DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models

Main Idea: traditional top-k MoE experts acquire non-overlapping and focused knowledge, creating challenges for expert specialization. DeepSeekMoE plans on architectural changes to enforce expert specialization through expert segmentation and isolating experts as shared ones to capture common/overlapping knowledge.

3 versions of DeepSeekMoE are trained (in total # of parameters):

- 2B
- 16B
    - Can run on a single GPU with 40GB of memory.
    - Experimented with SFT to create an instruction-tuned chat model.
- 145B
    - Largest model trained.

2 Potential Issues of Traditional (top-k) MoE:

- Knowledge hybridity
    - Current MoE models have a low number of experts (8 or 16). This division assigns each expert to a diverse part of the data, so the parameters are not used so efficiently (there is more sparsity in the data possible than reflected by the number of experts).

- Knowledge redundancy
    - Experts may benefit from common knowledge, but since they are isolated, some experts might end up learning the same information, causing redundancy in their parameters.

Solutions Proposed by DeepSeekMoE

- Fine-grained Expert Segmentation
    - Segment experts into a finer grain by splitting the FFN hidden dimension. More experts are also activated (increase the number of experts while maintaining the number of total and active parameters).
    - More flexibility on which parameters of the experts to use – introduce sparsity within each expert – while keeping computational costs constant.
- Shared expert isolation
    - Isolate certain experts to serve as shared experts, which are always activated. The goal is for these experts to retain the common knowledge between experts, avoiding parameter redundancy.
        - Leads to parameter-efficiency + increased specialization.

Architecture

- As mentioned above, DeepSeekMoE incorporates two new strategies on top of the generic MoE architecture:

- Fine-grained expert segmentation
    - Not just simply adding more experts but keeping the number of active and the number of total parameters the same while doing so.
    - A small number of experts combined with a low number of activated experts per input makes experts learn a diverse amount of knowledge when what we want is specialization.
    - To solve this, DeepSeekMoE divides the expert's weights (more specifically, the FFN hidden dimension) into m segments, creating another level of experts. This allows for a scaling of m in the number of experts (if m is 8, the total number of experts will be scaled by 8, for example).
        - mN possible expert combinations vs N possible combinations.
    - This allows for a more flexible combination of experts, since the router will not only pick specific experts, but specific segments within experts.
    - This allows for a greater number of experts to be activated without increasing computational costs.
- Shared expert isolation
    - Experts in conventional MoE are isolated. This means that if experts have overlap in knowledge in the data fed to them, this will be learned independently, so repeated parameters will exist for the same information, bringing parameter inefficiency.

        - DeepSeekMoE has shared experts – experts that are always activated – which have the goal of capturing this common knowledge so there is no parameter redundancy.
        - The number of shared experts is Ks. To keep computational costs, the number of routed experts will then decrease to mN-Ks and the nonzero gates (segment activations) will be mK-Ks.
    - Balance loss
        - An expert-level and a device-level balance loss are used, with more emphasis/weight on the device-level loss.

Experiments (2B parameter model)

- Substitute all FFNs by an MoE layer.
- 9 Transformer blocks with hidden dimension of 1280.
- Random initialization.
- 16 experts with 4 segments each (64 total expert segments), with 1 shared segment.
- Computation equivalent of top-k with k=2.
- 2B parameter model, 0.3B active parameters.
- Training of 100B tokens with 2k batch size.
- No dropout due to abundance of data used.
- Baselines:
    - Dense – equivalent to top-1 routing (~0.2B active parameters)

    - Switch – equivalent to top-1 routing (~0.2B active parameters)
    - Hash Layer – equivalent to top-1 routing (~0.2B active parameters)
    - GShard
- Results
    - Switch and Hash Layer perform better than Dense (with same number of active parameters but more total parameters).
    - GShard performs slightly better than Switch (with more active parameters).
    - DeepSeekMoE performs significantly better than GShard, with the same number of active and total parameters.
    - DeepSeekMoE closely aligns with the upper bound of MoE models (dense with same number of total parameters) (at least on the 2B total parameters scale when training with 100B tokens).
        - DeepSeekMoE 2B performs comparably to GShard 2.9B (1.5x the expert size) (the advantages increase when scaling to 13.3B and 19.8B, respectively).
        - DeepSeekMoE 2B achieves comparable performance to Dense with FFNs scaled by 16 (same number of total parameters, 16 is number of experts per layer used).
    - Ablation studies reassure the positive effects brought by fine-grained expert segmentation and shared expert isolation.

        - Additionally, the number of shared experts (1,2 and 4 tested with 64 total experts) did not seem to make much difference. A ratio of 1:3 (shared/total activated experts) is used when scaling the architecture.
    - Expert specialization
        - DeepSeekMoE was more sensitive to disabling the top-k experts, showing that there is less common knowledge between experts, thus less redundancy.
        - Shared experts are irreplaceable in DeepSeekMoE, that is, substituting a shared expert by a not-shared expert results in a significant drop in performance.
        - DeepSeekMoE can acquire knowledge more accurately and efficiently. Even using only 4 active experts (equivalent to top-1 routing), DeepSeekMoE performs similarly to GShard.
            - When using this setting of 4 active experts at training time, and not only at inference time, DeepSeekMoE outperforms GShard even with half of the number of active expert parameters.

<u>DeepSeekMoE 16B</u>

- Scaling up of the architecture to a model with 16B total parameters, trained on 2T tokens (same number of training tokens as Llama2-7B).
- 28 Transformer blocks, all FFNs are substituted by an MoE layer except for the first one (because the first layer takes longer to converge if the FFN is substituted by an MoE layer).

- Each MoE layer has 2 shared and 64 routed experts. Each FFN is divided into 4 experts.
- 8 experts per layer activated per input (2 shared, 6 routed), corresponding to 2.8B active parameters.
- Similar training setting to DeepSeekMoE 2B.
- Compared to DeepSeek 7B (its dense counterpart):
    - DeepSeekMoE 16B, with around 40% of active computation at inference, performs comparably to DeepSeek 7B.
    - DeepSeekMoE 16B performs especially well in language modeling tasks.
        - This indicates that scaling up the total FFN parameters helps with memorization.
    - DeepSeekMoE 16B does not perform well in multiple-choice questions.
        - A possible explanation for this can be due to the attention parameters. The number of attention parameters are thought of as being crucial for MC tasks, and the MoE version has around 5x less attention parameters than its dense counterpart (0.5B vs 2.5B).
- Compared to Llama2-7B:
    - DeepSeekMoE 16B, with about 40% of Llama2-7B activations at inference, outperforms it at most baselines (MC tasks like MMLU are the exceptions).
    - DeepSeekMoE 16B is stronger at math and reasoning tasks (strengths of Llama2-7B) probably due to the distribution of the dataset used for training.
    - Despite being trained on less English text, DeepSeekMoE 16B achieves equal or better performance at English understanding and knowledge-intensive tasks.

        - Consistent with MoE's advantage in memorization due to increase total parameter count compared to dense.
    - On Hugging Face's Open LLM leaderboard (collection of evaluation tasks), DeepSeekMoE 16B significantly outperforms models of the same size in terms of active parameters and achieves comparable performance to Llama2-7B.

Chat Alignment for DeepSeekMoE 16B (SFT/Instruction-Tuning)

3 models are compared in this section, all trained on the same data:

- Llama2 SFT 7B – Llama 2 instruction-tuned independently from its chat version, to control for the training data.
- DeepSeek Chat 7B.
- DeepSeekMoE 16B Chat – has around 40% of active computations compared to the other models used in this section.

Results:

- The MoE variant achieves comparable performance to the dense models in language understanding and reasoning, machine reading comprehension, and mathematical and knowledge-intensive tasks.
- The MoE variant performs significantly better at code generation.
- The gap in multiple-choice questions still exists but is narrowed.

Scaling DeepSeekMoE to 145B Total Parameters

- Trained on 245B tokens (will probably be scale dup in the future, so this can be seen more as a baseline).
- 62 Transformer blocks, all FFNs substituted by an MoE layer except the first one.
- 4 shared experts and 128 routed experts per MoE layer.
- Each expert is $1/8^{th}$ the size of a standard FFN (different than the ratio used for the smaller 2B and 16B models, which was $1/4^{th}$).
- At inference, the 4 shared experts and 12 routed experts are activated.
- Around 22.2B active parameters.

Results:

- 3 additional models were trained for comparison, using the same training corpus and hyperparameters:
    - DeepSeek 67B (dense)
    - GShard 137B (GShard architecture trained on the same data)
    - DeepSeekMoE 142B (half-activated)
        - Uses half of the activations of DeepSeekMoE 145B – 2 shared experts, 6 routed experts.
- With similar number of active and total parameters, the MoE 145B variant significantly outperforms GShard.
- With only 28.5% of its active computations, the 145B MoE model reaches comparable performance to DeepSeek 67B.

    - Exhibits strong performance in language understanding and knowledge-intensive tasks but struggles in multiple-choice (consistent with the 16B MoE model performance).
- Despite having only half of the activated parameters, the 142B version is not too far behind from the 145B fully activated version, still matches the performance of DeepSeek 67B (with around 18.2% of its computations at inference) and easily beats GShard 137B.

My takeaways:

- DeepSeekMoE has its 16B version with 7 checkpoints released to HF. This could add to a potential exploration of how experts in MoE specialize.