Mixture-of-Experts Meets Instruction Tuning: a Winning Combination for Large Language Models

Main Idea: this study aims to measure the impact of instruction-tuning in MoE models compared to its impact in dense models.

Instruction-tuning is related to fine-tuning as fine-tuning is training a pre-trained model on a specific task, while instruction-tuning consists of training a language model in a supervised manner to perform well in a dialogue setting. This means for the model to perform well on the task of predicting p(answer | question) instead of the pre-training objective of predicting p(word | context).

Three different scenarios were evaluated:

- Direct finetuning on individual tasks (no instruction tuning).
- Instruction tuning followed by in-context learning (no direct fine-tuning)
- Instruction tuning followed by further finetuning on individual tasks.

The conclusion of this paper was that MoE models outperform dense models of equivalent computational capacity on direct finetuning, but significantly outperform dense models on instruction tuning scenarios. Let's understand how they reached this conclusion.

Setup

Two dense models were considered: T5 and PaLM.

Four MoE architectures were considered:

- Switch Transformers
- GShard
- Expert-Choice
- ST-MoE

All instruction tuning was done using the FLAN dataset.

Results

- A base MoE architecture outperforms a dense architecture (T5) after instruction-tuning across all scales.
- Scaling the number of experts helps when fine-tuning on challenging tasks but saturates when fine-tuning on easier tasks (more experts is not always better as it might confuse the gating algorithm).
- As expected, increasing k in top-k routing improves performance at an increase in the inference cost.
- Overperformance of MoE compared to dense models when instruction-tuning only exacerbates with scale (the bigger the models, the bigger the performance gain of MoE over dense).

- Expert-choice outperforms GShard (token-choice) in an instruction-tuning scenario, however, this difference is bridged by incorporating advanced auxiliary loss (router z-loss) and pre-training strategy as employed in ST-MoE (also token-choice).
- Even though FLAN-PaLM62B (dense instruction-tuned model) has 3x the number of FLOPs per token than FLAN-ST32B (largest MoE instruction-tuned model trained for this work) at inference, FLAN-ST32B significantly outperforms FLAN-PaLM (57.6 vs 63.6 average score).
- Different auxiliary losses gave different results:
    - Z-loss worked better than balance-loss in FLAN-ST
    - Balance-loss worked better than z-loss in FLAN-EC
- Freezing certain parts of the MoE layers during fine-tuning was evaluated to investigate how to prevent overfitting in MoE fine-tuning:
    - Freezing the gate led to small improvements.
    - Freezing any other areas resulted in worse performance.

My takeaways:

- First thought is that instruction-tuning should work better in dense models than in MoE models based on the difficulties in obtaining good fine-tuning performance with MoE. This may not hold since the instruction-tuning process can be thought of a very specific type of fine-tuning.
    - This is shown to be false, as MoE significantly outperforms dense models when it comes to instruction-tuning. This is even more interesting when showed that this

advantage of MoE over dense in the task of instruction-tuning only increases with scale.

    - MoE results after instruction-tuning are quite promising. For some reason, MoE captures the instruction-tuning task much more efficiently than dense models.

- More experts do not guarantee better performance with fine-tuning. In fact, on easier tasks, more experts result in worse fine-tuning performance.
    - What was the size of the datasets used for fine-tuning? Perhaps easier tasks are more prone to overfitting, explaining the underperformance of fine-tuning MoE on easier datasets. If this was the case, these tasks would require more regularization -> how much regularization to use might depend on the difficulty of the task.
        - This makes sense to the overall MoE theory as easier tasks have less complex data distributions The less complex data distribution will lead to less of the experts being called consistently, causing them to overfit. In a complex task, the data distribution will result in a more distributed load balancing due to more semantic/syntax patterns being in place, thus using more experts, preventing overfitting.
    - There might be router issues leading to this difficulty in fine-tuning on easier tasks as well.
- Expert-choice seems to be better than regular token-choice routing. However, ST-MoE, which has improvements over traditional token-choice routing, surpasses expert-choice.

    - Why did Mixtral decide to not use Expert-Choice and seems to use a routing strategy that resembles GShard more, even though it underperforms both Expert Choice and ST-MoE's routing strategies? Maybe they started training before this paper came out? (investigate if Mixtral's routing strategy resembles more GShard than ST-MoE).

- Z-loss is better for token-choice, but balance-loss is better for Expert-Choice?
- The routing learned during pre-training is thought to already have a good estimate of data distributions at a semantic and syntactic level, therefore more specialization is not needed during fine-tuning. The idea is that the semantics and syntax at fine-tuning domains are not new, what changes is their distribution. Therefore, the routing algorithm does not to be updated -> gating/routing should be kept frozen during fine-tuning (this is not the first research work to come to this conclusion).
- MoE models are prone to overfitting, so often underperform dense models on single-task fine-tuning. MoE works better when scaling the number of tasks, that is, fine-tuning on more than just one domain. However, instruction-tuning seems to bring a reversal to this trend, with FLAN-MoE performing better than FLAN-T5 in single task fine-tuning.
    - Perhaps a reason for this is how FLAN does not have a single task per-say, it instead has data from many different domains with the common aspect being the structure how it is presented (in a dialogue format).