電機資訊學院 2025 作 BRAIN PLUS HAND 實作專題競賽

Pool of Experts: Cross Layer Cooperation to Enhance Performance

EECS02 吳冠霆 林禾堃

Abstract

Recently, the Mixture of Experts (MoE) technique has gained traction in the development of Large Language Models (LLMs), demonstrating its potential to lower barriers and enable individuals with limited resources to create powerful language models. Notable examples include Mixtral_8x7B, JetMoE, and DeepSeekMoE. However, a common limitation among these models is that experts within each layer are confined to their respective blocks, creating a barrier between layers. This restriction may lead to a lack of flexibility and limited representation capacity.

To address these limitations, we propose a new architecture called Pool of Experts (PoE). PoE merges experts from multiple layers into a shared pool, allowing routers to select experts across different layers. In the initial phase, we used TinyLlama MoE as the backbone for our experiments and observed improved performance on most benchmarks with minimal training resources. In the next experiment phase, we aim to scale this method to larger MoE LLMs to evaluate its potential as a low-cost solution for transfer learning.

Future experiments will explore some extended possibilities such as the pool of LoRA and DRL-based techniques. Our goal is to demonstrate that this economical architecture is ideally suited for our ever-evolving society, particularly highlighting its potential for green computation in LLMs given its affordable training requirements.

Motivation

Over the past few years, we have recognized the immense potential that large language models (LLMs) offer. One notable architecture is the Mixture of Experts (MoE), which integrates specialists from different fields to collaborate at each layer of the model, thereby demonstrating enhanced performance across various tasks. This approach addresses the common limitation of LLMs being overly generalized, which often makes them less effective at handling highly specialized or knowledge-intensive tasks.

Our observation extends to human collaboration, which bears similarities to the MoE architecture. Just like each layers in MoE, individuals possess their own unique skills and expertise. However, if these individuals work sequentially without interaction, the collective productivity of society may not be optimized. This lack of interaction can hinder the effective integration of specialized knowledge when tackling more complex problems.

Based on these insights, we propose a new architecture aimed at addressing the limitations observed in the MoE framework. Our goal is to enhance the overall skill set of LLMs, making them more adept at handling challenging real-world tasks. By fostering a more integrated and interactive approach, we hope to improve the performance and applicability of LLMs across a broader range of specialized and complex scenarios.

Experiment Result

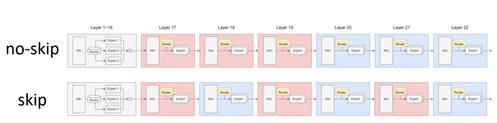
To determine whether our method can genuinely enhance the performance of Language Models, we conduct a series of experiments aimed at selecting the optimal architecture:

Model	TinyLlama-4x1.1B-MoE	**Pretrain dataset**	Wikipedia
Layers	22	**Device**	AMD MI210
Experts per layer	4	**Training step**	3000
Routing strategy	Top2	**Optimizer**	Adam

Setup

LM	HellaSwag (acc_norm)	OpenBookQA (acc_norm)	MMLU (0-shot) (acc)	MMLU (5-shots) (acc)	GSM8K (exact_match)	TriviaQA (exact_match)	Winogrande (acc)	GPQA (acc)
TinyLlama-4x1.1b-MoE	50.79%	33.80%	24.01%	24.93%	1.14%	2.48%	55.41%	25.00%

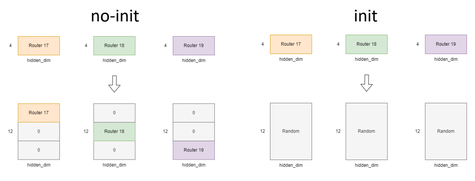
Baseline



LM	HellaSwag (acc_norm)	OpenBookQA (acc_norm)	MMLU (0-shot) (acc)	MMLU (5-shots) (acc)
TinyLlama-4x1.1b-3l-3l (no-skip)	**52.05%** (+1.26%)	**31.80%** (-2.00%)	**24.21%** (+0.20%)	**24.26%** (-0.67%)
TinyLlama-4x1.1b-3l-3l (skip)	51.88% (+1.09%)	**31.80%** (-2.00%)	23.12% (-0.89%)	23.91% (-1.02%)

LM	GSM8K (exact_match)	TriviaQA (exact_match)	Winogrande (acc)	GPQA (acc)
TinyLlama-4x1.1b-3l-3l (no-skip)	**1.14%** (+0.00%)	**2.86%** (+0.38%)	**56.83%** (+1.42%)	**28.35%** (+3.35%)
TinyLlama-4x1.1b-3l-3l (skip)	0.91% (-0.25%)	2.79% (+0.31%)	56.12% (+0.71%)	25.89% (+0.89%)

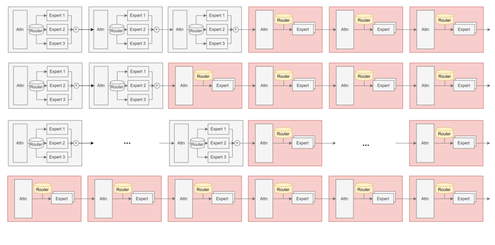
Pooling strategy



LM	HellaSwag (acc_norm)	OpenBookQA (acc_norm)	MMLU (0-shot) (acc)	MMLU (5-shots) (acc)
TinyLlama-4x1.1b-3l (init)	**52.02%** (+1.23%)	32.00% (-1.80%)	23.14% (-0.87%)	24.05% (-0.88%)
TinyLlama-4x1.1b-3l (no-init)	51.73% (+0.94%)	**32.60%** (-1.20%)	**24.83%** (+0.82%)	**24.63%** (-0.30%)

LM	GSM8K (exact_match)	TriviaQA (exact_match)	Winogrande (acc)	GPQA (acc)
TinyLlama-4x1.1b-3l (init)	**1.21%** (+0.07%)	**2.01%** (-0.47%)	55.72% (+0.31%)	24.55% (-0.45%)
TinyLlama-4x1.1b-3l (no-init)	**1.21%** (+0.07%)	1.94% (-0.54%)	**56.20%** (+0.79%)	**25.67%** (+0.67%)

Initialize strategy



LM	HellaSwag (acc_norm)	OpenBookQA (acc_norm)	MMLU (0-shot) (acc)	MMLU (5-shots) (acc)
TinyLlama-4x1.1b-MoE (baseline)	50.79%	33.80%	24.01%	24.93%
TinyLlama-4x1.1b-3l	52.02% (+1.23%)	32.00% (-1.80%)	23.14% (-0.87%)	24.05% (-0.88%)
TinyLlama-4x1.1b-4l	51.94% (+1.15%)	32.60% (-1.20%)	23.49% (-0.52%)	24.41% (-0.52%)
TinyLlama-4x1.1b-5l	52.07% (+1.28%)	32.40% (-1.40%)	23.77% (-0.24%)	24.36% (-0.57%)
TinyLlama-4x1.1b-6l	51.88% (+1.09%)	32.00% (-1.80%)	23.54% (-0.47%)	23.70% (-1.23%)
TinyLlama-4x1.1b-7l	52.33% (+1.54%)	32.80% (-1.00%)	23.80% (-0.21%)	25.47% (+0.54%)
TinyLlama-4x1.1b-8l	52.44% (+1.65%)	33.20% (-0.60%)	22.95% (-1.06%)	24.73% (-0.20%)
TinyLlama-4x1.1b-22l	**54.43%** (+3.64%)	**34.60%** (+0.80%)	**25.08%** (+1.07%)	**26.61%** (+1.68%)

LM	GSM8K (exact_match)	TriviaQA (exact_match)	Winogrande (acc)	GPQA (acc)
TinyLlama-4x1.1b-MoE (baseline)	1.14%	2.48%	55.41%	25.00%
TinyLlama-4x1.1b-3l	1.21% (+0.07%)	2.01% (-0.47%)	55.72% (+0.31%)	24.55% (-0.45%)
TinyLlama-4x1.1b-4l	0.99% (-0.15%)	2.30% (-0.18%)	55.88% (+0.47%)	23.44% (-1.56%)
TinyLlama-4x1.1b-5l	0.83% (-0.31%)	2.05% (-0.43%)	56.27% (+0.86%)	27.01% (+2.01%)
TinyLlama-4x1.1b-6l	1.21% (+0.07%)	2.48% (+0.00%)	56.43% (+1.02%)	24.78% (-0.22%)
TinyLlama-4x1.1b-7l	1.29% (+0.15%)	2.80% (+0.32%)	56.83% (+1.42%)	26.34% (+1.34%)
TinyLlama-4x1.1b-8l	1.06% (-0.08%)	3.66% (+1.18%)	55.56% (+0.15%)	**28.12%** (+3.12%)
TinyLlama-4x1.1b-22l	**1.67%** (+0.53%)	**8.58%** (+6.10%)	**59.67%** (+4.26%)	24.55% (-0.45%)

Pooling layer number

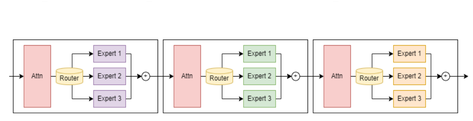
Method

Considering we've already had a MoE structured LLM, that is, in each layer (block), we have some expert feed-forward networks. What we are going to do is to select some layers and extract those expert feed-forward networks in these layers, then concatenate them into a pool. Finally, we will revise the router in those layers so that they can properly select from the mixed pool. The visualization of this concept is the figure below.

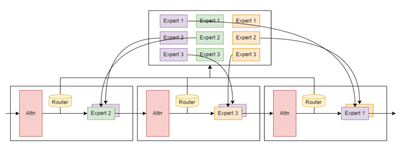
In each layer of normal MoE, the router can only choose experts inside its layer, so the barrier between layers occurs. Conversely, the below figure shows the architecture of PoE. In each layer, the router can choose not only experts in its block but also those in different layers which is in the same pool.

We pretrain our architecture on a dataset with only parameters inside router being set to be require_grad. Therefore, the computation graph will only need to update those parameters inside the routers.

This new architecture retains the advantages of the original MoE design, such as leveraging domain-specific expert networks and utilizing KV-cache in certain transformer-based LLMs to accelerate inference, as the attention order remains unchanged. Beyond these benefits, PoE enhances the architecture's ability to represent more complex concepts. Given these improvements, we anticipate significant advantages from our approach.



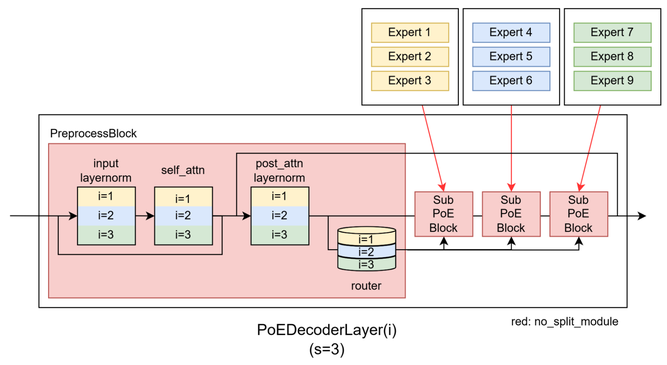
Normal MoE Architecture



Proposed PoE Architecture

To train large models exceeding 56 billion parameters, we use the Fully Sharded Data Parallel (FSDP) technique on an AMD HPC server, distributing the workload across multiple computational nodes. Since parameters, except for those in the router layers, are frozen during pretraining, we save only the router parameters in the model checkpoint.

For evaluating the PoE model, we optimized the PoE architecture to reduce evaluation time by over 6x, enabling efficient evaluation of larger PoE models with model parallelism.



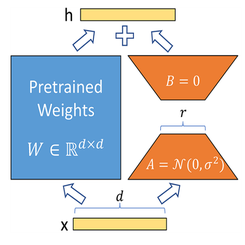
Proposed PoE Architecture with performance improvement

We gather all parameters including layer normalization, self attention, post attention layer normalization, and router into the PreprocessBlock. Instead of forming a single expert pool, we divide it into s SubPoEBlocks. Each block, as indicated within the red boundary, resides on the same GPU device, specified as a 6x improvement in evaluation performance. Consequently, we can evaluate PoE architectures with pool sizes of up to 32, as demonstrated in the Mixtral 8x7B configuration.

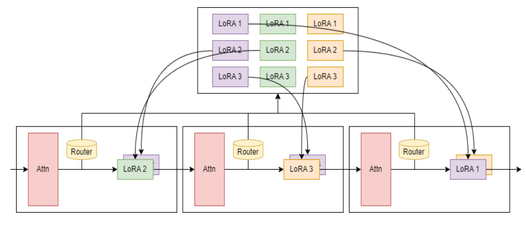
Application

Pool of LoRA: The new possibility combining PEFT (Parameter EfficientFineTuning) and PoE

Based on the success of the proposed PoE architecture, sharing various experts across layers can further enhance model performance. Another approach to create diversify networks can be done by Low Rank Adaptation (LoRA), which is expected to further improve the scalability of the PoE architecture, while also leveraging the benefits of the Pool of Experts.



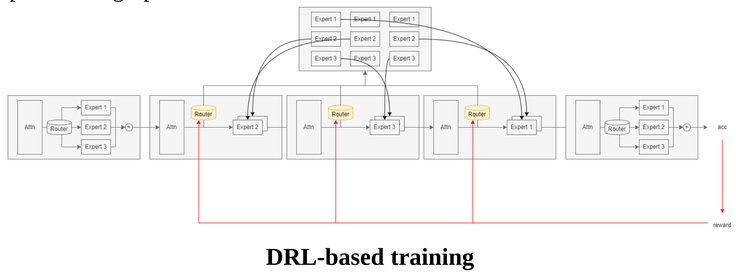
LoRA



Pool of LoRA

DRL-based training method: might be able to train the whole system with lower cost

While only the routers are required to be trained in the PoE architecture, using the whole computational graph is inefficient. Instead, leveraging Deep Reinforcement Learning (DRL) for training can reduce costs by shrinking the computational graph.



DRL-based training