On Difficulties of Attention Factorization through Shared Memory

Uladzislau Yorsh [a;*], **Ondřej Bojar**[a;**] **and Martin Holeňa** [b]

[a]Faculty of Mathematics and Physics, Charles University, Prague
[b]Institute of Computer Science, Czech Academy of Sciences, Prague
ORCiD ID: Uladzislau Yorsh https://orcid.org/0009-0003-2361-8073, Martin Holeňa https://orcid.org/0000-0002-2536-9328

Abstract.

Transformers are powerful deep learning models that have revolutionized various fields like natural language processing, computer vision, and audio processing. Their strength lies in the attention mechanism, which enables learning complex dependencies between inputs. However, this attention mechanism also comes with quadratic time and memory complexity, making it challenging to apply the model to larger inputs. Among other approaches to accelerate computation, researchers are exploring models that use external learnable memory to reduce attention computation to linear complexity. We focus on one such model, Linear Unified Nested Attention (Luna), to examine how memory size and block connectivity affect model training convergence and predictive accuracy. Our research brings some counter-intuitive results regarding the Luna model, e.g. that the size of the memory has much less impact on the performance than one might expect, and that the block might significantly benefit from layer reordering.

1 Introduction

In the era of big data and natural language processing, the ability to handle long-form text is becoming increasingly important. From news articles to scientific papers, legal documents to social media posts, there is a growing need for deep learning models that can efficiently process and understand large bodies of text. While Transformers have already shown promise in certain tasks, they scale poorly when faced with inputs that are too long—the attention framework, which serves as the basis of the model, induces the quadratic time and memory complexity, and this is where the so called *efficient Transformers* come in.

This model family, which spawned shortly after the paper that proposed Transformer [6], aims to address the computational complexity of the architecture. The models, among which are Longformer, Reformer, Transformer-XL and dozens of other approaches, are designed to handle sequences that exceed the typical length of a Transformer input which ranges between 1024 to 4096 tokens. The increased processing rate is achieved by introducing various modifications to the Transformer architecture, in particular to the attention mechanism.

At a high level, the attention mechanism works by assigning weights to different parts of the input based on how relevant they are to the task at hand. These weights are used to re-encode input elements as weighted sums of other vectors, either a different input or the same input after transformation. The new representation is then used as the input to the next layer of the model.

A simplified attention operation can be written as:

$$\text{Att}(Q, K, V) = \text{softmax}\left(QK^T/\sqrt{d_k}\right)V$$

where $(Q, K, V) = (XW_q, CW_k, CW_v)$ and (X, C) are the two module inputs. As it may be seen from the definition, given the inputs $X \in \mathbf{R}^{L\times d}$ and $C \in \mathbf{R}^{M\times d}$, the time and memory complexity of computing the attention is $\mathcal{O}(LMd)$, or $\mathcal{O}(L^2d)$ when $X = C$, which is a common case.

In the present work, we focus on the models that employ external learnable memory to reduce the attention complexity. This memory functions as an input used as X or C, and interacts with the original input through an attention operation. The model thus is intended to make memory cells to adapt to the data and learn the most suitable vectors for compressing an input sequence, while a random initialization should provide the diversity needed to break weight symmetry.

2 Related Work

One of the first models exploiting the described framework was the Set Transformer [3], which exploits the Transformer permutation equivariance. To deal with larger sets, the authors suggest the *inducing points* concept. Instead of performing self-attention directly, the authors suggest first to perform a cross-attention between a learnable memory $M \times d$ and input $L \times d$; after that, the projected induced points are used as a contextual input to the cross-attention. This may be seen as an analogue of an approximation of a matrix by a product of two lower-ranked ones; additionally, the attention framework is being employed for compression.

While the Set Transformer uses the uncontextualized memory for all projections, the Linear Unified Nested Attention (Luna) framework employs a *contextualized* approach. The main extension over Set Transformer is that the transformed memory is being sent to the next layer as a memory input. This allows to employ stacked Transformer layers to refine hidden memory representation for better input compression.

* Corresponding Author. Email: vlad.yorsh@mff.cuni.cz. The researcher was supported by the grant №290223 of the Charles University Grant Agency.
** Ondřej Bojar was supported by the grant 19-26934X (NEUREM3) of the Czech Science Foundation.

The Perceiver [2] model demonstrates an orthogonal approach to use the memory as an attention input: instead of using it as a proxy in an attention approximation producing the output of the same size, the model performs a series of memory-to-input cross attention operations followed by self-attentions on memory cells. To circumvent the lower efficiency of such operations[1] compared to the vanilla attention, the authors suggest to build a significantly deeper model with an extensive weight sharing.

3 Research Questions

In the reported research, we have addressed the following questions:

Q1: How effectively the memory is being used? We are going to analyze and visualize the memory changing dynamics in the Luna model. We also conduct experiments to establish, what is the lowest number of memory cells for the model to provide comparable performance.

Q2: How memory utilization can be improved? During preliminary experiments we have found, that the base Luna variant may not use its memory optimally. We suggest that countering the issue can lead to significant performance improvements.

Q3: What causes training instabilities in Luna? Additionally we have observed, that during training certain configurations of Luna after reaching peak performance can degenerate down to random prediction accuracy. Therefore, one of the objectives is to find the reasons and try to alleviate the problem.

4 Experimental Investigation

We conduct our experiments on the Long Range Arena (LRA) [5] benchmark, which we consider a preferable choice due to the following reasons:

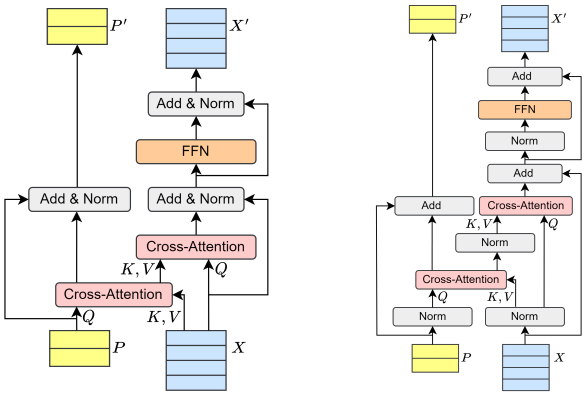
- **Training from scratch.** The benchmark suggests to train the models directly on the data without pretraining. This allows for a fairer comparison, since sophisticated pretraining pipelines have a significant impact on the final result and would not allow to judge about the inductive bias.
- **Parametrization limitation.** Another setup recommended for a fair comparison is to keep the relative number of additional parameters below 10%. This shifts the focus to the implementation of the proper architecture instead of making use of increased parametrization. We have found that some competing models may violate the condition (e.g. Linformer), but in general they stay below this limitation.
- **Variety of already tested models.** The LRA results appear in dozens of papers on alternative attention mechanisms, which allows for a detailed comparison—either between individual architectures or even whole categories.

To perform our research we have selected the following three subtasks of the benchmark:

- **Byte pair-encoded (BPE) text classification.** This task consists in binary classification (positive/negative sentiment) of the IMDB dataset texts encoded as byte pairs. This creates input sequences up to $4k$ tokens long with a relatively short subword units, which makes this task significantly more difficult than the ordinary IMDB classification.
- **BPE text matching**. The dataset is the ACL Anthology Network, encoded in a way similar to the previous task with sequences up to $4k$ tokens long. The model needs to process two inputs and to use the concatenated hidden representations as an input to the final two-class classifier to determine, whether there is a citation link between the documents.
- **ListOps**. This task consists in processing nested arrays of digits, coupled with aggregation operations such as max, min, median and sum modulo, $2k$ tokens long in total. The model has to predict the correct digit out of ten. This task tests the ability of the model to process hierarchical inputs. An example input looks like:
`[MED 9 [MAX 4 [MIN 6 3 7 8 9 X 1 2 ...`

We have not selected the Pathfinding task due to its high computing demands, nor the image classification task due to its similarity to the BPE text classification.

In our experimental setup, we closely follow the original LRA model architectures,[2,3] including the Post-Layer Normalization (Post-LN) [7] scheme.



(a) Luna (b) PreLuna

Figure 1: Base Luna model versus the proposed PreLuna. We rearrange normalization layers and insert an additional one between packing and unpacking attention.

Signal Vanishing. During the preliminary experiments with Luna, we have observed that the output generated by cross-attentions between input and memory has a significantly lower variance compared to the Transformer self-attention. The output of the attention are different convex combinations of value vectors, and given the random initialization with small weights, this leads to almost evenly distributed attention scores. Such an attention distribution leads to very similar output vectors generated for each query at the beginning of the training.

This flat character of the attention, the lack of attention of tokens to themselves and the consecutive packing-unpacking operations may lead to quick vanishing of signal. Because the normalization is being applied on the sum of original and transformed inputs, the extracted information signal does not get amplified.

To counter the issue, we propose to apply the Pre-Layer Normalization (Pre-LN) scheme on Luna [7]. In the case of vanilla Trans-

[1] Under the configuration setup given in the next sections, the Perceiver-like models did not converge on the considered datasets during preliminary experiments. We are thus not reporting them here.

[2] We use the PyTorch + PyTorch Lightning libraries for implementing the models and the original LRA code for data generation

[3] To accelerate the training, we apply automatic mixed precision with the `bfloat16` as the target format. We have not observed any significant accuracy drops in this setup, unlike in the case of basic `float16`.

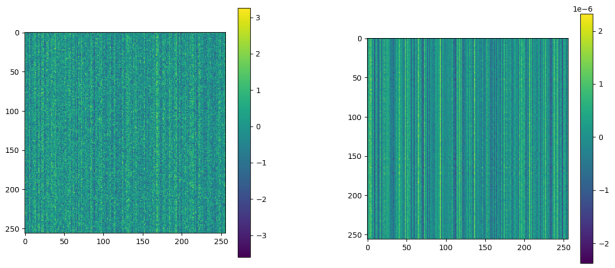
formers, Pre-LN tends to show consistent improvements over the Post-LN scheme [7]. Given that Luna has two attentions following each other, we find it even more important to apply normalization in a different way than it was originally proposed by LRA and Luna authors.

The new PreLuna block (see Figure 1) is arranged similarly to the Luna, with the main difference in normalization blocks placement: two are being applied first attention inputs (memory and input tokens), one to the attention output, and one to the FFN input. Note that LayerNorms do not interfere with skip connections and that PreLuna has one LayerNorm more than standard Luna. We test the model on the three suggested tasks, and the results are shown in Table 1.

Table 1: Vanilla Transformer and Luna compared with PreLuna. We report accuracy mean and standard deviation across five training runs for each setup. Values for the vanilla Transformer are taken from [5].

Model	Classification	Matching	ListOps
Transformer	64.27	57.46	36.37
Luna-256[1]	65.50 ± 0.21	79.44 ± 0.76	17.68 ± 0.38
Luna-8[1]	65.39 ± 0.20	72.00 ± 1.57	17.69 ± 0.26
Luna-1[1]	65.37 ± 0.22	72.66 ± 0.71	17.78 ± 0.79
PreLuna-256	65.41 ± 0.27	79.86 ± 0.47	18.11 ± 0.61
PreLuna-8	65.43 ± 0.25	74.34 ± 0.54	$21.72 \pm 8.57^{\dagger}$
PreLuna-1	65.10 ± 0.14	73.03 ± 0.86	35.15 ± 2.74

[1] Layer done according to the paper [4], full weight sharing between pack and unpack, [CLS] token as a sentence embedding.
† Highly dispersed values with significant outliers.

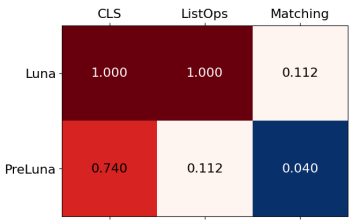


(a) Luna memory during training, step 6400 (b) Memory gradient, step 0

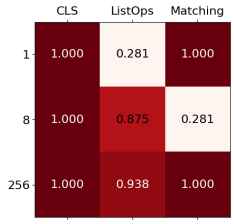
Figure 2: Base Luna memory when trained on the classification task. Each row corresponds to a single memory vector (i.e. query in a packing attention), the colorbar indicates raw values magnitude.

Memory Size Impact. An important observation we made is that memory cells tend to converge to a single value, or in rare cases to a small set of values, see Figure 2. This explains the very small differences in results between models of different memory sizes obtained by Luna authors and here (Luna-8 and Luna-1). We suggest that this issue is tied to the signal vanishing mentioned before, and that continuing output of similar vectors tends to yield similar memory gradients. These gradients, when being gradually accumulated during training, tend to eventually make memory cells similar to each other, making larger memory redundant.

We have conducted additional experiments with small memory sizes to support our claim about the inefficient memory use. Although the accuracy tends to drop in the unit memory setup (Luna-1), the models still consistently outperform the vanilla Transformer.



(a) Friedman test across all memory sizes with the H_0 = "expected accuracies are equal".



(b) Wilcoxon signed rank with the H_A = PreLuna > Luna.

Figure 3: Statistical tests p-values. The color coding are shades of blue for rejection on 5% confidence level, and of red if the null cannot be rejected. We apply the Holm p-value correction as in [1].

Paradoxically, for the PreLuna model decreasing the memory size leads to accuracy improvements in the ListOps task (see Table 1), while in the classification setup the positive effect from increasing memory rapidly diminishes.

Risk of Accuracy Degradation. Finally, we have found that given enough training steps, Luna training performance at some moment steeply degrades down to the random prediction accuracy. Observing the model weights reveals that they get affected in a similar way as the learnable memory.

5 Discussion, Conclusion and Future Work

During our preliminary work, we discovered that the potential of attention approximations done through shared memory is not yet fully revealed. While already showing solid performance on benchmarks, the models such as Luna suffer several issues which prevent them from achieving better input compression, and thus better results. Circumventing the discovered issues may not only allow to build faster and better models, but also provide an additional source of interpretability, namely the packed memory. Our proposed PreLuna with differently placed LayerNorm blocks is the first step towards that.

References

[1] Salvador García and Francisco Herrera, 'An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons', *Journal of Machine Learning Research*, **9**(89), 2677–2694, (2008).

[2] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and João Carreira, 'Perceiver: General perception with iterative attention', *CoRR*, **abs/2103.03206**, (2021).

[3] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh, 'Set transformer', *CoRR*, **abs/1810.00825**, (2018).

[4] Xuezhe Ma, Xiang Kong, Sinong Wang, Chunting Zhou, Jonathan May, Hao Ma, and Luke Zettlemoyer, 'Luna: Linear unified nested attention', *CoRR*, **abs/2106.01540**, (2021).

[5] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler, 'Long range arena: A benchmark for efficient transformers', *CoRR*, **abs/2011.04006**, (2020).

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, 'Attention is all you need', in *Advances in Neural Information Processing Systems 30*, Curran Associates, Inc., (2017).

[7] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu, 'On layer normalization in the transformer architecture', *CoRR*, **abs/2002.04745**, (2020).