GLaM: Efficient Scaling of Language Models with Mixture-of-Experts

Main Idea: with the improvement of language models mainly coming from scaling the number of parameters in a dense setting, training these models requires more and more compute and resources. GLaM looks to explore sparse language models (MoE) to reach comparable or superior performance to dense models while decreasing training costs. During evaluation, GLaM focuses on zero-shot and few-shot learning capabilities. The importance of data quality during pre-training is also analyzed.

The largest GLaM model has:

- 1.2T total number of parameters.
- 96.6B active parameters.
    - For comparison, GPT-3 is a 175B parameter dense model.
- 64 experts per MoE layer.

GLaM seems to have been the first study to use a decoder-only MoE on a model of comparable size to GPT-3. Switch, for example, had only around 1B active parameters (compared to 96.6B of GLaM) per input and was an encoder-decoder model.

The training dataset used to train GLaM was highly filtered to ensure that low-quality content was not prominent (although a small collection of low-quality training data is kept to prevent systematic biases).

Architecture

- Alternate between FF (dense) and MoE (sparse) layers.
- Regular top-2 routing, with the output being a weighted average based on the scores given by the routing.
- Auxiliary load balancing loss.

Evaluation Setting

- Mainly focuses on zero-shot, one-shot and few-shot performances of the models being evaluated.
    - This is different to Switch, which focuses on fine-tuning performance.
    - This is consistent with new capabilities shown by scaling LMs as shown by GPT-3.

Results

- GLaM (64B/64E) (96.6B active parameters) has consistent gains in zero, one and few-shot performances over GPT-3, while requiring roughly only half of the compute FLOPs at inference (96.6B vs 175B).
- Improved performance (over GPT-3) on the challenging TriviaQA domain indicates that the additional capacity of GLaM plays a crucial role in its performance gains.

    - On GPT-3's paper, GPT-3 was shown to consistently improve on this task (TriviaQA) given an increase in parameters, which was attributed to its ability to retain more knowledge with an increase in parameters.
- Using a small model (GLaM (1.7B/64E)), it was shown that the quality of the pretraining data plays a crucial role in determining the quality of the model.
    - The impact of data quality was bigger in NLG tasks compared to NLU tasks.
- MoE models can be scaled in two ways:
    - Increasing the number of experts
        - Keeps the number of active parameters (and thus the compute FLOPs at inference) constant.
        - Increasing the number of experts generally resulted in better performance up to 64 experts (there was a decline in performance in further increases after 64).
    - Increasing the size of experts
        - Leads to an increase in inference costs.
        - Results in improved performance.
- GLaM MoE models perform consistently better than GLaM dense models for similar effective FLOPs per token.
    - MoE models perform similarly to dense at smaller scales but obtain an advantage when scaling the model.
- In terms of data efficiency, GLaM MoE models require significantly less data than dense models of comparable FLOPs.

    - When the same amount of data is used for training, MoE models perform much better, and the difference in performance becomes larger when training up to 630B tokens, so this advantage increases with scale.
- In terms of computational efficiency and energy consumption, sparse models take much less computational resources to achieve the same performance.
    - GLaM (64B/64E) has around 1/3 of training costs of GPT-3, while also halving the inference cost and using 1/6th of the energy costs.
        - These gains can be attributed to the MoE architecture's superior training efficiency.

My takeaways:

- In terms of architecture, GLaM does not seem to provide any significant advancements in MoE. The main quality of this research was to analyze how the MoE architecture would perform at a large scale (especially of number of active parameters) in a decoder-only model for NLG.