Evaluating Attention Mechanisms Beyond Softmax in Long Contexts

Hamza Berqoq El Alami[1], Jibril El Hassani[1] | [1]CentraleSupélec

Introduction

While self attention has become the cornerstone of large scale deep learning applications, evaluating improvement directions for this mechanism has become a major area of interest. In this paper we focus on the Softmax :

- In self-attention, Softmax normalizes scores over the number of visible keys n.
- As context length grows, this normalization causes attention distributions to flatten (*attention fading*), even when relative scores are unchanged.
- This induces a length *distribution shift*: models trained at short context are evaluated in a different normalization regime at test time.
- The softmax function was not meant to be used on all layers of a neural network and is therefore relatively slow to compute.

Methodology

Recall that the self attention function reads :

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

With

$$softmax(X) = (\frac{e^{x_k}}{\sum_{i=0}^{n} e^{x_i}})_{k\in[|0,n|]}$$

To improve the mechanism, we evaluate different replacements for the softmax function :

- Scalable softmax (SSMax) [1] :

$$SSMax(X) = (\frac{n^{sx_k}}{\sum_{i=0}^{n} n^{sx_i}})_{k\in[|0,n|]}$$

. Where s is a scaling parameter. The intuition being that as context length grows, Softmax probabilities become mechanically smaller and less selective. SSMax amplifies score differences proportionally to $\log n$, restoring sharper attention distributions in long-context regimes.

- SimA : As in [2], we simply replace the key and query matrices by :

$$\hat{Q} = \frac{Q}{||Q||_1}, \hat{K} = \frac{K}{||K||_1}$$

instead of using softmax. This allows for faster inference as there is no exponentiation. It also avoids vanishing of attention.

- Element-wise sigmoid

$$sigmoid(X) = (\frac{1}{1+e^{-x}})_{k\in[|0,n|]}$$

The goal is to compare speed and representation with an element-wise nonlinearity.

We test these methods on Tiny Shakespeare text generation, text8 text generation for long context and CIFAR-10 image classification.

Our experiments follow the evaluation logic of Sections 3.1–3.3 of [1] using scaled-down models and lightweight datasets. We compare Softmax and SSMax during training from scratch and evaluate robustness to longer contexts without retraining, testing optimization stability and generalization as the number of keys n increases (Tasks 1–2). Finally, we design a synthetic needle-in-a-haystack retrieval task where a key token must be recovered from a long noisy context, directly probing attention dilution in long-context settings (Task 3).

Experimental evaluation on text8

Task 1 - Learning curve analysis

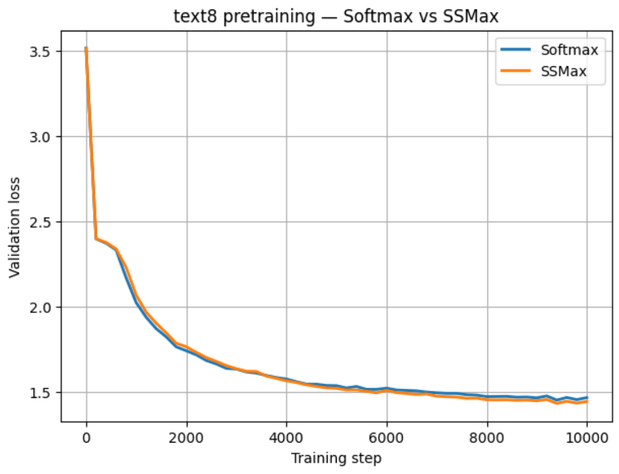

Fig. Task 1: Pretraining validation loss on text8. SSMax trains stably and reaches a slightly lower final loss than Softmax.

Task 2 - Length generalization

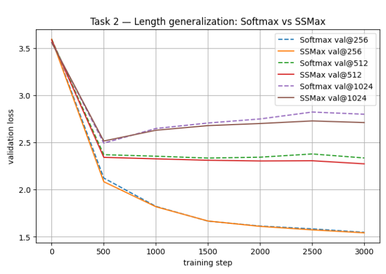

Fig. Task 2.A: Val loss evaluated at 256/512/1024 during training.

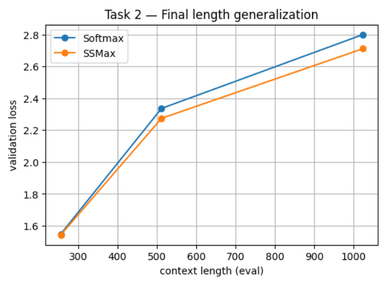

Fig. Task 2.B: Final val loss vs evaluation context length.

When evaluated on longer contexts, Softmax suffers from a strong degradation in validation loss. SSMax consistently mitigates this effect, with the performance gap increasing as context length grows.

Task 3 - Key information retrieval

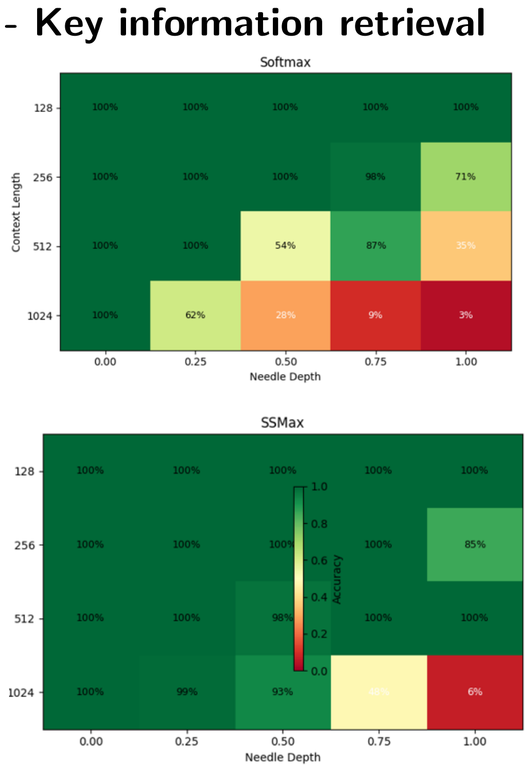

Figure: Retrieval accuracy heatmaps (Softmax vs SSMax). SSMax preserves accuracy for longer contexts and harder depths.

In a needle-in-a-haystack retrieval task, SSMax preserves higher retrieval accuracy than Softmax, indicating reduced attention dilution in long contexts.

Theoretical speedup

Activation	FLOPs	FLOPs diff	n = 32k
Softmax	$\frac{3n(n+1)}{2} - n$	/	15.3
SSMax	$\frac{4n(n+1)}{2} - n$	$+\frac{n(n+1)}{2}$	20.5
Sigmoid	$n(n+1)$	$-(\frac{n(n+1)}{2} - n)$	10.2
SimA	$\frac{n(n+1)}{2} - n$	$-n(n+1)$	5.1

Table: Comparison of the number of FLOPs of each activation function depending on the context length. The FLOPs diff column gives the differences between the activation function and the softmax activation. The last column gives the number of operations (in GFLOPs) for a context length of 32k

Practical speedup

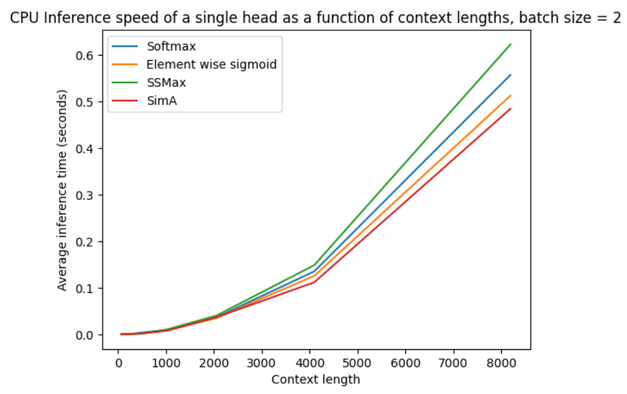

Fig. Inference time comparison of different attention mechanisms on sequences of 8192 token.

Representation learning

We compare qualitatively the representation of the sigmoid activation function on Tiny Shakespeare and SimA on CIFAR-10 to the Softmax function. The learned representations and model behaviors seem to vary widely when changing the activation function.

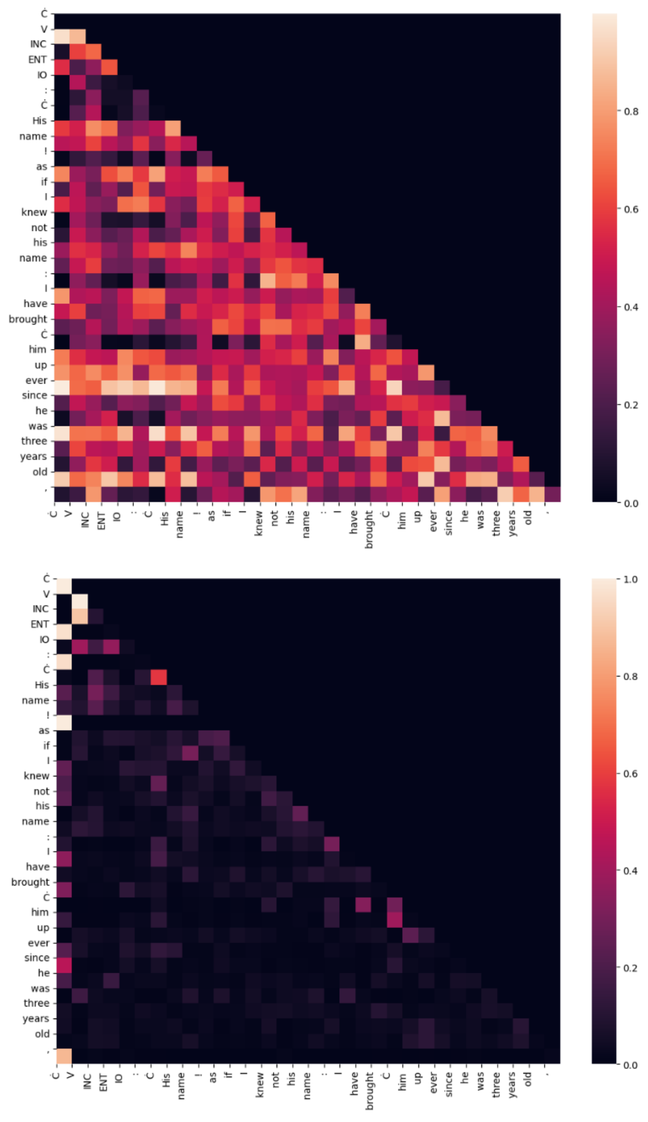

Figure: Comparison of the latent representation of sigmoid activation (top) vs softmax activation (bottom). The sigmoid activation is richer and does not have attention sinks

We see that the model is able to attend to all the tokens. While this is not necessarily the best setup, we hypothesize that it might be relevant on some tasks (e.g. long context or complex images).

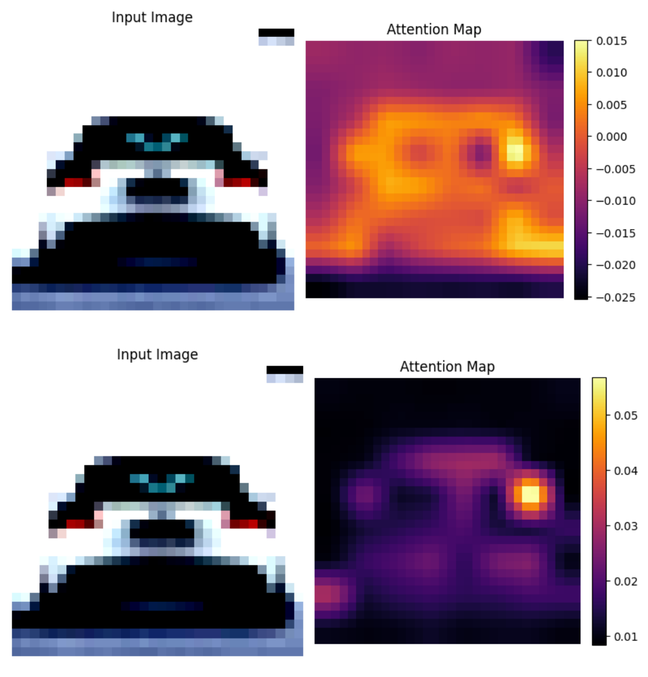

Figure: Comparison of the latent representation of SimA activation (top) vs softmax activation (bottom). We notice that the SimA model leverage negative attention and has stronger activations on regions of interest.

Conclusion

We benchmarked different replacements for the Softmax activation function and showed that SimA and SSMax are good alternatives to Softmax attention if the use case requires lower inference time or better long context retrieval performances. Furter exploration could include how the unique latent space of those activation functions could enhance performances on particular tasks.

References

- [1] Nakanishi, K. M. (2025). Scalable-softmax is superior for attention. arXiv preprint arXiv:2501.19399.
- [2] Koohpayegani, S. A., Pirsiavash, H. (2024). Sima: Simple softmax-free attention for vision transformers. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 2607-2617).