StableMoE: Stable Routing Strategy for Mixture of Experts

Main Idea: the goal of this paper is to solve the sample efficiency issue of training MoEs. The expert selection for a specific input may change during training, causing the weights of experts to be updated that will not be using it in inference – suboptimal training with experts being updated based on an input space that is not attributed to them during inference (routing fluctuation problems).

Problem

By observing the routing fluctuation issue when using BASE layers, it was observed that:

- 40.9% of tokens are unstable (inconsistent in routing) after 20% of the training steps.
- this number decreases to 29.1% after 50% of training, and to 15.4% after 80% of training.

Solution

Split training into 2 parts:

- Stage 1
    - start by training a router (with a new balance loss introduced – not much different, simply penalizes the loss in the case of expert overloading) and using sigmoid instead of SoftMax (sigmoid is thought to propagate the signal better) for determining the assigned expert's weight.

    - During stage 1 of training, the router is distilled. This distillation process is accounted for in the training loss:
        - Total loss = task loss + balance loss + distillation loss.
        - The components that are important for this distillation are the experts' centroids and the routing feature of the token t (distilled through a word embedding).
    - At the end of training stage 1, the parameters for the distilled router (which were being trained synchronously) (these parameters are the word embeddings for the tokens and the experts' centroids) are frozen and kept frozen for the remainder of training (which consists of stage 2).
- Stage 2
    - In stage 2 of training, the router is distilled and stable, so only the task loss is needed. The sigmoid gate is kept so the gating signal is still being trained (I believe this is only for the actual weights given to each expert at inference). Everything else remains the same.

Results

The StableMoE method is compared to a dense Transformer, a Base MoE, a Hash Layer MoE and Switch Transformer at a base and a large setting (454M and 3.22B total parameters, respectively).

- StableMoE outperforms all others in all settings and shows robustness in scaling both model parameters and number of experts.

- Models improve perplexity with a higher number of experts (tested up to 64), given the same model size.
- Stacking MoE layers in-between Transformer blocks was shown to have the best results in comparison to sticking them in other positions.

My takeaways:

- At first glance, it seems logical that the routing fluctuation issue presented will result in suboptimal training, so traditional MoEs leave room for improvement in terms of training efficiency, especially in early stages of training.
- The part which seems to help the most is the routing distillation. The idea is to learn parameters to learn optimal expert centroids and token embeddings. Once this is learned, the router can be frozen to keep stability during training.
- The paper provides evidence that scaling the number of experts with StableMoE leads to improved performance not only in pre-training but also in downstream tasks like multilingual machine translation, as evidenced by higher average test BLEU scores compared to other models. This indicates that the advantages of scaling are not confined to pre-training. However, the paper doesn't provide an extensive evaluation on a variety of downstream tasks or fine-tuning with different amounts of data, which would be valuable for comprehensively understanding the scalability and efficiency of the model in varied contexts.