OpenMoE: An Early Effort on Open Mixture-of-Experts Language Models

Main Idea: an open-source project, OpenMoE analyzes decoder-only MoE LLMs from 650M to 34B total parameters and trained on up to 1T tokens (the largest version – 34B – was only trained on 200B tokens, 1T training tokens were used in the 8B version). The findings and recommendations of these experiments are shared in the paper.

The 34B version (largest one trained) has 6B active parameters per input and 32 experts per MoE layer. 5 intermediate checkpoints for the 8B model (every 200B training tokens) were released, and a Chat version of this 8B model was also trained (instruction-tuned).

Design

- Inspired by the facts that including code data in the pre-training dataset boosts performance and that code is always precise (contrary to text), which leads the authors to think that LLMs would more easily understand it, leading to better training, code data is aggressively sampled during pre-training (excessively/to a fault).
- Generally, follows the ST-MoE architecture and routing design. The reason for this is ST-MoE's focus on training stability, a characteristic OpenMoE aims to achieve.
- Top-2 routing used during the entire training process.
- An MoE layer is inserted every 6 Transformer blocks, so most Transformer blocks do not have an MoE layer.

- Use UL2 method for the training objective (mix of span corruption and prefix language modeling).
- SFT for instruction-tuning is done on a dataset of 58k conversations, each with 1.8 turns on average, to analyze alignment (although this is not a big focus of this work).

Analysis

- MoE experts did not seem to specialize at the domain or at the task levels, but at the token level.
    - This is intuitive and rather obvious since the routing is done at the token-level.
- Context-independent specialization
    - MoE routing is done based on token ID and independent of the context around that token. This means that the routing is not really done based on semantics (context) but on syntax (the token being routed).
- Experts cluster tokens together, that is, they seem to specialize on a specific cluster of the token input space (the raw token's embeddings without regard to context). Similar tokens are routed to the same expert.
- The token routing is learned at very early stages of training and remains fixed throughout the rest of training.
- Drop-Towards-the-End
    - Due to this fixed routing characteristic, something like instruction-tuning can lead to issues. This is because instruction-tuning data is out-of-domain, presenting a

distribution shift from the pre-training data. Since the routing is learned from the pre-training data and is fixed, the distribution shift from instruction-tuning data will lead to overloaded experts, subsequently leading to token dropping in later rounds of the conversation (assuming multi-turn chat).

Takeaways/Recommendations

- The amount of code present in the pre-training data of over 50% was too aggressive (around 30% is recommended instead) and hurt the performance of the model in text tasks.
- The finding that MoE routing is fixed and established at early stages of training indicates that the router can be frozen after a warmup stage.
- The Context-Independent Specialization of experts indicates that the FFN (expert) computation can be done independently from the attention layer, thus an approach that would compute the expert FFN and the attention layers in parallel would make sense, bringing a speedup in training and inference.
    - Future research proposition.
- To alleviate the Drop-Towards-the-End issue, mixing instruction-tuning data into the pre-training data mix while the routing is being learned (the warmup stage) can be effective. This would allow the router to learn the instruction-tuning data distribution, so the token dropping issue experienced in later rounds of multi-chat conversation would be somewhat mitigated.

My takeaways:

- 5 checkpoints for the 8B OpenMoE model were released. This could potentially add to the routing analysis project I have planned.
- The conclusion that experts specialize on a specific cluster of the token input space seems to be inconsistent with the Hash Layers paper comparison of cluster-based hashing vs the opposite.
- The conclusion that token routing is fixed at very early stages of training seems to be inconsistent with the analysis done in the StableMoE paper.