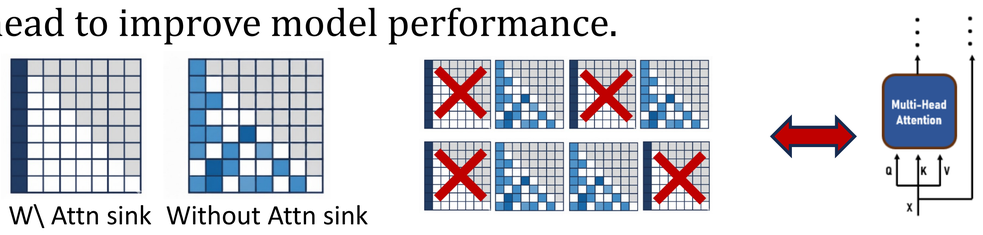
電機資訊學院 2026 作 BRAIN PLUS HAND 實作專題競賽

Towards Compute-Aware Attention: A Skip Mechanism for Redundant Transformer Heads

EECS02 游竣閔 Chun-Min Yu

Abstract

We propose an **attention skip mechanism** that detects and bypasses not only redundant, but harmful heads during attention computation in Transformer. Experiments on NLP benchmarks show that a significant portion of attention computations are redundant, and that reducing redundant head contributions **improves performance** on GPT-2. Our findings clarify attention sink dynamics and demonstrate a practical approach to **compute-aware attention**, enabling potential training-time early stopping per head to improve model performance.



Method

Motivation

Multi-head self-attention often exhibits **sink behavior**, where the attention distribution for each row collapses onto a single column for a large number of query positions. **Tokens in these "sink" columns tend to have low-norm value vectors** $\| v_j \|$ .As a result, even when the attention weight α_{ij} is large, **the effective term** $\alpha_{ij} v_j$ **contributes little to the output** o_i .Furthermore, since the softmax mass focuses on the sink, the remaining probability mass available for other tokens is small. This means that the **aggregate term** $\sum_{j \neq j^*} \alpha_{ij} v_j$ **is also negligible**, even if some of these value vectors contain valuable information.

We assume that the heads displaying this behavior are essentially attempting to perform a "**no-op**", meaning their **contribution to the final output is nearly zero,** but not 0 due to the limitation of softmax, making the attention weight nearly impossible to gather all their attention at a specific token.

Setting

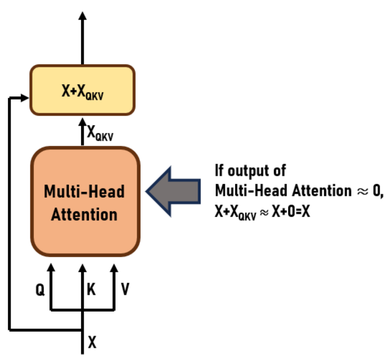
Consider a Transformer block with H heads operating on input $x \in R^{B \times T \times C}$ .Each head forms

$$Q, K, V \in R^{B \times H \times T \times d_h}, \quad d_h = \frac{C}{H},$$

and computes causal attention

$$A = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_h}} + M\right) \in R^{B \times H \times T \times T}$$

followed by $Y = AV$ and the standard output projection and residual addition.



Sink Indicator

We diagnose sinkness inside each head after masking and softmax by measuring *row-wise peakedness*. For each query position i, define

$$\text{row\_max}(i) := \max_j A_{ij}$$

.A row is deemed *peaked* if $\text{row\_max}(i) > \tau_{\text{peak}}$(default 0.95).

A head is classified as *sinky* for a given sequence when

$$\frac{(\#\text{ peaked rows})}{(\#\text{ rows})} > p_{\text{head}} \text{ (default 0.8).}$$

At each step, for each batch item and head, we evaluate this *peaked*-row ratio; if it exceeds p_{head}, the head is skipped. **By skipping these heads, the model is able to perform "no-op" operations on them, a capability that is nearly impossible to achieve with the original architecture design due to the flaw of softmax**.

Experiment & Result

The experiment are conducted using plain GPT2 as baseline model ,and modified-GPT2 as our target model.

$\tau_{\text{peak}} = 0.95$, $p_{\text{head}} = 0.8$

Model	SST-2	MNLI-m	MNLI-mm	QNLI	QQP (F1/Acc)	IMDb	SQuAD v1.1	SQuAD v2.0
GPT2	92.9	82.3	81.9	89.7	69.5/88.0	92.04	83.95	72.51
GPT2 mod	**93.2**	**82.5**	**82.7**	**90.0**	**69.5/88.5**	**93.20**	**84.15**	**72.78**

The results demonstrate that the performance of the modified model outperforms the baseline model not only on classification tasks but also on generation tasks, such as SQuAD v1.1, and even the harder version, SQuAD v2.0, which requires the model to respond with "no answer" when no valid answer is present. These findings provide strong evidence that **our modification enhances performance across a variety of tasks with different difficulties**, rather than improving performance in a specific area.

	SST-2	MNLI	QNLI	QQP
Layer 0–3	0.10%	0.00%	0.00%	0.00%
Layer 4–7	3.54%	2.90%	2.14%	5.50%
Layer 8–11	4.28%	0.75%	3.20%	3.18%
Layer 12–15	3.82%	3.40%	4.03%	2.09%
Layer 16–19	6.52%	4.57%	8.03%	2.81%
Layer 20–23	6.73%	13.9%	7.59%	11.42%

Token Type	Value vector L2 norm
Sink tokens	**0.89**
Random tokens	7.35
Δ_v	-6.46

As shown in the table above, **the L2 norm of the sink token's value vector is significantly smaller than that of other tokens**, validating our initial assumption regarding the design. Additionally, we observed that attention tends to concentrate in the deeper layers, which are responsible for learning high-level semantic structures, rather than in the shallower layers that capture low-level features.

Conclusion

We introduced an *attention skip mechanism* that detects attention sinks and selectively bypasses heads whose behavior is effectively near no-op. The method is model-agnostic, and compatible with standard Transformer implementations.

Performance test for different combinations of threshold and test on other architectures are yet to be done. Insert a zero-value token into per sequence to let the model dynamically adjust to "bypass the head" instead of setting a hard threshold combining with hard-clipping softmax is also an idea that I will try to implement in the future.