and therefore able to withstand the removal of redundant aspects. This is done by either removing components of the model (pruning), reducing the precision of the representation of the model (quantization), or by find a low-rank approximation of the parameters [1]. Of these three major approaches, the low-ranked approximation of transformers surprisingly seems understudied. While there has been past work supporting the idea of low intrinsic dimesionality of pre-trained transformer-based large language models, attempts to apply this idea has had mixed results. Efforts to find low-rank approximations of transformer components, specifically of the weight matrices, typically require additional steps beyond simple low-rank approximation to recover the original performance such as altering the low-rank decomposition algorithm or retraining the resulting compressed model to restore its original function [2], [3]. This raises questions about whether these models truly have a low intrinsic dimensionality, and if so, the best methods to take advantage of this property.

This project seeks to study the specific nature of low-rankedness in transformer-based models. We first observe the properties of the most distinctive aspect of the transformer, the attention mechanism, in the single-headed and multi-headed cases. We expand on prior work on methods for approximating the weight matrices by going beyond the standard approaches for low-rank decomposition - mainly, the truncated Singular Value Decomposition - and implementing alternative algorithms that incorporate both low-rank and sparsity assumptions. Our study of the attention probabilities and sentence embeddings from these different approaches provide insight on compression for transformer-based models.

II. Related Works

Prior work on pre-trained large language models, one of the most prominent examples of transformer-based models, suggests that these models have a low intrinsic dimensionality [4]. This finding has been useful in inspiring methods for reducing the computational burden of transformer-based models, mainly as the theorectical basis for LoRA [5]. LoRA applies a low-rank appoximation to the changes of the weights during finetuning, which reduces the number of parameters required during training.

Despite the success of LoRA, other attempts to apply low-rank assumptions to transformers have had mixed outcomes. Given the need to decompose a matrix, one simple approach would be to utilize the truncated Singular Value Decomposition on the weights of the transformer. A standard technique for finding matrix low-rank approximation, truncated SVD is one of the most straightforward way to obtain a solution for the eq. (5). Given a matrix $W \in \mathbb{R}^{d \times d}$, first we perform SVD on it to obtain $\text{SVD}(W) = D\Sigma R$. Next, we retain the r largest singular values to obtain the $U = D_r\Sigma_r \in \mathbb{R}^{d \times r}$ and

$V = R_r \in \mathbb{R}^{r\times d}$. Then, the r-rank approximation L is given by,

$$L = UV = (D_r\Sigma_r)R_r.$$

This method is easy to implement, and it provides a bilateral factorization $L = UV$. This is advantageous in terms of compression and parameter efficiency as it allows us to represent the low-rank approximation with just $2dr$ parameters, instead of d^2.

Unfortunately, despite its convenience, the truncated SVD seems unideal for compressing transformer-based models. In past works that utilizes the truncated SVD as the basis of their model compression, a blind application of SVD on the weight matrices resulted in significant drops in performance, which forces the additional steps such as feature distillation in order to recover the capabilities of the original model [3]. While the SVD may provide structural similarity to the original weight matrices, it seems unable to maintain the functional equivalence. To avoid performance loss, the SVD has be changed to take into account additional factors such as the importance of certain parameters [2] or the distribution of the activations [6]. All in all, the difficulty of applying SVD suggests that we cannot simply assume that the general weights of a transformer are low-ranked.

One possible reason for this difficulty could be the variable rankedness between layers. [6] notes a higher resiliency to compression in the weights of the multi-headed attention (MHA) layers compared to other layers. [7] supports this observation that the MHA layers exhibit low-rank structure. Based on these observed properties, focusing on the multi-headed attention (MHA) layers appears to be a more educated approach when examining low-ranked approximation for transformer compression.

III. Background

A. The Attention Operation

The key component of the transformer architecture is the scalar dot-product attention ("attention" in short) operation it computes. Given an n-length and d-dimensional sequence $X \in \mathbb{R}^{n\times d}$, and the key, value, and query weight matrices $W_K, W_V, W_Q \in \mathbb{R}^{d\times d}$, the attention operation is given by,

$$\text{Attention}(X) = \text{Softmax}\left(\frac{XW_Q(XW_K)^\top}{\sqrt{d}}\right)XW_V \quad (1)$$

This is also known as the single-head attention. A slightly more involved variant of this is the multi-head attention (MHA), which is widely used in virtually all the state-of-the-art transformers. The multi-head attention is computed as follows.

$$\begin{aligned}\text{MHA}(X) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W_O \quad (2)\\ \text{head}_i &= \text{Attention}(XW_Q^i, XW_K^i, XW_V^i)\end{aligned}$$

Here, $W_Q^i, W_K^i, W_V^i \in \mathbb{R}^{d\times d/h}$ and $W_O \in \mathbb{R}_{d\times d}$ where h is the number of heads. When implementing MHA, we consider only three $d \times d$ weight matrices W_K, W_Q and W_V, and let W_*^i be the submatrix of W_* with columns ranging from id/h to $(i+1)d/h$, where $* = \{K, Q, V\}$.

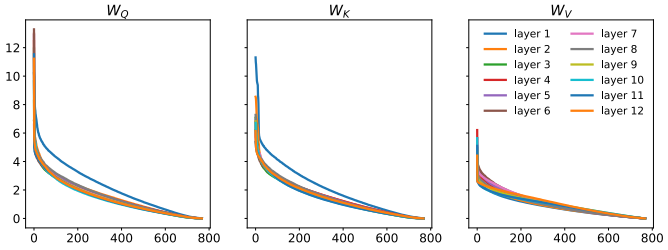

Fig. 1. Spectrums of the weight matrices of each pre-trained RoBERTa layer

While typical transformers include numerous fully connected feed forward layers and embedding layers, in this work, we focus solely on the attention layers and explore ways of introducing low-rank and sparse approximations to the corresponding weight matrices.

B. RoBERTa Architecture

Throughtout this work, we use the RoBERTa architecture [8] as a testbed for our analysis and experiments. It comprises of a series of $N = 12$ hidden layers each equiped with multi-head attention with 12 heads. The model dimension d is set at 768. The model is pre-trained on a large natural language datasets using masked language modeling (MLM) and next sentence prediction (NSP) objectives. For downstream tasks such as sentiment classification, we replace the pooling layer at the end of the pre-trained RoBERTa model with a trainable multi-layer perceptron (MLP) and finetune using methods such as LoRA.

C. Initial Observations

1) Pre-trained weight matrices are not low-rank: To initiate our investigation, we first considered taking a closer look at spectrums of the the pre-trained MHA weight matrices in each layer of the RoBERTa model. Observing these spectrums (see fig. 1), we notice that the pre-trained W_K, W_V, W_Q matrices are not inherently low-rank. Therefore, a purely low-rank approximation would result in a non-trivial information loss in the weight matrices, that will degrade the performance of the approximated model.

2) A redundancy within single-head attention: When we rewrite eq. (1), we obtain the following expression for the single-head attention operation.

$$\text{Attention}(X) = \text{Softmax}\left(\frac{XW_QW_K^\top X^\top}{\sqrt{d}}\right)XW_V \quad (3)$$

Therefore, if $B = W_QW_K^\top$, then the attention probabilities $\text{AP}(X)$ are given by

$$\text{AP}(X) = \text{Softmax}\left(\frac{XBX^\top}{\sqrt{d}}\right). \quad (4)$$

The matrix B summarizes the W_K and W_Q matrices into a single matrix. Interestingly, upon observing the spectrum of the B matrices of each pre-trained RoBERTa layers (see fig. 2),

we observed that they are highly low-rank when compared to the individual W_K and W_Q matrices. However, the low-rank nature of B cannot be directly exploited to approximate multi-head attention operation, as it involves partitioning the weight matrices into different heads prior to computing attention probabilities for each head. Therefore, in the case of mulit-head attention, we focus on introducing low-rank and sparse approximations to W_K and W_Q weight matrices individually. On the other hand, single-head attention can benefit from the low-rankedness of B, and we exploit it to provide low-rank approximations. We consider that the approximation is appropriate if the approximate attention probabilities do not deviate significantly from the original attention probabilities.

IV. Proposed Study

We first discuss several popular low-rank and sparse approximation strategies below. High-dimensional matrices generally tend to have a lot of redundancy in terms of the information it contains. This redundancy is associated with the low-rank structure of the matrix. In other words, all the instances live on a subspace spanned by a few number of bases. If a matrix W has a low-rank structure, then it can be written as the sum of a few rank-1 matrices such that $W = \sum_{i=1}^{r} U_i V_i$ where U_i is a column vector and V_i is a row vector. The problem of finding a low-rank approximation $L = UV$ for W can be formally written as,

$$\begin{aligned} &\min_{L} \|W - UV\|_F^2 \\ &\text{s.t. } \text{rank}(U) = \text{rank}(V) \leq r \end{aligned} \tag{5}$$

where, r is the target rank.

A. Robust PCA

A pure low-rank approximation to W may not be able to capture it complex structure. Therefore, we adopt a more general assumption by letting $W = L + S$, where we consider that W can be decomposed into the sum of a low-rank matrix L and a sparse matrix S. L describes the instances that lie on a common and shared subspace, while S represents the spiky anomalies that are rarely shared. The problem of finding L and S can be formally stated as follows.

$$\begin{aligned} &\min_{L,S} \text{rank}(L) + \lambda \|S\|_0 \\ &\text{s.t. } W = L + S \end{aligned} \tag{6}$$

A convex relaxation of this problem is given by,

$$\begin{aligned} &\min_{L,S} \|L\|_* + \lambda \|S\|_1 \\ &\text{s.t. } W = L + S, \end{aligned} \tag{7}$$

where, $\|\cdot\|_*$ is the nuclear (trace) norm. This is known as robust PCA, and it can be solved using popular optimization algorithms such as augmented Lagrangian multiplier.

While $\hat{W} = L + S$ yields a better approximation via robust PCA, a notable drawback is the requirement of $d^2 + \|S\|_0$ parameters to represent $\hat{W}$, because L is not bilaterally factorized into UV. We also do not get to control to the level of compression, as we cannot set a target rank r when solving the robust PCA problem. Moreover, optimizing eq. (7) requires running a full SVD step each iteration which results in a high computational budget.

B. Greedy Bilateral Smoothing

As expressed in the section above, we prefer the model specified by $W = UV + S$ when attempting to compress weight matrices. We can take a step further and relax the exact equality to have $\|W - UV - S\|_F^2 \leq \epsilon$ where ϵ controls the level of dense noise allowed. This leads to a noisy robust PCA problem which we can formally write as follows.

$$\begin{aligned} &\min_{U,V,S} \|W - UV - S\|_F^2 + \lambda \|S\|_1 \\ &\text{s.t. } \text{rank}(U) = \text{rank}(V) \leq r \end{aligned} \tag{8}$$

This formulation satisfies two of our desired requirements: (1) having a bilateral factorization UV for the low-rank matrix, and (2) accepting a target rank r. The GreGoDec [9], [10] algorithm solves eq. (8) in a greedy fashion without the need to have repeating SVD steps each iteration. Therefore, it provides a faster way of computing a $UV + S$ approximation to W. This algorithm has been successfully used to compress deep convolutional neural networks, which further motivates the interest to employ it in compressing the attention layers of the transformer models.

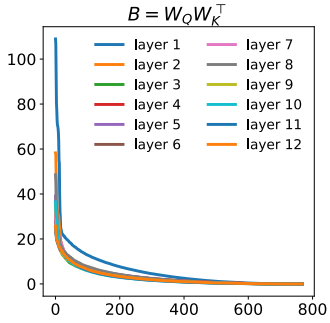

Fig. 2. Spectrums of the B matrices of each pre-trained RoBERTa layer

V. Analysis and Results

Following our intuition on the redundancy within single-head attention, we first experiment with compressing the matrix B from eq. (3). For multi-head attention, since the involvement of the matrix B is less obvious, we experiment with compressing the W_Q and W_K matrices separately instead. Finally, we compare the original and approximated models and attention operations, to understand the trade-off between performance and compression. Throughout our analysis, we predominantly use truncated SVD and GreGoDec, as they yield bilaterally factorized low-rank matrices. The folowing two sections describe the specific details associated with our experiments involving single- and multi-head attention models.

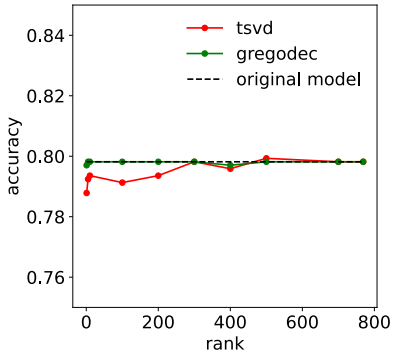

Fig. 3. Target rank (**x-axis**) of the low-rank approximation of B vs. the model accuracy on the test set (**y-axis**)

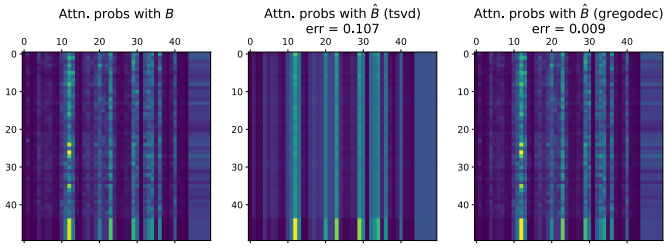

Fig. 4. Attention probabilities computed by the original (**left**), tsvd-approximated (**middle**), and GreGoDec-approximated (**right**) models for the sentence: "*there's enough melodrama in this magnolia primavera to make pta proud yet director muccino's characters are less worthy of puccini than they are of daytime television.*"

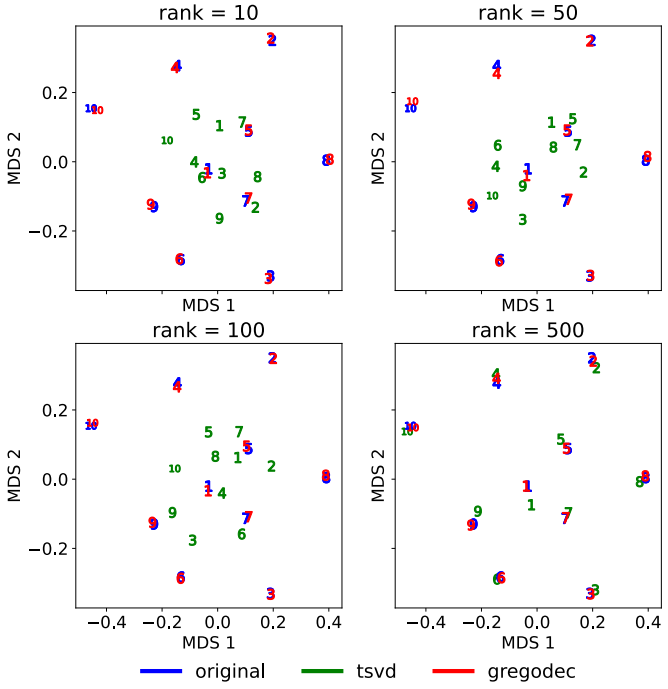

Fig. 5. Visualizations of the sentence embeddings produced by original (**blue**), tsvd-approximated (**green**), and GreGoDec-approximated (**red**) models.

A. Single-Head Attention

To obtain a trained model, we reconfigured a randomly initialized RoBERTa model with 6 hidden layers with single-head attention. The output from the last hidden layers was fed into a multi-layer perceptron with 1 hidden layer to perform binary classification. We trained this model from scratch on the sentiment classification dataset SST-2 from the GLUE Benchmark, which comprises of 67.3k train samples and 872 test samples. The trained model achieved an accuracy of 0.798 on the test set.

Next, we extracted the W_K and W_Q matrices from each layer of the trained model, computed the matrices $B = W_Q W_K^\top$, and used truncated SVD and GreGoDec to find the corresponding approximations $\hat{B}$. We then evaluated the model accuracy after plugging in $\hat{B}$ back to the model in place of B. We report these accuracy values in fig. 3. There, we observe that sparse and low-rank approximation yielded by GreGoDec does not result in a drop in performance with respect to that of the original model. Even utilizing rank-1 approximation via truncated SVD does not result in a significant loss of model performance. Therefore, we conclude that the low-rank nature of the B matrices can be exploited to achieve high compression rates in the attention layers without sacrificing the accuracy of the model.

In addition, for several selected setences from the test set, we compared the attention probabilities computed by the first layer of the original model against those computed by the approximated models via truncated SVD and GreGoDec. One such comparison is illustrated in fig. 4. It can be noticed that while the attention probabilities computed by the tsvd-approximated model retains the global structure of its original counterpart, but fails to retain the local, more refined details. Such local details are captured by the gregodec-approximated model, largely owing to the sparse component. This further demonstrate the utility of employing low-rank and sparse approximations to the matrix B.

B. Multi-Head Attention

For our experiments with the multi-head attention models, we worked with the pre-trained RoBERTa model. We approximated the W_K and W_Q matrices of each layer using truncated SVD and GreGoDec to obtain approximated models. To compare the models with each other, we randomly sampled 10 sentences from SST-2 test set and embedded them in the 768-dimensional latent space of the RoBERTa model. Thereafter, we further reduced the dimensionality of the embeddings to visualize them on a 2D grid. Multi-dimensional scaling was used for this purpose in order to preserve the intra-embedding distances in the visualizing space. We report 4 such visualization in fig. 5. We observe that GreGoDec-approximated model yields similar sentence embeddings to that of the original model, even at extremely low-target ranks. On the other hand, tsvd-approximated model produces sentence embeddings closer to that of the original only when the target rank is significantly large. This indicates that low-rank approximation alone is not sufficient to achieve a approxima-

tion that preserves the model performance.

VI. Discussion and Conclusions

In this work, we demonstrated ways of compressing single-head attention layers by exploiting a redundancy in them, which results in a low-rank matrix. For multi-head attention models, we explored and demonstrated the utility of using low-rank and sparse approximations instead of using solely the former component.

The key to a good compression strategy of weight matrices in a neural network is to make sure that the original and the approximated operations have an functional equivalence. We have not explore this in our work, but one possible way to do this is to force the approximation strategy to preserve the attention probabilities. This can be written formally as,

$$\min_{U,V,S} \frac{1}{2m} \|\text{AP}(X,B) - \text{AP}(X,UV+S)\|_F^2 + \lambda\|S\|_1$$
$$\text{s.t. } \frac{1}{2}\|B - UV - S\|_F^2 \leq \epsilon,\ \text{rank}(UV) \leq r,$$

where $\text{AP}(X,B) = \text{Softmax}(XBX^\top/\sqrt{d})$ and $X = [X_1,\ldots,X_m]$ are inputs used to evaluate functional equivalence component of the objective. Similar objectives have been used to successfully compress convolutional neural networks [11] via GreGoDec. Therefore, it may be of interest to adapt GreGoDec to solve the problem stated above.

Moreover, low-rank/sparse approximated pre-trained models can be combined with techniques such as LoRA to achieve a better degree of parameter efficient finetuning.

References

[1] X. Zhu, J. Li, Y. Liu, C. Ma, and W. Wang, "A survey on model compression for large language models," 2023.

[2] Y.-C. Hsu, T. Hua, S. Chang, Q. Lou, Y. Shen, and H. Jin, "Language model compression with weighted low-rank factorization," 2022.

[3] M. Ben Noach and Y. Goldberg, "Compressing pre-trained language models by matrix decomposition," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, K.-F. Wong, K. Knight, and H. Wu, Eds. Suzhou, China: Association for Computational Linguistics, Dec. 2020, pp. 884–889. [Online]. Available: https://aclanthology.org/2020.aacl-main.88

[4] A. Aghajanyan, L. Zettlemoyer, and S. Gupta, "Intrinsic dimensionality explains the effectiveness of language model fine-tuning," 2020.

[5] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021.

[6] Z. Yuan, Y. Shang, Y. Song, Q. Wu, Y. Yan, and G. Sun, "Asvd: Activation-aware singular value decomposition for compressing large language models," 2023.

[7] G. Li, Y. Tang, and W. Zhang, "Lorap: Transformer sub-layers deserve differentiated structured compression for large language models," 2024.

[8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019.

[9] T. Zhou and D. Tao, "Greedy bilateral sketch, completion & smoothing," in *International Conference on Artificial Intelligence and Statistics*, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:14236992

[10] ——, "Godec: Randomized lowrank & sparse matrix decomposition in noisy case," in *International Conference on Machine Learning*, 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:1387290

[11] X. Yu, T. Liu, X. Wang, and D. Tao, "On compressing deep models by low rank and sparse decomposition," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 67–76, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:24553488

VII. Individual Contributions

Both members contributed equally to this work.

Ashwin De Silva: 

Esther Whang: