https://arxiv.org/pdf/2305.14705.pdf

MoE is a neural architecture design that can add learnable parameters to LLM without additional inference cost

- Conditional computation: enhance the number of model parameters without a corresponding rise in computational expense.
    - Selectively activating only the relevant portions of the model, based on input-dependent factors
- Combine it with instruction tuning for really good performance
    - Conventional, task- specific finetuning of MoE leads to suboptimal performance
- Build upon observation that language models can be decomposed into smaller, specialized sub- models, “experts” that focus on distinct aspects of the input data
    - More efficient computation and resource allocation
- Instability of MoE during fine- tuning or multitask learning is a challenge
- Contributions
    - Expand on known benefits of instruction tuning for task specific downstream finetuning
        - Larger impact on MoE than dense
    - Necessity of instruction tuning stage for MoE models to surpass dense models on downstream and held out tasks
- Flan- moe == flan-st

Instruction tuning: enhances performance on specific tasks by adapting their pre- trained representations to follow natural language instructions

- model is trained using pairs of input-output instructions, enabling it to learn specific tasks guided by these instructions

Three experimental setups

- Direct finetuning on individual downstream tasks
- Instruction tuning then in context, few shot, zero shot generalization on downstream (MoE better)
- Instruction tuning enhanced with subsequent finetuning on individual downstream (MoE better)

Flan mixture

MoE layer: collection of independent feed forward networks, “experts”

- Gating function uses softmax to model probability distribution over experts (how well each expert is able to process the incoming input)
    - each capable of handling distinct tasks or aspects of the problem space
- Even though each layer has more parameters, experts are sparsely activated
- Each layer’s learnable gating network trained to use its input to activate best two experts for each token of an input sequence

    - Collection of O(E^2) different combinations of feed forward networks instead of one in classic Transformer
- Routing strategy: intelligently distribute input data among multiple specialized experts, each optimized for handling specific subsets of the input space
    - crucial for maximizing the utilization of the model's capacity while minimizing the risk of overfitting
    - Token choice: token select top K experts
    - Expert choice: experts select top K tokens