Your Transformer is Secretly Linear

Anton Razzhigaev[1,2], Matvey Mikhalchuk[1,5], Elizaveta Goncharova[1,4], Nikolai Gerasimenko[3,5], Ivan Oseledets[1,2], Denis Dimitrov[1,3], and Andrey Kuznetsov[1,3]
[1]AIRI, [2]Skoltech, [3]SberAI, [4]HSE University,
[5]Lomonosov Moscow State University
razzhigaev@skol.tech

Abstract

This paper reveals a novel linear characteristic exclusive to transformer decoders, including models such as GPT, LLaMA, OPT, BLOOM and others. We analyze embedding transformations between sequential layers, uncovering a near-perfect linear relationship (Procrustes similarity score of 0.99). However, linearity decreases when the residual component is removed due to a consistently low output norm of the transformer layer. Our experiments show that removing or linearly approximating some of the most linear blocks of transformers does not affect significantly the loss or model performance. Moreover, in our pretraining experiments on smaller models we introduce a cosine-similarity-based regularization, aimed at reducing layer linearity. This regularization improves performance metrics on benchmarks like Tiny Stories and SuperGLUE and as well successfully decreases the linearity of the models. This study challenges the existing understanding of transformer architectures, suggesting that their operation may be more linear than previously assumed. [1]

1 Introduction

Transformers have revolutionized the field of natural language processing, offering unprecedented advances in a wide range of applications (Islam et al., 2023). However, despite their widespread adoption and success, the complex work of these models remains an area of active research (Lin et al., 2021). One aspect that has received less attention is the inherent linearity of intermediate embedding transformations within these architectures. In this study, we embark on an in-depth analysis of the linearity properties of transformers, specifically focusing on decoders, and explore its implications during the pretraining and fine-tuning phases.

Our investigation reveals a surprising discovery: the embedding transformations between sequential layers in transformer decoders exhibit almost linear properties. This observation is quantified using Procrustes similarity analysis, demonstrating a near-perfect linearity score of 0.99. Such a discovery not only challenges the traditional understanding of transformer architectures but also opens new opportunities for model optimization and efficiency.

Based on this insight, we introduce several new contributions to the field:

- Extensive analysis of the linearity properties of transformer decoders and its dynamics at the pretraining and fine-tuning stages.
- The development of new algorithms for depth pruning of transformer decoders, allowing to remove the most linear layers without a significant loss in performance.
- A novel distillation technique that involves pruning, replacing certain layers with linear approximations, and then distilling layer-wise embeddings to preserve model performance.
- Introducing a new regularization approach for pretraining based on the cosine similarity, designed to decrease the layer linearity. This method not only enhances the performance of transformer models on benchmark datasets such as SuperGLUE and TinyStories Eldan and Li (2023), but also improves the expressiveness of embeddings, as evidenced by linear probing tasks.

With our findings, we are paving the way for more computationally efficient transformer architectures without sacrificing their effectiveness, thereby addressing one of the critical challenges in deploying these models.

[1] https://github.com/AIRI-Institute/LLM-Microscope

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5376–5384

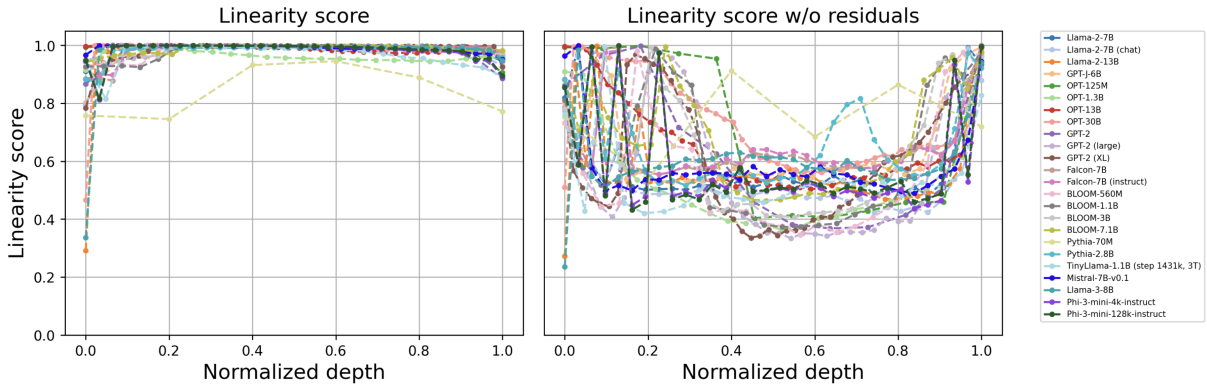

Figure 1: Linearity profiles for different open source models. Normalized depth is the layer index divided by the total depth.

2 Related Work

Research on evaluating and leveraging sparsity for model pruning has become one of the most significant topics within the machine learning community. Molchanov et al. (2016) explored the sparsity of convolutional neural networks through backpropagation and fine-tuning, laying the groundwork for understanding the potential applications of sparsity in resource-efficient inference. The verification approach utilized in a more recent DejaVu (Borse et al., 2023) paper is based on Molchanov's research.

Previous work (Kurtic et al., 2023) has addressed the challenges associated with naive sparse fine-tuning in the context of LLMs. Issues such as training instability, poor recovery, and overfitting have prompted an exploration for alternative approaches. The study introduced SquareHead distillation, a method that consistently addresses the challenges in naive sparse fine-tuning, demonstrating accurate recovery even at high sparsity levels.

In a more recent study WANDA (Sun et al., 2023), the authors present a technique for pruning LLMs to high degrees of sparsity without modifying the remaining weights. Unlike SparseGPT (Frantar and Alistarh, 2023), WANDA seamlessly implements pruning in a single forward pass, leveraging feature norm statistics for efficient pruning. This method achieves noticeable sparsity without the need for a sophisticated iterative weight update procedure, differentiating itself from other pruning techniques.

Contextual sparsity introduced by Borse et al. (2023) involves sparsifying MLP and attention blocks in LLMs to reduce generation latency. The study identifies essential attention heads and MLP neurons for computation, maintaining performance across in-context learning and language modeling tasks.

Recent work by Ashkboos et al. (2024) shows that LLMs can be sparsified post hoc. Their approach introduces a scheme to replace each weight matrix with a smaller dense matrix, thereby reducing the dimensionality of the networks. Their results show that models of different sizes can be reduced with varying degrees of success. For example, LLAMA-2 70B and OPT 66B can maintain 99% zero-shot accuracy while reducing 25% of the parameters reduced while performing LLM evaluation tasks. In contrast, the smaller Phi-2 is more sensitive to pruning, experiencing a 10% drop compared to its dense version.

The inner structure of transformer models has captured significant attention among researchers (Nostalgebraist, 2020; Xu et al., 2021; Belrose et al., 2023; Din et al., 2023). Primarily, in "logit lens" (Nostalgebraist, 2020) and subsequently in (Belrose et al., 2023), the authors have focused on analyzing how hidden representations evolve across different layers of transformer architecture, aiming to elucidate their impact on final model outputs. Complementing these findings, the Anthropic team's research into small transformer-based models (Elhage et al., 2021) uncovers a profound linear structure inherent in this architecture. Their work demonstrates the effectiveness of decomposing operations into individual sum components and multiplying chains of matrices, thus highlighting the linear complexity within these sophisticated neural architectures.

Structure-based pruning Topological features that analyze the structure of inner embeddings in

transformer-based models are also useful in LLM pruning and distillation. Previous research examined the intrinsic dimensionality of neural networks to evaluate their capacity and effectiveness in the fine-tuning process (Ansuini et al., 2019; Aghajanyan et al., 2020; Razzhigaev et al., 2023). Decoder-based models are shown to achieve a high level of anisotropy, especially in their middle layers, and have low intrinsic dimensionality (Razzhigaev et al., 2023). Recent popular approaches include low-rank approximation, which replaces or adjusts the weight matrix with the product of two matrices with a smaller inner dimension. This approach typically requires a fine-tuning procedure that adjusts the matrix representations. For example, LoRA (Hu et al., 2021) was inspired by the previous work (Aghajanyan et al., 2020) showing that neural networks can be successively trained in lower-dimensional subspaces. The research also shows that there it is not necessary to update millions of parameters on small fine-tuning datasets. Our results are on par with the results of this research, showing that via fine-tuning, the linearization of models grows steadily.

The Bonsai model (Dery et al., 2024) tends to prune the LLMs relying only on the inference step, while they achieve performance comparable to half-sized semistructured sparsity with WANDA 2:4 and outperforms the LLM-Pruner (Ma et al., 2023) and LoRAPrune (Zhang et al., 2023) on 4 out of 6 evaluation settings in the experiments conducted.

In this paper, we investigate several techniques for pruning LLMs, leveraging the linearity of the decoder-based layers. Our techniques offer efficient yet lightweight methods, maintaining high model performance on the evaluated benchmarks.

3 Analysis of Pretrained Architectures

In our study of the embedding properties of various layers of transformer decoders, we focus on understanding the degree of linearity and smoothness of transformations between sequential layers.

3.1 Linearity Score

To determine the degree of linear dependence of two sets of vectors, we used a metric obtained by generalizing the Procrustes similarity (Gower, 1975) to the case of arbitrary linear transformations.

Let $X, Y \in \mathbb{R}^{n \times d}$ represent the centered sets of embeddings, to calculate linearity score we use normalized matrices $\tilde{X} = X/||X||_2$, $\tilde{Y} = Y/||Y||_2$

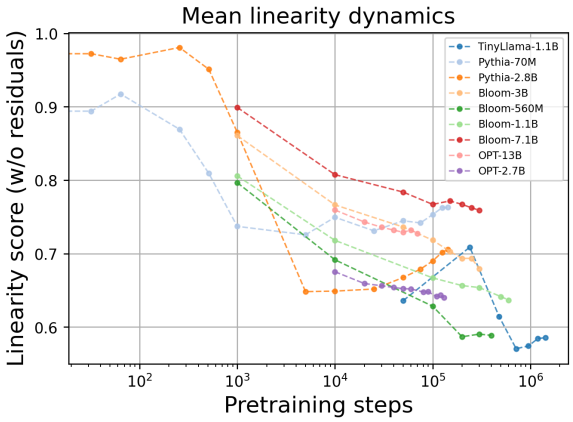

Figure 2: Linearity score (averaged across layers) at different pretraining steps of open source models.

(where $||\cdot||_2$ denotes the Frobenius norm of the matrix) and defined

$$\text{linearity\_score} := 1 - \min_{A \in R^{d \times d}} ||\tilde{X}A - \tilde{Y}||_2^2$$

This is almost the same formula as in Procrustes similarity, the only difference is that, instead of considering the minimum among orthogonal transformations, we use the minimum among all linear transformations to find the optimal mapping in terms of squared errors.

We chose such approach for its robustness in evaluating the linearity of embeddings, especially considering the scale variance across transformer layers. Unlike L_2 norm, which lacks scale invariance, Procrustes normalization offers a bounded metric in the range [0,1].

Surprisingly, the linearity scores of layers in all tested transformer decoders were found to be close to 1, indicating a high degree of linearity in embedding transformations (Figure 1).

This phenomenon can be partly explained by the observation that the norm of each block's contribution to the residual stream is remarkably low (Figure 3). Moreover, when assessing the linearity of the main stream (embeddings w/o residual component) by subtracting the embedding values of each layer from the previous layer, one can notice that the degree of linearity significantly decreases (Figure 1). This suggests that the inherent linearity is not as straightforward as it is initially estimated. Moreover, the low norm contribution of individual blocks resulted in embeddings from adjacent layers being closely aligned in terms of cosine similarity.

One more insight is that the combination of seemingly linear blocks can lead to non-linear out-

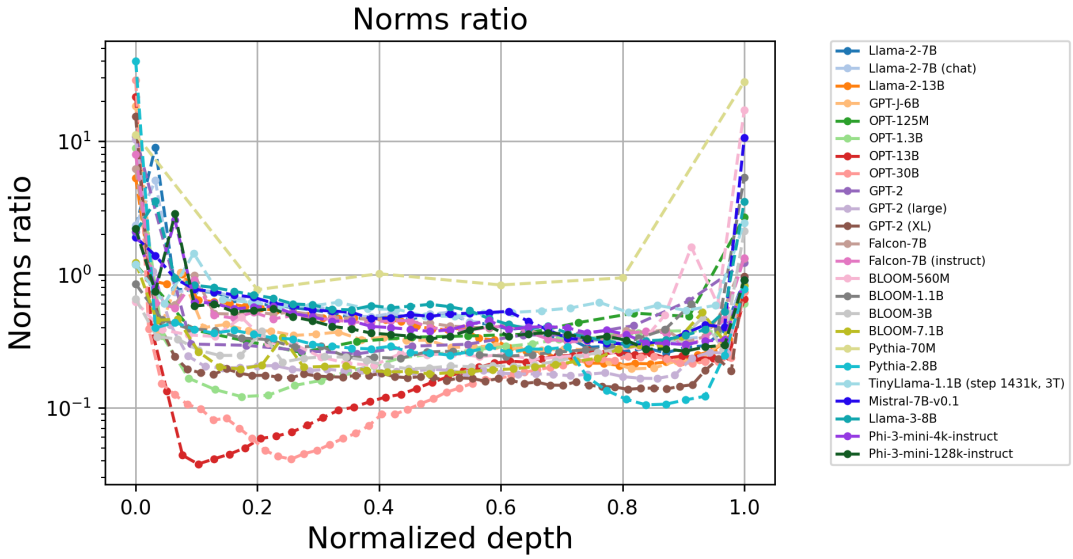

Figure 3: The relationship between transformer block output norm and resulted residual stream embedding norm.

comes. Elhage et al. (2022) suggests that complex features can be encoded across components of neural networks, applicable to attention heads in transformers. This indicates that the cumulative effect of linear transformations might enable the encoding of intricate non-linear representations.

Furthermore, our feature triggering regime hypothesis proposes that rare specific features on a few tokens with high non-linearity significantly influence model behavior — in the Figure 9 one can see that some layers of OPT-1.3B have the long tailed distribution of L_2 errors, which means that there are still sparse spikes of non-linearity.

Borse et al. (2023) explored how a sparse subset of model parameters can be dynamically activated for efficient inference, supporting the idea that within predominantly linear architectures, certain non-linear interactions are crucial for model functionality.

3.2 Linearity Dynamics at Pretraining and Fine-tuning

Our exploration extends to examining the linearity dynamics of both open-source models with publicly available intermediate checkpoints and our custom models trained on small datasets. Through this analysis, we aimed to understand the dynamics of linearity, especially in the main stream (contextualized embeddings including the residual component), across different stages of model training.

As illustrated in the Figure 2, the analysis reveals a notable trend: as the models undergo pretraining, the linearity of the main stream gradually decreases on average. This phenomenon is consistently observed in all models examined, indicating a fundamental aspect of transformer-decoder learning dynamics.

In our analysis of the fine-tuning phase across diverse tasks, including those in the SuperGLUE benchmark (Wang et al., 2019) and the reward-modeling task on the Anthropic-Helpful dataset (Bai et al., 2022), we notice an interesting change. Contrary to the decreasing trend of linearity observed during the pretraining phase, all models under study show an increase in linearity during fine-tuning. This finding indicates that task-specific fine-tuning tends to reinforce and amplify the linear characteristics of transformer models, as shown in Table 1.

In fine-tuning, we train models on three NLI tasks from the SuperGLUE benchmark: MultiRC, BoolQ, and CB, treating them as binary text classification challenges. In the BoolQ task, for instance, we combine the question and the passage into a single text, marking them with "question:" and "passage:" respectively, and consider the binary answer as the classification label.

Reward models trained on text pairs with contrastive loss (Ouyang et al., 2022) demonstrate a similar trend in linearity scores, proving even more stability across different seed values.

Model Name	Super_Glue/MultiRC	Super_Glue/BoolQ	Super_Glue/CB	Reward Modeling
OPT-125M	0.085 ± 0.008	0.217 ± 0.038	0.048 ± 0.009	0.060 ± 0.008
OPT-1.3B	0.055 ± 0.021	0.382 ± 0.004	0.088 ± 0.010	0.062 ± 0.007
OPT-2.7B	0.061 ± 0.025	0.356 ± 0.005	0.066 ± 0.029	0.054 ± 0.003
Llama2-7B	0.141 ± 0.006	0.051 ± 0.024	0.081 ± 0.070	0.194 ± 0.027
GPT2	0.085 ± 0.021	0.048 ± 0.016	0.004 ± 0.003	0.092 ± 0.013
GPT2-Large	0.049 ± 0.003	0.023 ± 0.008	0.025 ± 0.014	0.085 ± 0.008
GPT2-XL	0.040 ± 0.007	0.037 ± 0.007	0.028 ± 0.019	0.038 ± 0.008

Table 1: Delta of linearity score w/o residuals after fine-tuning various tasks. Note that all values are strictly positive, which means that linearity always increases during fine-tuning.

Model/Task	boolq	cb-acc	cb-f1	copa	multirc	record-f1	record-em	rte	wic	xstorycloze-en	mean
Mistral 650M	48.50	**42.86**	21.96	56.0	56.97	21.80	21.05	51.26	**51.10**	61.75	43.33
Mistral 650M + cosine (0.5)	**57.50**	41.07	**28.57**	**61.0**	**57.10**	**23.20**	**22.54**	**55.23**	50.00	**64.39**	**46.06**
Mistral 150M	38.84	**42.86**	**27.39**	56.0	44.16	20.07	19.42	51.26	51.10	59.89	41.10
Mistral 150M + MSE (0.5)	38.84	39.29	19.30	60.0	**57.59**	20.46	19.77	**53.07**	50.47	57.64	41.64
Mistral 150M + MSE (2.0)	39.39	41.07	19.41	57.0	46.53	**22.62**	**21.89**	51.99	50.00	56.52	40.64
Mistral 150M + cosine (0.5)	**44.16**	37.50	24.18	**62.0**	54.54	21.67	20.99	50.90	50.47	**61.35**	**42.78**

Table 2: SuperGLUE results.

4 Improving Linearity with Regularized Pretraining

Aiming to understand the impact of linearity on transformer models, we embark on pretraining experiments using the Mistral architecture with model sizes of 150M, and 650M. These models are pretrained on carefully selected clean datasets, TinyStories (Eldan and Li, 2023) and Tiny-textbooks (Li et al., 2023), chosen for their diverse and rich content, which has been proven to be suitable for fast training of the small models (Zhao et al., 2023) and architecture experiments (Sharifnassab et al., 2024).

We introduce specific loss terms to adjust the relations between embeddings within transformer layers:

- **MSE regularization term:** Experimentation with mean squared error (MSE) loss between embeddings of consecutive layers, designed to minimize the distance between these embeddings, thereby promoting consistency across the layers.

$$L_{MSE} = \lambda \sum (\|emb_i - emb_{i-1}\|^2).$$

- **Cosine Similarity regularization term:** The application of a cosine-based regularization that encourages contextualized embeddings from sequential layers to align closer to each other, effectively reducing their angular difference to zero on average.

$$L_{cosine} = \lambda \sum (1 - \cos(emb_i, emb_{i-1})).$$

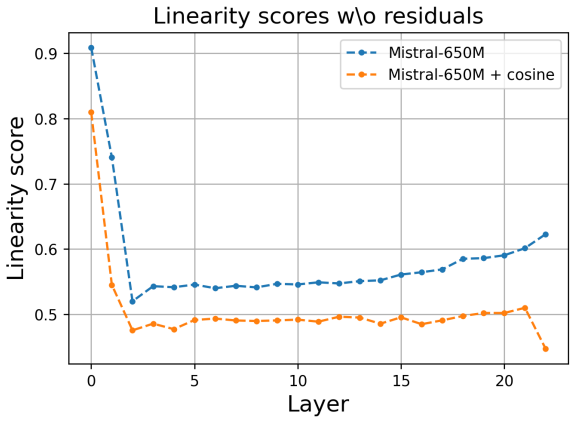

Figure 4: Linearity score of different layers with and without cosine regularization used at pretraining.

The most promising results are achieved using a cosine-based approach that encourages the embeddings of sequential layers to converge, effectively making the cosine similarity between them closer to 1 on average. This method shows significant perspectives in the enhancing model performance. We evaluate the effectiveness of our approach through validation using GPT-4 on TinyStories prompts according to the Eldan and Li (2023) methodology, linear probing techniques, and evaluation on SuperGLUE benchmarks. The results are presented in the Table 2 and Table 3. As it can be seen in the Figure 5, linearity scores are lower at each layer of the model after pretraining with such regularization.

To further assess the expressiveness of embeddings across different layers, we conducted linear probing on outputs of all the layers of the Mistral-650M model, both pretrained with and without cosine regularization, on the xstorycloze-en task

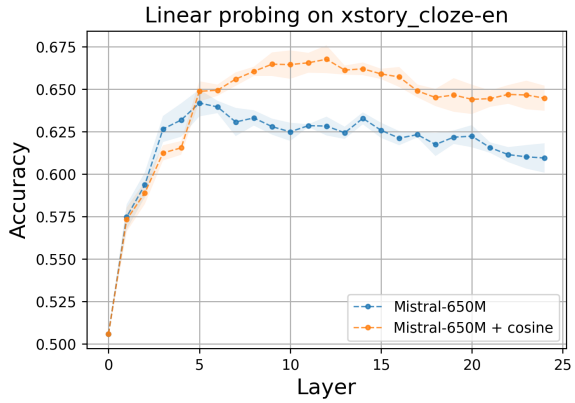

Figure 5: Linear probing of embeddings from different layers of Mistral-650M pretrained with and without suggested cosine regularization.

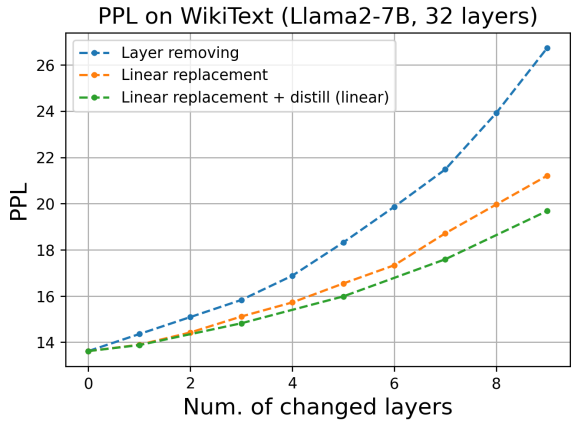

Figure 6: Perplexity on WikiText for various pruning and distillation methods (lower is better).

Mistral config	Grammar	Creativity	Consistency	Plot	Mean
650M	5.47	6.60	4.81	4.67	5.39
650M + cosine (0.5)	**6.07**	**7.02**	**5.74**	**5.48**	**6.08**
150M	4.88	6.51	4.16	3.88	4.86
150M + MSE (0.5)	**5.19**	6.70	4.47	4.20	5.14
150M + MSE (2.0)	5.00	6.81	4.56	4.29	5.17
150M + cosine (0.5)	5.14	**6.91**	**4.77**	**4.95**	**5.44**

Table 3: TinyStories prompts completions evaluation via GPT-4

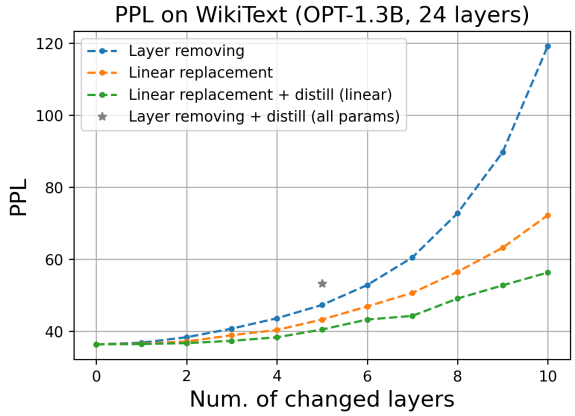

Figure 7: Perplexity on WikiText for various pruning and distillation methods (lower is better).

from SuperGLUE. The results clearly indicate that embeddings from the model pretrained with regularization exhibit better performance compared to those from the standard model (Figure 4).

This contradictory outcome, where the term appears to draw embeddings from neighbouring layers closer together, making them more similar in terms of cosine similarity, has prompted a deeper investigation. Our observations suggest that as embeddings become more similar across layers, the model may compensate for the reduction in variability by amplifying non-linear processing capabilities in the residual stream. Although this hypothesis requires further exploration, it offers a fascinating insight into the adaptive mechanisms of transformer models in response to altered internal dynamics.

5 Exploiting Linearity for Pruning

Leveraging the inherent linearity of transformer layers, we explore a pruning strategy that sequentially removes the most linear layers. This approach allows you to reduce the size of the model slightly by removing just a few layers without significantly compromising performance. Further enhancement of this strategy involves replacing the pruned layers with linear approximation and incorporating a distillation loss (specifically MSE layerwise) to minimize performance degradation. The training focuses on these linear replacements, fine-tuning them to effectively mimic the original layers' function. The effectiveness and the impact of these methods are detailed in the Figure 8. We use TinyStories for linear approximation and distillation training stage. As it can be seen in the Figure 7, perplexity is less affected by pruning with linear replacements and following distillation compared to just removing transformer layers.

6 Conclusion

In our study we provide an in-depth exploration of linearity within transformer decoders, revealing their inherent near-linear behavior in various models. We discover that while pretraining tends to increase nonlinearity within layers, fine-tuning on specific tasks can paradoxically reduce it. We propose new pruning and distillation techniques inspired by previous observations, demonstrating

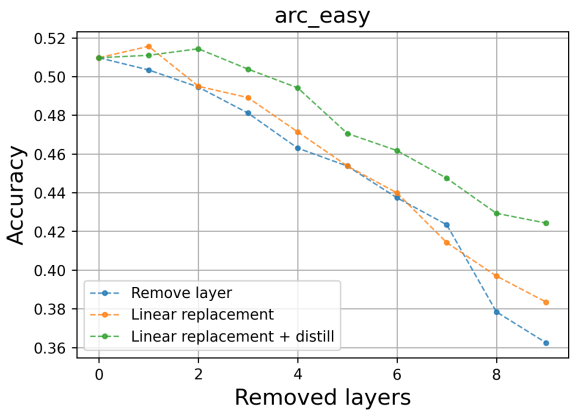

Figure 8: OPT-1.3B results on ARC-easy dataset with suggested pruning techniques.

that it is possible to refine and optimize transformer models without compromising their performance. The suggested cosine-based regularization approach during pretraining further contributes to model efficiency and performance on benchmarks such as SuperGLUE and TinyStories, while reducing the linearity of its layers (w/o residual components).

Our study highlights the significant relationship between linearity and performance of transformer decoders, offering strategic guidance for future developments in the efficiency and flexibility of these models.

7 Limitations

Despite the promising advancements presented in this study, it is essential to acknowledge its limitations. Firstly, our analysis predominantly focuses on transformer decoders, thus the generalizability of our findings to encoder-only or encoder-decoder architectures may be limited.

Secondly, the depth pruning and distillation techniques, while being effective in our experiments, were evaluated within a specific set of conditions and models. The scalability of these methods to larger, more complex models or different domains is yet to be fully ascertained.

Moreover, the new regularization approach aimed at pretraining demonstrates potential, yet its effectiveness across a broader spectrum of tasks requires further validation.

8 Ethics Statement

We are committed to ethical principles for AI research, focusing on transparency and responsible experimentation. Our research, while suggesting efficiency improvements, prompts consideration of implications such as privacy and fairness.

References

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. 2020. Intrinsic dimensionality explains the effectiveness of language model fine-tuning.

Alessio Ansuini, Alessandro Laio, Jakob H. Macke, and Davide Zoccolan. 2019. Intrinsic dimension of data representations in deep neural networks.

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. 2024. Slicegpt: Compress large language models by deleting rows and columns.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens.

Shubhankar Borse, Debasmit Das, Hyojin Park, Hong Cai, Risheek Garrepalli, and Fatih Porikli. 2023. DejaVu: Conditional regenerative learning to enhance dense prediction.

Lucio Dery, Steven Kolawole, Jean-François Kagy, Virginia Smith, Graham Neubig, and Ameet Talwalkar. 2024. Everybody prune now: Structured pruning of llms with only forward passes.

Alexander Yom Din, Taelin Karidi, Leshem Choshen, and Mor Geva. 2023. Jump to conclusions: Short-cutting transformers with linear transformations.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english? *CoRR*, abs/2305.07759.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *CoRR*, abs/2209.10652.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan†, Nicholas Joseph†, Ben Mann†, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah‡. 2021. A mathematical framework for transformer circuits.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot.

J. Gower. 1975. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Saidul Islam, Hanae Elmekki, Ahmed Elsebai, Jamal Bentahar, Najat Drawel, Gaith Rjoub, and Witold Pedrycz. 2023. A comprehensive survey on applications of transformers for deep learning tasks.

Eldar Kurtic, Denis Kuznedelev, Elias Frantar, Michael Goin, and Dan Alistarh. 2023. Sparse fine-tuning for inference acceleration of large language models.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: phi-1.5 technical report.

Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. 2021. A survey of transformers.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2016. Pruning convolutional neural networks for resource efficient transfer learning. *CoRR*, abs/1611.06440.

Nostalgebraist. 2020. interpreting GPT: the logit lens. https://www.alignmentforum.org/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Anton Razzhigaev, Matvey Mikhalchuk, Elizaveta Goncharova, Ivan Oseledets, Denis Dimitrov, and Andrey Kuznetsov. 2023. The shape of learning: Anisotropy and intrinsic dimensions in transformer-based models.

Arsalan Sharifnassab, Saber Salehkaleybar, and Richard Sutton. 2024. Metaoptimize: A framework for optimizing step sizes and other meta-parameters. *ArXiv*, abs/2402.02342.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2023. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *CoRR*, abs/1905.00537.

Hongfei Xu, Josef van Genabith, Qiuhui Liu, and Deyi Xiong. 2021. Probing word translations in the transformer and trading decoder for encoder layers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–85, Online. Association for Computational Linguistics.

Mingyang Zhang, Hao Chen, Chunhua Shen, Zhen Yang, Linlin Ou, Xinyi Yu, and Bohan Zhuang. 2023. Loraprune: Pruning meets low-rank parameter-efficient fine-tuning.

Xingmeng Zhao, Tongnian Wang, Sheri Osborn, and A Ríos. 2023. Babystories: Can reinforcement learning teach baby language models to write better stories? *ArXiv*, abs/2310.16681.

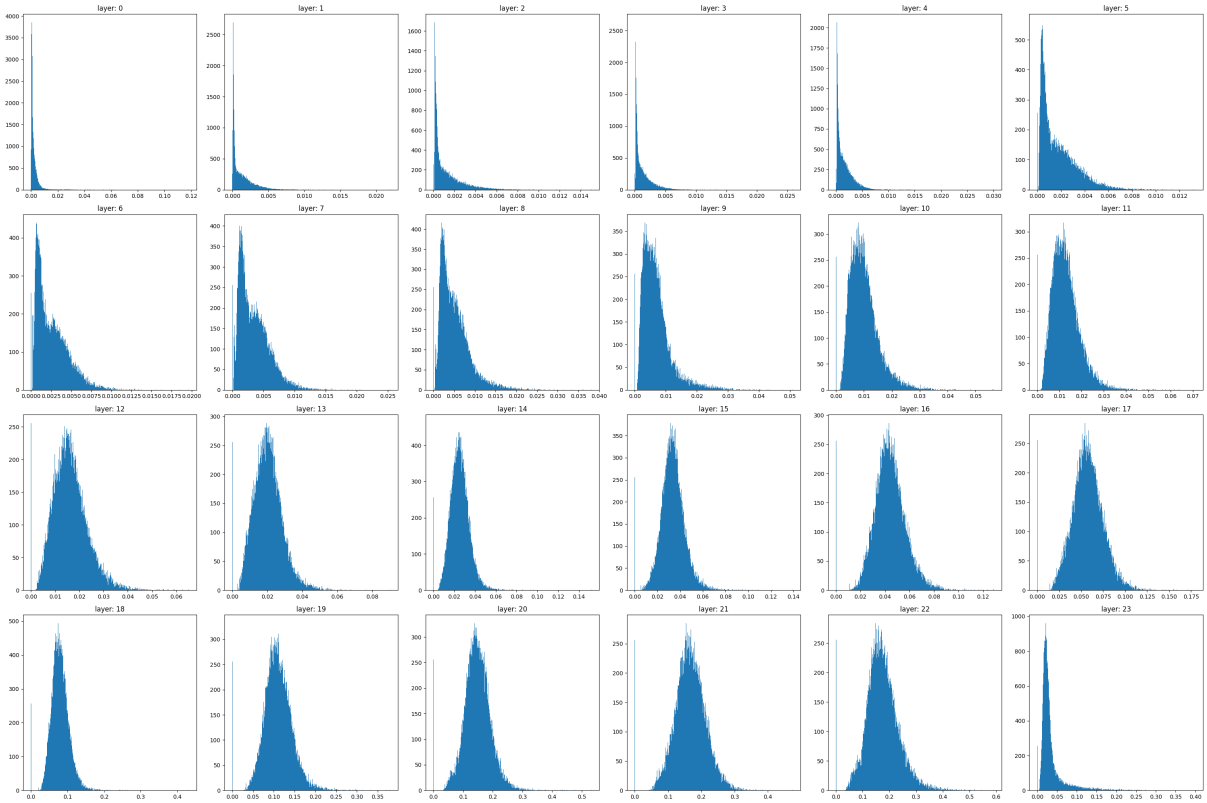

Figure 9: L_2 error distribution of linear approximation across different layers of OPT-1.3B.

A Error Distribution by Layers

In the Figure 9 we present a visualization of L_2 error distribution across several layers of OPT-1.3B decoder architecture.