Linear Log-Normal Attention with Unbiased Concentration

Yury Nahshan, Joseph Kampeas, Emir Haleva

Distributed and Parallel Software Lab, Huawei Technologies

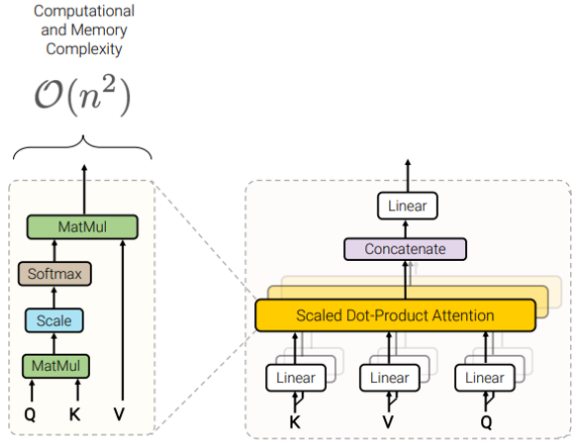
ICLR 2024

Introduction

Self-Attention layer allows model to learn connections between different tokens in the sequence.

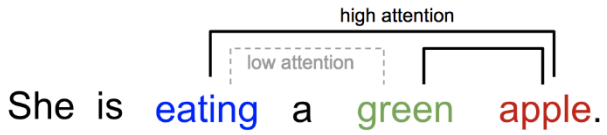
Scaled Dot-Product attention / SoftMax Attention (SA)

$$\text{Attn}(\boldsymbol{q}_i, \{\boldsymbol{k}_1, \ldots, \boldsymbol{k}_N\}, \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N\}) = \sum_{j=1}^{N} \text{softmax}\left(\frac{\boldsymbol{q}_i^\top \boldsymbol{k}_j}{\sqrt{d}}\right) \boldsymbol{v}_j^\top$$



✓ Effectively captures token interactions yielding contextual representation.

✗ Requires **quadratic** memory and computational complexity with respect to the sequence length.



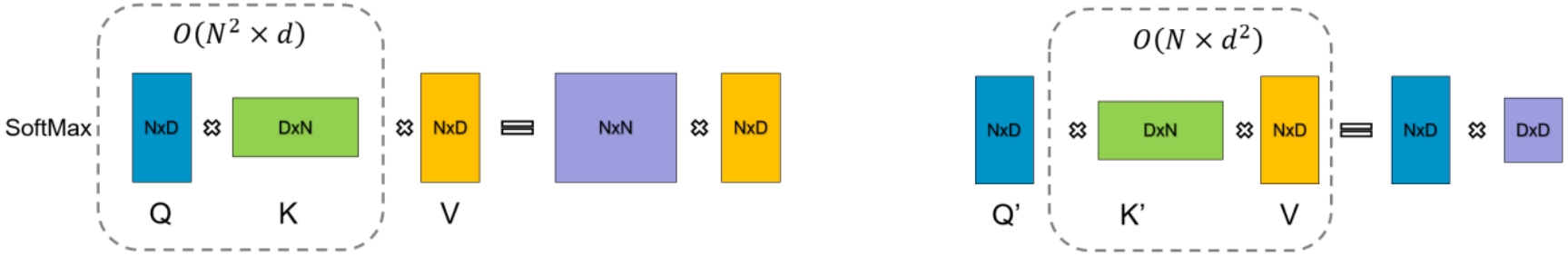
Linearized Attention - background

Linearized Attention (LA) suggests a Kernel-based approach to decompose computation of attention matrix.

Kernel-based Attention requires selecting a feature embedding function ϕ to compute the LA kernel.

$$\mathrm{Attn}_{\mathrm{lin}}(\boldsymbol{q}_i, \{\boldsymbol{k}_1, \ldots, \boldsymbol{k}_N\}, \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N\}) = \frac{\sum_{j=1}^{N} \Phi(\boldsymbol{q}_i)^\top \Phi(\boldsymbol{k}_j)}{\sum_{l=1}^{N} \Phi(\boldsymbol{q}_i)^\top \Phi(\boldsymbol{k}_l)} \boldsymbol{v}_j^\top$$

- Requires **linear** memory and computation complexity.
- Often underperform compared to standard self-attention.



(a) Softmax attention (b) Linear attention

Related work

Different works suggested different kernel-based solutions for linearized attention

- **Linear Transformer** *[A. Katharopoulos, 2020]*
  - Propose to use a simple feature map $\phi(x) = elu(x) + 1$. Authors empirically show their method to perform on par with the standard attention on simple tasks.
- **Performer** *[K. Choromanski, 2020]*
  - Propose random feature map $\phi(x) = e^{\omega^T x - \frac{\|x\|^2}{2}}$. Authors show that for ω drawn from Gaussian distribution it roughly approximates the SoftMax scoring function. Authors also explore $\phi(x) = ReLU\left(\omega^T x - \frac{\|x\|^2}{2}\right)$ function and show its effectiveness for various tasks
- **Random Feature Attention** *[H. Peng, 2021]*
  - Authors propose Random Fourier Features (RFF) approximation for the SoftMax based attention.
  - However, our experiments shown it does not scales well to larger models especially when trained in low precision formats like fp16.
- **cosFormer** *[Zhen et. El., 2022]*
  - Authors explorer various functions for feature map (Leaky ReLU, ReLU and Softmax)
  - In addition suggest linear feature map embedding and novel scheme of non-linear cosine re-weighting for the queries and keys. Their method achieves SOTA results on Roberta model.

However, no one yet studied the connection between feature embedding function and concentration of attention.

Method overview

We propose a systematic way to develop the LA method which has comparable performance to the SA.

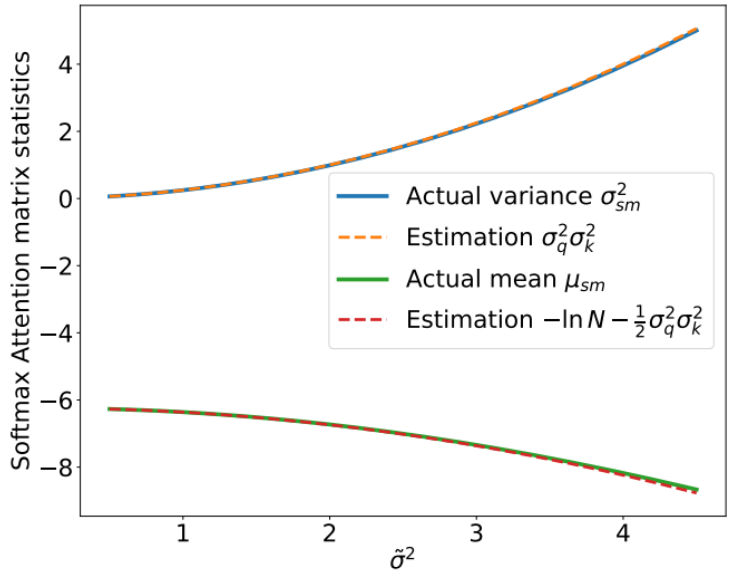
- First, we define an analytical model of the SA layer.
- Conduct an in-depth analysis of this model, characterizing its statistical, informational, and algebraic properties.
- Build tools to analyze concentration behavior of the attention and apply them to SA.
- Using the model and proposed tools, we designed Linear Log-Normal Attention (LLN Attention) which has linear memory and computational complexity in the sequence length and comparable performance to the SA.

Model definition

- Model assumptions:
  - We assume inputs are Gaussian vectors $q_{\mathrm{i}}, \mathrm{k_j}$ where elements have zero mean and variances σ_q^2, σ_k^2
- Denote
  - σ_{sm}^2 the variance of the SA matrix
  - $a_{ij} = \frac{\langle q_i, k_j \rangle}{\sqrt{d}}$ then, its normalized version $\tilde{a}_{ij} = \frac{a_{ij}}{\sigma_{sm}}$
- Define temperature of the SoftMax Attention $\tau_{\mathrm{sm}} = \frac{1}{\sigma_{\mathrm{sm}}} = 1/\sqrt{\sigma_q^2 \sigma_k^2 + C_{\mathrm{cross}}}$
- The SA matrix can be represented as $P_{ij}^{(\mathrm{SM})} = \frac{e^{\tilde{a}_{ij}/\tau_{\mathrm{sm}}}}{\sum_{l=1}^{N} e^{\tilde{a}_{il}/\tau_{\mathrm{sm}}}}$

Analysis of the SoftMax Attention

- We show that SA matrix P^{sm} follows a log-normal distribution with moments:
  - $\sigma_{sm}^2 = \sigma_q^2 \sigma_k^2$
  - $\mu_{sm} = -\ln(N) - \frac{1}{2}\sigma_q^2 \sigma_k^2$



- The log-normal distribution helps us to understand the concentration ability of the SA.

- *Claim: The entropy of the SA is monotonically increasing with temperature. Further, the spectral gap $\gamma = 1 - |\lambda_2|$ is increasing with temperature.*

Linear Log-Normal (LLN) Attention

The LLN Attention matrix

$$P_{ij}^{(\mathrm{LLN})} = \frac{e^{\alpha \boldsymbol{q}_i^\top} e^{\beta \boldsymbol{k}_j}}{\sum_{l=1}^{N} e^{\alpha \boldsymbol{q}_i^\top} e^{\beta \boldsymbol{k}_l}}$$

- Feature embedding functions $\Phi(q) = e^{\alpha q}$, $\Phi(k) = e^{\beta k}$
- The hyperparameters α, β allows to match the concentration of LLN Attention with that of SA
- The LLN Attention also follows a log-normal distribution similarly to the SA

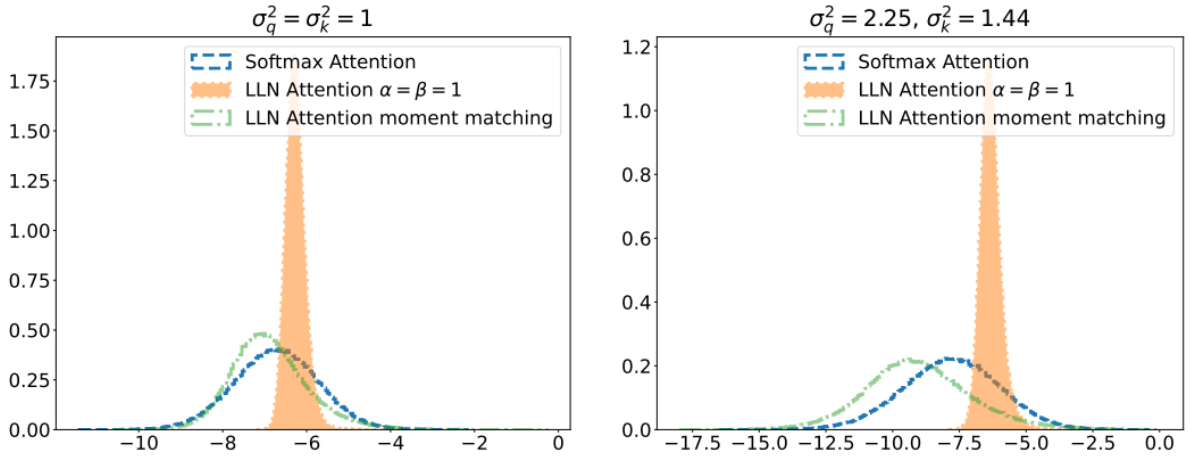


Figure 2. Histogram of the SA and LLN Attention with and w/o moment matching.

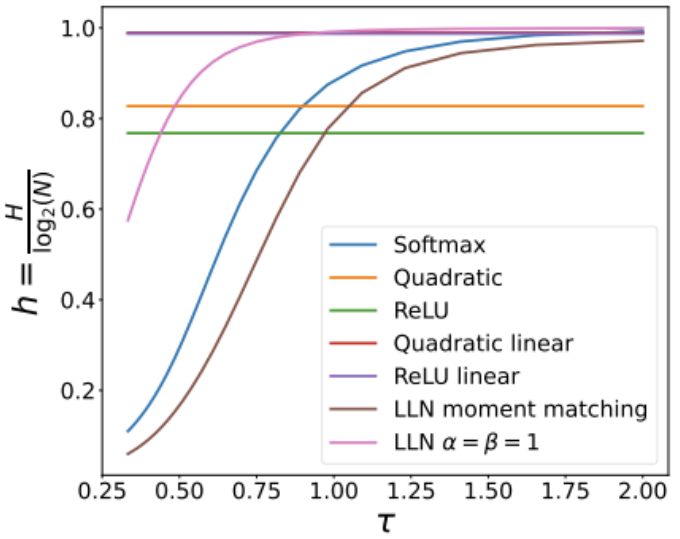
LLN Attention moment matching

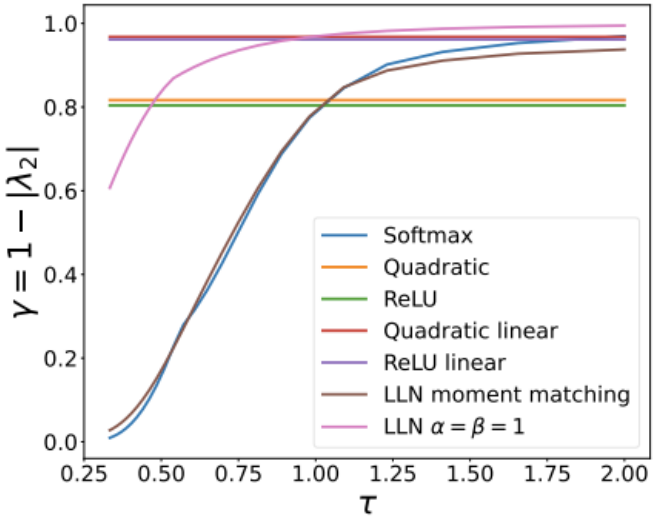
- Finding appropriate values hyperparameters α, β is crucial to achieve required concentration

- We requiring $\sigma_{lln} = \sigma_{sm}$ and performing linear interpolation on random Gaussian samples α, β.

$$\alpha = \frac{\tilde{\sigma}}{\sqrt{2}\sigma_q}; \quad \beta = \frac{\tilde{\sigma}}{\sqrt{2}\sigma_k}; \quad \tilde{\sigma} = \sqrt{\frac{1}{a}(\sigma_q^2\sigma_k^2 - b)}$$

- The entropy and the spectral gap of the LLN Attention with moment matching is similar to those of the SA.

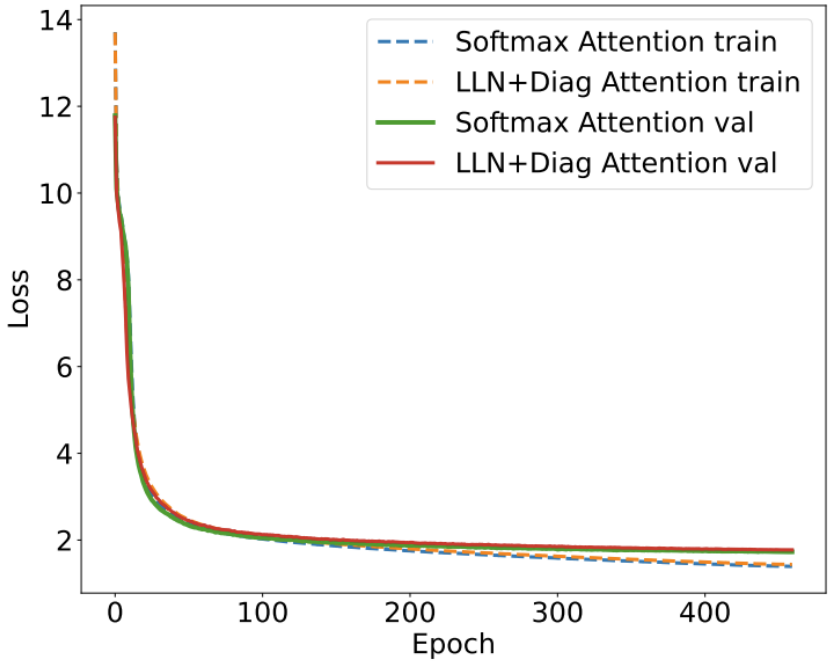




Experiments on NLP task

- We validate LLN Attention on two phases of the NLP tasks:
  - First, on pre-train the bidirectional RoBERTa encoder model on wikitext-103.
  - Second, we fine-tune our pretrained model on GLUE dataset.

- The LLN Attention method outperforms other LA methods with an average accuracy of 86.9%.



Method	MNLI	QNLI	QQP	SST-2	Avg ↑
SA baseline (Bahdanau et al., 2015)	80.3	87.2	89.9	90.6	87.0
Reformer (Kitaev et al., 2020)	35.4	-	63.2	50.9	49.8
Performer (Choromanski et al., 2020)	58.8	63.4	79.1	81.4	70.6
ELU (Katharopoulos et al., 2020)	74.8	82.5	86.9	87.2	82.8
Longformer (Beltagy et al., 2020)	77.2	-	85.5	88.6	83.7
Transformer LS (Zhu et al., 2021)	77.0	84.8	86.8	90.2	84.7
TNN (Qin et al., 2023)	76.72	85.06	88.3	90.6	85.17
T2 (Qin et al., 2022b)	77.28	85.39	88.56	90.71	85.48
CosFormer (Qin et al., 2022a)	76.7	-	89.2	91	85.6
T1 (Qin et al., 2022b)	79.06	87.0	88.61	91.17	86.46
Flash (Hua et al., 2022)	79.45	**87.1**	88.83	90.71	86.52
Nyströmformer* (Xiong et al., 2021)	80.9(-1.5)	88.7(-1.6)	86.3(-1.)	91.4(+1.4)	86.8(-0.7)
LLN Attention (Ours)	77.0	85.1	88.9	90.6	85.4
LLN+Diag Attention (Ours)	**80.0**	86.5	**89.7**	**91.6**	**86.9**

Note that methods marked with $*$ have a higher baseline in the original paper, which may lead to superior results.

Thanks