Exploration of LAyer-SElective Rank reduction

Maya Ha Gustavo Carvalho Gurshaan Lobana Ogheneyoma Akoni

Abstract

Recently "The Truth is in There: Improving Reasoning in Language Models with Layer-Selective Rank Reduction" (Sharma et al., 2023) demonstrated how the performance of a Large Language Model (LLM) could be improved simply by factoring a layer of the neural network with Singular Value Decomposition (SVD) and removing the rows with the smallest σ values. Our goal with this project was to attempt to expand the paper by (1) investigating the effects of doing this SVD-based ablation on the LLM's output and (2) explore the possibility of using this as a pruning method for LLMs.

1. Introduction

LAyer-SElective-Rank-reduction (LASER) is a simple intervention that selectively removes the higher-order components of a learned weight matrix in a specific layer using singular value decomposition. This intervention is done after training, and it doesn't require any additional data. The paper found that LASER can improve the performance on certain tasks by 20-30%.

As interesting as the paper is, there were two missed opportunities which we felt would be interesting to tackle for our project. First, the paper does not expand in detail on *how the LASER intervention affects the LLMs output*. They posit that the intervention functions as a "de-noising" mechanism, but only conduct a single experiment on one of the several datasets used in the paper to support this claim. For this project, we conducted analysis on three distinct datasets to, not only verify their hypothesis, but also expand their conclusions by answering *how does this "de-noising" manifest in the behaviour of the LLM in distinct tasks*.

Additionally, another direction we felt the paper didn't explore fully was *LASER's potential as a pruning method*. Pruning is a research area who's goal is to develop methods that can make neural networks more efficient by remove weights from the model without sacrificing too much performance. However, even modern pruning methods still cause the model to lose a couple of percentage points of performance after pruning on average (Sun et al., 2024).

Figure 1. Image above illustrates how LASER removes the higher-frequency components of the SVD matricies. The grey parts are removed.

What makes the LASER paper so interesting is that the model with the pruned layer gains performance. This makes LASER and other SVD-based ablations promising candidates for pruning methods. Unfortunately, the paper does not find any method to determine which layer of the model to apply LASER to, nor how much of the weights should be removed of the chosen layer. For our project, (1) we explore how much we are able to remove from layers that benefit from the LASER intervention without deprecating model performance and (2) we also attempt to find some metric that could inform of what layers would benefit from a LASER intervention.

2. Methods

2.1. LAyer-SElective Rank reduction

Here is a quick explanation on how the LASER intervention works. Given a weight matrix W_l which represents the l-th layer of the model, decompose the matrix using SVD such that $W_l = U_l \Sigma_l V_l^t$, where U_l are the left singular vectors, V_l^t are the right singular vectors, and Σ is the diagonal matrix of singular values. After doing the SVD, assuming that the singular values are ordered from biggest to smallest, remove the columns of U_l and the rows of V_l^t corresponding to the k smallest singular values, as well as the singular values from Σ themselves. Both l and k are hyperparameters. The LASER paper finds what l and k to use via brute force search. Figure 1 illustrates the LASER ablation.

2.2. Investigation on Effects

We use three separate datasets to investigate the effect of a LASER intervention, FEVER (Thorne et al., 2018), BigBench-WikidataQA (Srivastava et al., 2022), and Bios-Professions (De-Arteaga et al., 2019). We describe each of them in detail at the Results section. We use GPT-J for

all of our experiments. To explore the effect of LASER in GPT-J's outputs we analyse the outputs of GPT-J (1) without any intervention, (2) with the best l and k hyperparameters for a LASER intervention found by the original LASER paper, and (3) by using LASER to ablate the biggest singular values instead of the smallest (we call these the high-order components).

We analyze the outputs by doing TF-IDF counts and named-entity recognition. The idea was to use TF-IDF to find the most relevant terms present in the output samples. Named-entity recognition, on the other hand, was used specifically because we qualitatively noticed that there seemed to be less named-entities mentioned before the LASER intervention. By using spaCy, a named-entity recognition library, we could find a quantitative measure for this qualitative observation.

2.3. Exploration of Pruning Potential

To explore the potential of LASER as a pruning method we first search to see how small we can make the hyperparameter k given a favorable l before we see performance deprecation.

In terms of finding a metric that predicts good LASER intervention hyperparameters, our first intuition was to assume that the layers that benefited from LASER were rank deficient. However, the original LASER paper tests for this hypothesis and they find that not to be the case! Therefore, we had to explore alternative ideas for a metric.

Our first idea was to measure the nuclear norm of the all the subsets of the Σ matrix. The intuition behind this was that, as we considered more and more singular values, maybe we would be able to see some kind of elbow graph where it would be noticeable that most of the information was aggregated at some top k singular values.

Our second idea was to measure the rank of layers relative to previous layers. Perhaps the individual weight matrices were not rank deficient by themselves, however maybe they were rank deficient in relation the the layers prior to it. To test this, we would first zero-out the values of the weight matrix $W_l - 1$ and W_l that were below some threshold t. Then, we individually computed the rank for each of the matrices. Finally, we would concatenate $W_l - 1$ and W_l column-wise into a single matrix and then compute the rank of concatenated matrix. We could then compute the ration between the sum of their individual ranks and the rank when they were concatenated. We did this for every pair $l - 1$ and l until the last layer of the model. We also experimented with calculating this rank of shared subspace using the U_l or V_l^t matrices instead of W_l. Lastly, note that GPT-J is a transformer and hence has different modules with typical neural network weight layers. Due to time constraints and to simplify the problem, we only consider the MLP Output Weight layers. This will be important to interpret our results and know what part of Figure 2 is relevant to our experiments.

3. Results

3.1. Effects of LASER

3.1.1. FEVER

Example:

- Prompt: Tennessee is a state in the United States.
- True Answer: True
- Base Model: False
- LASER: True
- Higher-Order LASER: False

The FEVER dataset consists of claims and labels characterizing each as either true or false. We found that all of the predictions that were changed to the correct label by LASER were originally 'False' without it, and changed to 'True' with LASER. We also found that for all the samples, the higher-order component approximation predicted 12,046 'False' labels and only 1,040 'True' labels, which might indicate that removing the higher-order components allowed the model to correctly predict more 'True' labels. To further back this theory, of the 6,174 samples that the higher-order approximation predicted incorrectly, the base model predicted almost all of them as 'False.' It seems that the higher-order components are skewing the predictions of the base model towards 'False.'

We also investigated the top 5 predicted labels, and before LASER they include many words that are not 'True' or 'False', such as 'not', 'a', 'about', 'that', 'clearly', etc. After LASER, it is much more consistently predicting 'True' and 'False' for all top 5 labels. Before LASER, 37,771 of the top 5 predicted tokens were words besides 'True' or 'False'. After LASER, 14,137 of the top 5 predicted tokens were words besides 'True' or 'False.'

3.1.2. BIOS-PROFESSION

Example:

- Prompt: He earned his Ph.D. from Kent State University. His research focuses on: stress, coping, perfectionalism, mindfulness and the teaching of counseling. Dr. Moate previously taught at Heidelberg University and Indiana University of Pennsylvania.
- True Answer: professor
- Base Model: teacher
- LASER: professor
- Higher-Order LASER: teacher

The Bios-Profession dataset consists of descriptions of peo-

ple, and the task is to predict each person's occupation from eight labels. The higher-order approximation seems to provide answers that correlate more with specific words from the sentence description rather than utilizing those words within the given context of the sentence. For instance, if the word "teach" appears in the description, the higher order components are more likely to predict "teacher" as the profession. In contrast, the lower order components demonstrate a superior ability to consider the broader context of the sentence. They can recognize qualifications like a doctorate and activities such as research at a university, which may indicate a profession beyond the immediate association with the word "teach."

3.1.3. BIGBENCH-WIKIDATAQA

Example:

- Prompt: The capital of California is
- True Answer: sacramento
- Base Model: the capital of california is a city of many faces. it is a city
- LASER: the capital of california is sacramento. it is located in sacramento county. sacramento
- Higher-Order LASER: the capital of california is a city of many faces. it's

In this dataset, we observe that despite the presence of related data within the higher order component predictions, there's a noted lack of specificity in the predictions. The higher-order approximation predictions tend to be more verbose and rather than providing precise answers, these components tend to offer predictions that hover around the correct answer without pinpoint accuracy. This suggests that while they may capture some aspects of the underlying patterns or relationships within the data, they struggle to provide clear and specific predictions.

3.2. Analysis of BigBench-WikidataQA output

To quantitatively analyze the text data and back up our observations, we used spaCy's named entity recognition, so we could identify how specific each model's output was as well as if the responses were relevant (Honnibal et al., 2020).

We decided to analyze two cases where LASER responded correctly, one where the base model was wrong and the other when higher-order LASER was wrong. Our findings consistently validate the observation made in the BigBench-WikidataQA dataset, indicating that the higher-order approximation label often includes filler words and lacks specificity, whereas the lower-order, standard LASER is able to pinpoint the predictions.

For these 3 tables, Set A denotes the samples where the base model was incorrect but standard LASER was correct. Set B denotes the samples where higher-order LASER was incorrect but standard LASER was correct.

Table 1. Average count of words/phrases which belong to an entity

	A	B
Base Model	1.67	1.69
LASER	2.54	2.44
High-order LASER	1.84	1.66

Table 2. Portion of responses where the model contains the same spaCy named entity as the true answer, averaged over Set A and Set B respectively

	A	B
Base Model	0.280	0.405
LASER	0.901	0.876
High-order LASER	0.583	0.301

Table 3. Portion of the generations in which no entities were found, averaged over Set A and Set B respectively

	A	B
Base Model	0.356	0.311
LASER	0.112	0.116
Higher-order LASER	0.238	0.321

Table 1 demonstrates that on average, LASER's generations have more named entities. Assuming that the answers to these prompts are typically named entities, this might indicate that LASER's generations have a higher tendency to contain an answer. Table 2 supports this assumption by showing that LASER has more named entities that match up with the answer. Table 3 demonstrates that the base model and higher-order LASER have more instances of zero found entities, which might indicate the lack of an answer, and more filler words.

3.3. Vast Pruning Potential

The results of exploring even smaller values of k (or in other words, rank) were very promising. As seen in table 4, Although very low ranks might not perform better than larger ranks, they still outperform the model with no intervention in all three datasets.

Table 4. The accuracy of each LASER altered model at different ranks. Notice how even at rank 5, the model still sees improvement over the base accuracy of the model without LASER insertion as reported by the paper

Desired Rank k:	Fever	Bbh qa	Bios pro
40	54.7	65.3	82.0
30	54.6	65.6	82.2
20	53.5	65.3	82.1
10	53.4	64.5	82.0
5	53.3	64.2	82.2
Base Model Accuracy	50.0	51.8	75.6

3.3.1. Finding a Metric

Our results for finding a metric to predict the hyper-parameters to be used for the LASER interventions were mixed. Calculating the nuclear norm as more and more singular values are considered did not work. Figure 2 was taken from the LASER paper and it plots the loss of different layers at different % of ablation. The relevant plot for our work is the one pertaining to the MLP Output Matrix. As you can see, the LASER intervention deprecates the performance of the model at earlier layers, while improving the performance when applied to later layers.

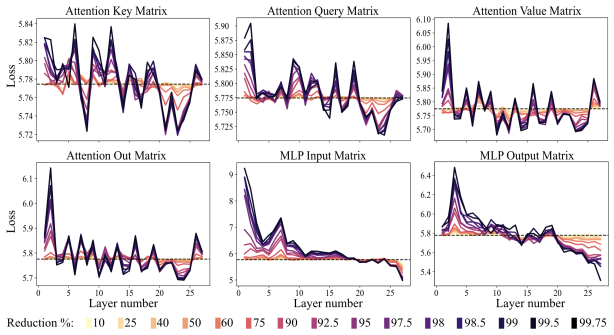

Figure 2. Loss for different hyper-parameters k for the different layers in the GPT-J transformer. (Sharma et al., 2023)

As seen in Figure 3, as we increase the rank k, the difference of the nuclear norm (i.e. % info) is not drastic for any of the layers. This does not match what we know regarding the later layers benefiting from LASER, and hence we do not believe this is a good metric to predict good LASER hyper-parameters.

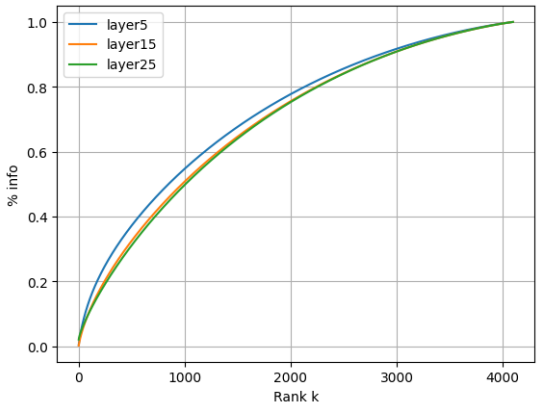

Figure 3. Nuclear Norm as a larger percentage of the singular values are considered (i.e. larger k) (Sharma et al., 2023)

Regarding our exploration of shared subspace, we have some positive results to speak of. After much trial and error, we found that comparing the shared subspace of the right singular vectors V_{l-1}^t and V_l^t with a V_l^t $t = 0.06$ produces a graph that is more in line with our knowledge of LASER's performance with the MLP Output layers. As seen in Figure 4, the full rank subspace percentage goes down at later layers, mirroring how the loss goes down when LASER is applied at later layers.

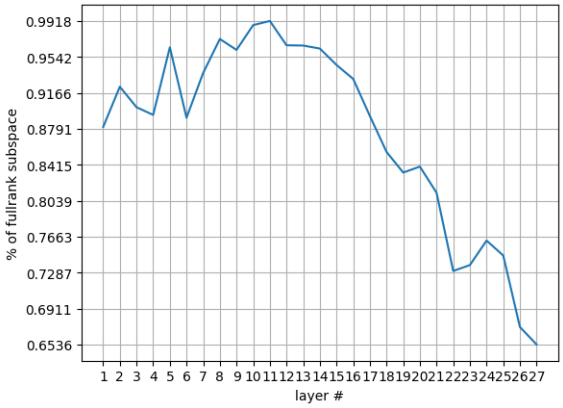

Figure 4. (Sharma et al., 2023) Percentage of full rank subspace for all the MLP Output Layers. The percentage of a given layer is calculated by dividing the rank of the column-wise concatenation of the MLP Output layer and the prior MLP Output Layer with what the rank would have been if the concatenation was full rank. Note that we start the plot at layer #1. This is because even though the first layer index in GPT-J is 0, layer #0 has no prior layer.

That being said, it is unclear how exactly to determine the mapping from "% of fullrank subspace" to hyperparameter k. Furthermore, this metric seems to be very sensitive to the value of t, as at slightly higher or lower values of t we do not obtain results that follow the known pattern of Figure 2. Hence, further work has to be done to validate the robustness of these results.

4. Process

Initially, our project aimed to delve into understanding LASER's performance across different layers within the transformer network and how it tackles noise removal at various layers. The plan was to identify the most effective layer and investigate the nature of the noise eliminated for each task. However, due to computational constraints, we had to recalibrate our approach and streamline the task. Instead, we used the layer and reduction level that the paper found was best for each dataset.

We initially conducted LASER experiments solely on the FEVER dataset. The FEVER dataset is structured solely on binary true/false answers, which while easy to work

with when analyzing output, limits our ability to thoroughly examine noise patterns in both higher and lower order predictions. To gain more nuanced insights, we expanded our scope to include additional datasets. We chose the BIOS-profession and BigBench-WikidataQA datasets because it provides more contextual information, enabling a more comprehensive analysis of noise characteristics. The BIOS-profession is a multi-class classification task, and BigBench-WikidataQA is an open-ended question-answering task, giving us a diverse range of tasks.This adjustment allowed us to gain a deeper understanding of how higher and lower order components behave across distinct datasets and noise removal for each component in various contexts.

We had some difficulties interpreting the results on the BIOS-profession dataset because of the distribution of the samples. As shown in Figure 5, 11,799 of the 19,212 total dataset has a true answer of "professor," so naturally, most of the data we investigated are instances of this. We tried running Term Frequency-Inverse Document Frequency (TF-IDF) on the prompts to analyze the most important words, but it was dominated by professor-related words like "research" or "education." To gain more insight on the other seven labels, we tried restricting TF-IDF to run on samples that belonged to a certain label, but there weren't enough samples. For example, if we wanted to run TF-IDF on the samples that LASER got correct but the higher-order intervention got incorrect, with the true label being "teacher," those restrictions leave us with only three samples.

Lastly, the search for a metric to help to choose the LASER hyperparameters was very challenging. Before we even arrived at the idea of calculating the nuclear norm, various forms of "effective rank" and "numeric rank" were attempted. After nuclear norm proved itself to not work, it took a while for us to come up with the idea of measuring shared subspace. Getting shared subspace to work also involved a lot of trial and error. Other than tuning the t threshold, we also tried different combinations of the right and left singular vectors, as well as experimented with row-wise concatenation. Something that helped us early on was focusing on the MLP Output Layers. Since we are working with an LLM, we did not have the means to validate our metric cheaply. Focusing on trying to match the graph in Figure 2 allowed us to iterate quickly since we could immediately validate whether a metric was likely wrong.

5. Contributions

Portions of this project were developed with the assistance of code from the LASER GitHub repository, https://github.com/pratyushasharma/laser, authored by Pratyusha Sharma, Jordan T. Ash and Dipendra Misra. We incorporated the parse_outputs.py script and altered the base code slightly for our experiment, including the higher-order approximation code. We also re-ran their code with different rank reduction rates. We used the spaCy library to run Named Entity Recognition for the BigBench-WikiQA dataset analysis, https://spacy.io/universe/project/video-spacys-ner-model-alt.

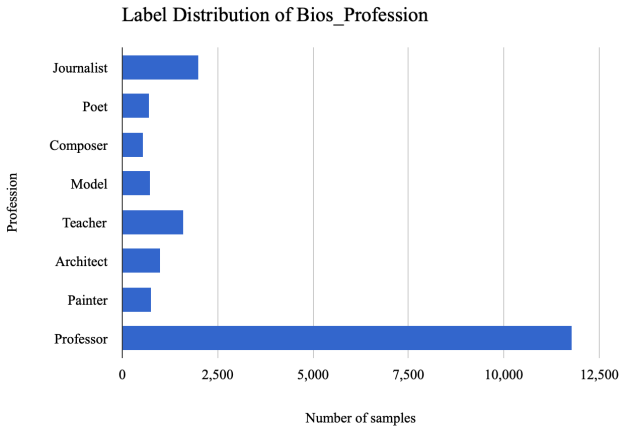

Figure 5. The distribution of labels is skewed towards "Professor".

6. Code

GitHub Repository

The commands we used to run the LASER intervention are outlined in the README.md of this repository. Results pertaining to the metric searching part of this report can be replicated by running the LASER_metric_exploration.ipynb notebook.

For the Data Exploration on the output for the Bios Dataset, simply run the .ipynb files with the resulting output.txt files.

To re-run the laser code, follow the instructions provided by the .readme, which is part of the original LASER GitHub, and to get different ranks reductions, change the rate argument.

References

Chaudhary, M. Tf-idf vectorizer scikit-learn, Jan 2021. URL https://medium.com/@cmukesh8688/tf-idf-vectorizer-scikit-learn-dbc0244a911a.

De-Arteaga, M., Romanov, A., Wallach, H. M., Chayes, J. T., Borgs, C., Chouldechova, A., Geyik, S. C., Kenthapadi, K., and Kalai, A. T. Bias in bios: A case study of semantic representation bias in a high-stakes setting. *CoRR*, abs/1901.09451, 2019. URL http://arxiv.org/abs/1901.09451.

Honnibal, M., Montani, I., Van Landeghem, S., and Boyd,

A. spacy: Industrial-strength natural language processing in python. *Zenodo*, 2020. doi: 10.5281/zenodo.1212303.

Sharma, P., Ash, J. T., and Misra, D. The truth is in there: Improving reasoning in language models with layer-selective rank reduction. *arXiv preprint arXiv:2312.13558*, 2023.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., Kluska, A., and Lewkowycz. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models, 2022. URL https://arxiv.org/abs/2206.04615.

Sun, M., Liu, Z., Bair, A., and Kolter, J. Z. A simple and effective pruning approach for large language models, 2024.

Thorne, J., Vlachos, A., Christodoulopoulos, C., and Mittal, A. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*, 2018.