A Unified View for Attention and MoE

Taiqiang Wu[◇] **Ngai Wong**[◇]
[◇]The University of Hong Kong

Abstract

This report presents a unified view of the attention mechanism and Mixture of Experts (MoE), highlighting their structural similarities. Both mechanisms compute a weighted sum over a set of candidates—tokens for attention and experts for MoE—based on dynamically computed probability distributions. We formalize this similarity as $\sum f(x) \cdot g(x)$, where $f(x)$ represents weights and $g(x)$ represents values to aggregate. While attention captures token-to-token interactions, MoE focuses on input-to-expert interactions, with MoE typically activating only a subset of the experts. We also discuss advancements like sparse attention and enhanced MoE routers, which further bridge the gap between these paradigms. By unifying these concepts, we aim to foster a deeper understanding of their interplay and inspire further innovation in these mechanisms.

1 Preliminary

Attention (Vaswani, 2017) mechanism has been widely used in the Transformer, which shows promising abilities in natural language processing (NLP) and computer vision (CV) tasks. At the same time, Mixture of Experts (MoE) (Fedus et al., 2022; Bi et al., 2024) has emerged as a popular solution for larger LLMs, enabling parameter scaling while maintaining computational efficiency.

In this report, we provide a unified view of attention and MoE, viewing them as mechanisms that compute a weighted sum over a set of candidates (tokens or experts) based on a dynamically computed probability distribution.

1.1 Attention Mechanism

Let $\mathbf{X} = \{x_i\}_{i=1}^{M} \in \mathbb{R}^{M \times d}$ be the representations of the sequential tokens $x_i \in \mathbb{R}^{1 \times d}$ for $i = 1, 2, ..., M$. For each token x_i, the Attention mechanism computes a weighted sum over all tokens in the sequence:

$$y_i = \sum_{j=1}^{M} a_{ij} \cdot v_j \tag{1}$$

where we have:

- $v_j = x_j W_v \in \mathbb{R}^{1 \times d}$ is the value vector for token x_j, with $W_v \in \mathbb{R}^{d \times d}$ being a learnable weight matrix.

- a_{ij} is the attention coefficient between token x_i and token x_j, computed as:

$$a_{ij} = \frac{\exp(q_i k_j^\top)}{\sum_{j'=1}^{M} \exp(q_i k_{j'}^\top)}. \tag{2}$$

  Here, $q_i = x_i W_q$ is the query vector for token x_i, and $k_j = x_j W_k$ is the key vector for token x_j, with $W_q, W_k \in \mathbb{R}^{d \times d}$ being learnable weight matrices.

The attention coefficients a_{ij} represent the importance of token x_j to token x_i, and the output y_i is a context-aware representation of token x_i. In summary, the attention mechanism first calculates a probability distribution over tokens and then aggregates the information.

1.2 Mixture of Experts (MoE)

For the given input x_i, the MoE mechanism computes a weighted sum over a subset of experts selected via a Top-K operation (Fedus et al., 2022) or ReLU operation (Wang et al., 2024). Let N be the total number of experts. For token i, MoE first computes the router scores for all experts:

$$r_{i,l} = \text{Router}(x_i) = x_i W_l \quad \text{for } l = 1, 2, \ldots, N \tag{3}$$

where $W_l \in \mathbb{R}^{d \times N}$ and $r_{i,l}$ denote the score of token i towards expert l. Then, we define the Top-K

binary mapping function $\mathbf{m} = [m_1, m_2, \ldots, m_N]$, where:

$$m_{i,l} = \begin{cases} 1 & \text{if expert } l \text{ is in the Top-K experts,} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Therefore, we can compute the activation coefficients for the selected experts using the softmax function:

$$p_{i,l} = m_{i,l} \cdot \frac{\exp(r_{i,l})}{\sum_{l'=1}^{N} m_{i,l'} \exp(r_{i,l'})} \tag{5}$$

Meanwhile, if we replace the softmax+Top-K with ReLU function(Wang et al., 2024), it equals

$$p^*_{i,l} = ReLU(r_{i,l}) \tag{6}$$

Finally, we can compute the weighted sum over the outputs of the selected experts:

$$y_i = \sum_{l=1}^{N} p_{i,l} \cdot h_l(x_i) \tag{7}$$

where $h_l(x_i)$ is the output of expert l for input x_i.

2 Unified View

2.1 Formulation

Comparing (1) and (7), we can conclude that both attention and MoE share a similar structure:

$$\sum f(x) \cdot g(x), \tag{8}$$

where $f(x)$ are the weights and $g(x)$ are the values to aggregate. For the attention mechanism, $f(x)$ is the attention distribution among input tokens, and $g(x)$ is the linear projection of the input. For the MoE, $f(x)$ is the learned weights among experts, and $g(x)$ is the outputs of experts.

2.2 Similarity

From a single-input perspective, both attention and MoE share the following structure.

Coefficient Computation. In Attention, the coefficients a_{ij} are computed using a softmax over the dot product of queries and keys, which are derived from the input tokens x_i and x_j. In MoE, the coefficients p_l are computed using a softmax over the router's scores, which are derived from the input x, but only for the Top-K selected experts (as indicated by the binary mapping m_l). Recently, ReMoE (Wang et al., 2024) proposes to calculate the coefficients using ReLU to replace the Top-K selection.

Weighted Sum. In Attention, the weighted sum is over the values v_j corresponding to tokens x_j. In MoE, the weighted sum is over the outputs $h_l(x_i)$ of the Top-K selected experts. Both mechanisms employ the weighted sum operation.

Dynamic Selection. Both mechanisms dynamically select relevant candidates (tokens or experts) based on the input.

2.3 Differences

The key differences lie in the scope of the candidates and the selection mechanism.

Scope. In Attention, the candidates are tokens within a sequence, and the coefficients capture token-to-token interactions. In MoE, the candidates are experts, and the coefficients capture input-to-expert interactions.

Selection Mechanism. In Attention, **all** tokens are typically considered (though sparse attention variants exist). In MoE, only the Top-K experts are activated, as enforced by the binary mapping $m_{i,l}$. In ReMoE, a sparsity loss is proposed to ensure only a subset of the experts are activated.

2.4 Rethinking Variants

Given the unified view, we can rethink the variants of attention and MoE.

The sparse attention (Child et al., 2019; Beltagy et al., 2020; Xiao et al., 2023) is quite similar to the MoE router. In sparse attention, the model dynamically selects a subset of tokens to attend to based on the positional priority, rather than computing attention scores for all possible pairs of tokens. This is particularly useful in handling long sequences, where the quadratic complexity of full attention becomes prohibitive. Similarly, in MoE architectures, the router dynamically assigns input tokens to a subset of expert networks, each specialized in processing different types of inputs. This selective processing allows the model to scale efficiently with larger datasets and more complex tasks.

In MoE, the router can be enhanced by aggregating information from other tokens (Wu et al., 2024), following the same idea as attention. In Yuan 2.0-M32 (Wu et al., 2024), an attention router is designed to consider the correlation between experts, resulting in higher accuracy compared to the classical router structure.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Ziteng Wang, Jianfei Chen, and Jun Zhu. 2024. Remoe: Fully differentiable mixture-of-experts with relu routing. *arXiv preprint arXiv:2412.14711*.

Shaohua Wu, Jiangang Luo, Xi Chen, Lingjun Li, Xudong Zhao, Tong Yu, Chao Wang, Yue Wang, Fei Wang, Weixu Qiao, et al. 2024. Yuan 2.0-m32: Mixture of experts with attention router. *arXiv preprint arXiv:2405.17976*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.