MoE-Mamba: Efficient Selective State Space Models with Mixture of Experts

Main Idea: the goal of this work is to compare vanilla MoE (Transformer MoE) with vanilla Mamba and MoE-Mamba to explore if these architectures are compatible with each other. The main highlight of this paper is that MoE-Mamba outperforms both Transformer-MoE and vanilla-Mamba, reaching the same performance of vanilla-Mamba with 2.2x less training steps, while preserving Mamba inference gains over the Transformer. This shows that MoE results in performance gains when combined with the Mamba architecture, similarly to when applied to Transformers. In theory, this should result in easier scaling for Mamba, with even more inference gains due to the sparsity of MoE.

Mamba

Mamba is an SSM (State-Space Model) architecture (SSM meaning that it is based on hidden states that update and drop/forget irrelevant information) like RNNs, GRUs and LSTMs.

- Mamba is an improvement over previous SSM architectures because it is optimized for GPUs and can make use of parallelism.
- Mamba is an improvement over Transformers because the characteristic of dropping irrelevant info of SSM architectures allows for a much lesser complexity as the input size increases. In theory, this should result in increased quality and reduced inference costs for Mamba compared to Transformers when scaling the context length.

    - Transformers' complexity increases quadratically with an increase in input size (O(n^2)). Mamba does not impose this constraint.

MoE-Mamba Architecture

- MoE-Mamba makes use of a similar architecture to Switch Transformer.
    - Token-choice routing (top-k) with k=1 (one expert used per token)
    - Every other Mamba layer is replaced with an FF MoE (each block alternates between dense (Mamba) and sparse (Mamba MoE) layers).
- The active parameters of the models experimented with were ~26M per token.
    - The total number of parameters of the biggest MoE-Mamba model used was 416M parameters (32 experts).
- MoE-Mamba scales well with an increase in the number of experts (expert size was constant, so increasing the number of experts means increasing the number of total parameters while keeping the number of active parameters constant). The largest number of experts experimented with was 32.
    - MoE-Mamba needed at least 8 experts to improve over vanilla Mamba.

My takeaways:

- Mamba's main advantage over Transformers seems to be of the handling of large context lengths due to SSM architectures inherently having the ability to drop irrelevant info from

token to token. This is not true for the attention process in the Transformer architecture, which has an exponential increase in complexity with an increase in the context length.

    - The main questions about Mamba's legitimacy today are:
        - How will Mamba scale in terms of increasing parameter size and data?
        - Will Mamba work given huge context lengths (tens/hundreds of thousands of tokens)?
    - More research on the Mamba architecture is needed on my end.

- More research on Mamba-MoE needs to be done at increased parameter scales. A 416M parameter model with 26M active parameters per token is too small. Thus, the results of this paper should be seen as a mere indication and be taken with a grain of salt.