Parameter-Efficient Mixture-of-Experts Architecture for Pre-Trained Language Models

Main Idea: proposes an architecture to make more efficient use of parameters in MoE models by sharing information among experts. Mainly uses matrix product operator (MPO), a tensor decomposition approach from quantum physics to reconstruct the expert layer, then shares parameters from the central tensor (core information) between experts while maintaining specificity through auxiliary tensors (complementary to the central tensor). The intuition behind this approach is to solve MoE's issue of expert redundancy (different experts learning common knowledge, leading to parameter-inefficiency).

Approach – MPOE

- Core idea is to share the central tensors from the expert layers and enable specificity via expert-specific auxiliary tensors based on the matric decomposition strategy.
    - The final MoE layer would consist of a shared central tensor (looks the same for each expert) and small auxiliary tensors (unique to each expert).
- The central tensor acts like a global parameter – is the same for each expert in a layer.
    - Less total parameters are then needed in total since each expert layer will contain a globally shared tensor for all experts (the central tensor) while retaining expert specificity through auxiliary tensors specific to each expert.

    - Idea is to capture the shared knowledge between experts in the central tensor, and the specialized expert knowledge in the auxiliary tensors.
- In theory, MPOE leads to suboptimal optimization since central tensors are always updated. To stabilize the optimization process, a gradient mask strategy is used:
    - The central tensor is not always updated (determined randomly).
    - Equivalent to a gradient dropout, employed in the central tensor of each MoE layer.
- MPOE is employed on already pre-trained language models (for the matrix decomposition to make sense, the models need to already have been pre-trained, having knowledge to decompose).

<u>Experiments</u>

- GPT-2 (decoder-only) and T5 (encoder-decoder) are used as base models for MPOE.
- 8 experts per MoE layer are generally used.
- Adding MPOE to fine-tune pre-trained LMs in downstream tasks leads to better performance than Switch with a 27.2x parameter reduction.
    - MPOE is especially better at low-resource tasks, indicating that MPOE's parameter-sharing leads to positive task transfers.
    - The caveat is that MPOE needs an already pre-trained LLM.
- Adding more experts (and thus having more auxiliary tensors) leads to improved MPOE performance.

- MPOE can also potentially work well in a multi-task setting (with task-level routing).

My takeaways:

- DeepSeekMoE is a recent model that was also trained with the idea of improving parameter efficiency by sharing weights of experts to capture common knowledge.
- Also is like sparse upcycling and parameter-efficient sparsity crafting in the sense that it takes a pre-trained LLM and modifies its architecture to have the advantages of MoE.
- This approach is compatible with distillation techniques to further improve inference time.