Efficient Large Scale Language Modeling with Mixtures of Experts

Main Idea: this paper has the goal of comparing how the traditional MoE architecture from "Sparsely-Gated MoE", using top-2 routing, scales in relation to dense models.

Model sizes trained for this experiment range from (in total number of parameters):

- 125M to 13B (in a dense setting).
- 15B to 1.1T (in a MoE setting).

The maximum number of experts used was 512, and the capacity factor used for MoE models was 2 (to support top-2 routing).

Dense and sparse models were compared on a FLOPs-matching basis (models with the same FLOPs are comparable). The dense baseline used was GPT-3.

Evaluations done:

- Perplexity (from next-token predictions).
- Performance on downstream tasks (benchmarks, zero-shot, few-shot).
- MoE speedup factor – how much more efficient MoEs are at achieving a specific performance level relative to dense models (how many training FLOPs are needed to reach a certain performance goal).

Results:

- MoE outperforms dense in all evaluation datasets, although at a different scale depending on the dataset's domain and model size.

- MoEs are the most efficient (highest speedup factor in in-domain tasks), reaching an 8x to 16x speedup (8x-16x less compute needed for the same performance)
    - This speedup decreases to a 2x-4x speedup in out-of-domain tasks.
- The speedup advantages of MoE decrease at scale, especially in in-domain tasks.
- The closer the data used for evaluation is to the training corpus, the larger the speedup obtained by MoE.
- On downstream zero-shot task evaluation, MoE also outperforms the dense model (which performs on par with GPT-3), but this gain is, again, diminishing at scale.
- In a few-shot setting, MoE still outperforms dense, but the MoE improvements over zero-shot are smaller than dense. This indicates that although MoE still outperforms dense in a few-shot setting, dense models benefit more from few-shot examples.
- In terms of fine-tuning, dense models (as expected) always incur substantial gains. Although this is true in some cases for MoE, fine-tuning MoE models on some domains/datasets leads to worse performance. More research is needed to determine why. Perhaps this comes from fine-tuning MoEs not being deeply explored yet, with an alternative approach needed to obtain good results (the same setting as dense was used for fine-tuning after all).

<u>My takeaways</u>:

- The results from this paper's experiments show that the traditional MoE architecture does indeed provide speedups over a dense setting. The results from the speedup

provided by MoE are bigger the closer the evaluation domains are from the training domains. This seems to indicate that the biggest gains from MoE come from memorization. Generalization gains provided by MoE over dense are not as apparent, although there still are gains (MoE still provides a speedup when evaluated in out-of-domain tasks).

    - The diminishing gains from MoE at scale are more apparent in out-of-domain tasks, as they stay relatively constant when training domains (or close to) are used for evaluation.

- It is interesting to note that few-shot has a bigger effect on dense performance than on MoE performance (dense benefits more), although MoE outperforms dense in this scenario.
- A previous work, ST-MoE, concludes that sparse models benefit from smaller batch sizes and larger learning rates during fine-tuning, while the opposite is observed for dense models. ST-MoE also concludes that MoEs are significantly more prone to overfitting during fine-tuning compared to dense. The fine-tuning results from this paper can be replicated and analyzed with these two aspects in mind as future research.