HydraLM: Swappable-QLora MoE Llama-2 v1

Current Status: POC/V0 IN-PROGRESS
Discord Server: https://discord.gg/CZAJcWTZxX

Summary (A brief overview of the project, including its goals, objectives, and timeline.**)**

The main goal is to enhance the Llama v2 base model by converting it into an MoE framework through swappable QLora Expert Adapters in order to gain capabilities closer to the GPT4 model. This would also serve as an open-source attempt to understand if MoE-style modeling can be used to enhance the model capabilities.

Project Goals and Objectives

The Llama v2 model has recently been released by Meta-AI (both base and chat variants) and is released in multiple sizes 7B, 13B, and 70B which are available for commercial use as well. We want to use this model and enhance its capabilities by converting it into a MoE model. The final model would be released for public use and could also be used as an open-source proof of concept to understand if MoE-type expert models can outperform dense models.
We need to design the overall project along with the goals and objectives. Moreover, we need to concretize the experimental settings, high-level modeling choices, and design as well.

We would create an MoE model by stitching together a bunch of QLora domain-specific adaptations on the base Llama v2 models. Each example during the inference would be routed through a set of QLora modules or their combination. This routing the main things that need to be learned in some way or the other. The goal is to solve an existing task better by using specialized domain experts.

Ideally, we would release a MoE model better than Llama v2 along with a research paper.

Modeling Approach

1. *Llama 2 7B*

2. 4-bit QLora ExpertLM Forest
    a. Our experts would be 4bit-QLora modules finetuned using the bnb library. To do this first we would need to define some set of tasks/domains on which each of the experts will specialize (maybe coding/math/reasoning etc or just cluster the data or maybe specific downstream datasets). Then we need to train the QLora modules for each of these domains.
    b. Tasks
        i. QLora script (with bnb):
            1. Original Implementation: https://github.com/artidoro/qlora/blob/main/qlora.py

2. Hugging Face Example: https://github.com/huggingface/peft/blob/main/examples/fp4_finetuning/finetune_fp4_opt_bnb_peft.py
3. Third Party Implementation: https://gist.github.com/younesbelkada/9f7f75c94bdc1981c8ca5cc937d4a4da

ii. Experts

1. **Math**
2. **Reasoning**
3. **Coding**
4. **Truthfulness**
5. **Science**
6. **Tool Calling abilities**
7. **Handling structured inputs and outputs** (jsons etc)

c. **Expert Evaluation:**

i. Evaluation Codebase: https://github.com/EleutherAI/lm-evaluation-harness

ii.

3. MoE Routing
   a. During inference, we would have access to a base Llama v2 model (say \theta_b) and multiple QLora adaptation parameters from each expert for each layer. For instance, the i^th expert QLora modules would contain Ei = {e^i_1, e^i_2, … , e^i_n} parameters for each of the layers in the model. Given \theta_b and E1, E2, … Ek and a query Q, during inference the goal is to create a query expert Eq = {e^q_1, e^q_2, … , e^q_n} such that the performance on the query Q is the best and where e^q_j could be any function of f(e^1_j, e^2_j , … , e^k_j). The function F could either perform the forward passes on each QLora adaptation and then ensemble them or route through top-k modules at each layer or merge some modules to avoid multiple forward passes in order to generate a response.w
   b. Inference:
      i. Load adapters/experts
      ii. Centroids for each dataset/adapter/expert
      iii. Centroids for Query
      iv. Retrieve expert (or multiple experts)
      v. If multiple (max 2) (then weighted merge adapters)
      vi. Attach adapters to model
      vii. Run inference

Capabilities to Consider: Please add datasets for each one of these that in your experience are good to train on. These capabilities would turn out to be our expert models and then we would like to adaptively route through them which would be an important part.

a. **Math/Science:**
   i. gsm8k · Datasets at Hugging Face
   ii. camel-ai/math · Datasets at Hugging Face
   iii. metaeval/ScienceQA_text_only · Datasets at Hugging Face
   iv. sciq · Datasets at Hugging Face
   v. camel-ai/physics · Datasets at Hugging Face
   vi. camel-ai/chemistry · Datasets at Hugging Face
   vii. camel-ai/biology · Datasets at Hugging Face
b. **Reasoning/Logic:**
   i. QingyiSi/Alpaca-CoT · Datasets at Hugging Face
c. **Coding:**
   i. sahil2801/CodeAlpaca-20k · Datasets at Hugging Face
   ii. GPTeacher subset - Code Instruct: teknium1/GPTeacher: A collection of modular datasets generated by GPT-4, General-Instruct - Roleplay-Instruct - Code-Instruct - and Toolformer (github.com)
   iii. camel-ai/code · Datasets at Hugging Face
   iv. https://huggingface.co/datasets/vikp/code_with_explanations
d. **Science:**
   i. metaeval/ScienceQA_text_only · Datasets at Hugging Face
   ii. sciq · Datasets at Hugging Face
   iii. camel-ai/physics · Datasets at Hugging Face
   iv. camel-ai/chemistry · Datasets at Hugging Face
   v. camel-ai/biology · Datasets at Hugging Face
e. General Instruct/Reasoning:
   i. Open-Orca/OpenOrca · Datasets at Hugging Face
   ii. **ShareGPT instruct dataset (52k examples)**
   iii. https://huggingface.co/datasets/WizardLM/WizardLM_evol_instruct_70k
f. **Truthfulness:**
g. Tool Calling abilities:
   i. https://github.com/ShishirPatil/gorilla/tree/main/data (API calling)
h. Handling structured inputs and outputs (jsons etc):
i. **Ease of Instruction following:**
j. Writing and Storytelling
   i. story-generation

Related works

1. GLaM: Efficient Scaling of Language Models with Mixture-of-Experts
2. DEMix Layers: Disentangling Domains for Modular Language Modeling
3. Branch-Train-Merge: Embarrassingly Parallel Training of Expert Language Models
4. Scaling Expert Language Models with Unsupervised Domain Discovery

5. Soft Merging of Experts with Adaptive Routing
6. AdapterFusion: Non-Destructive Task Composition for Transfer Learning
7. Nearest Neighbor Zero-Shot Inference
8. Eliciting and Understanding Cross-Task Skills with Task-Level Mixture-of-Experts
9. Mixture-of-Supernets: Improving Weight-Sharing Supernet Training with Architecture-Routed Mixture-of-Experts
10. Sparse Upcycling: Training Mixture-of-Experts from Dense Checkpoints
11. AdaMix: Mixture-of-Adaptations for Parameter-efficient Model Tuning

Datasets Dump:

1. openbookqa · Datasets at Hugging Face
2. Open-Orca/OpenOrca · Datasets at Hugging Face
3. GAIR/lima · Datasets at Hugging Face
4. QingyiSi/Alpaca-CoT · Datasets at Hugging Face
5. neulab/conala · Datasets at Hugging Face
6. yahma/alpaca-cleaned · Datasets at Hugging Face
7. riddle_sense · Datasets at Hugging Face
8. gsm8k · Datasets at Hugging Face
9. ewof/code-alpaca-instruct-unfiltered · Datasets at Hugging Face
10. anon8231489123/ShareGPT_Vicuna_unfiltered · Datasets at Hugging Face
11. ehartford/WizardLM_alpaca_evol_instruct_70k_unfiltered · Datasets at Hugging Face
4. ehartford/wizard_vicuna_70k_unfiltered · Datasets at Hugging Face
5. teknium/GPT4-LLM-Cleaned · Datasets at Hugging Face
6. teknium/orca50k-flagged · Datasets at Hugging Face
7. teknium/GPTeacher-General-Instruct · Datasets at Hugging Face
8. sahil2801/CodeAlpaca-20k · Datasets at Hugging Face
9. GPTeacher subset - Code Instruct: teknium1/GPTeacher: A collection of modular datasets generated by GPT-4, General-Instruct - Roleplay-Instruct - Code-Instruct - and Toolformer (github.com)

10. metaeval/ScienceQA_text_only · Datasets at Hugging Face
11. sciq · Datasets at Hugging Face
12. camel-ai/math · Datasets at Hugging Face
13. camel-ai/code · Datasets at Hugging Face
14. camel-ai/physics · Datasets at Hugging Face
15. camel-ai/chemistry · Datasets at Hugging Face
16. camel-ai/biology · Datasets at Hugging Face
17. crumb/Wizard-EvolInstruct70k-k4 · Datasets at Hugging Face
18. https://huggingface.co/datasets/vikp/code_with_explanations

Notes

7/26 Sync

Datasets Experts:

- **Function Calling, JSON output**
  - …
- **Instruction**
  - **shareGPT:**
    - ShareGPT 52k
  - **Multilingual Instruct**
    - **Guanaco's OASST1 https://huggingface.co/datasets/OpenAssistant/oasst1**
- **Science**
  - sciq · Datasets at Hugging Face
  - metaeval/ScienceQA_text_only · Datasets at Hugging Face
  - **Math**
    - camel-ai/math 50k
  - **Biology**
    - camel-ai/biology · Datasets at Hugging Face
  - **Physics**
    - camel-ai/physics · Datasets at Hugging Face
  - **Chemistry**
    - camel-ai/chemistry · Datasets at Hugging Face
- **Coding**
  - GPT-teacher
  - https://github.com/teknium1/GPTeacher
- **Reasoning/Logic:**
  - https://huggingface.co/datasets/conceptofmind/cot_submix_original
  - QingyiSi/Alpaca-CoT · Datasets at Hugging Face
- **Writing/Storytelling:**
  - story-generation: take only "wp" (todo)

Cluster Experts Dataset: (All dataset experts data + Original Nous Hermes training data) and then cluster

- **Deduplicate**
- **Clean Artifacts (like "Human")**
- **Combine**
- **Cluster**
-

7/20 SyncM

Next Steps

1. -Select 1 dataset per capability (above)
2. -Finalize Finetuner script + Hyperparameters (https://github.com/hydrallm/llama-moe-v1/tree/main)
3. w-Train QLora Experts/adapters (can be decentralized/async)

Key points

Fine-tuning

- No pre-training on the instruct dataset - so we can compare directly against llama2 base performance from the technical report

Evals

- Important to compare with llama2, which should use similar benchmark datasets

Inference

- Routing between adapters vs adding adapters
- Switching adapters – 200 ms per adapter
- Decision: run sequentially

Datasets for initial POC with 5 experts

- Instruct: ShareGPT 52k
- Math: camel-ai/math 50k
- Science: a mixture of all these subsets from camel-ai.
    - camel-ai/physics · Datasets at Hugging Face
    - camel-ai/chemistry · Datasets at Hugging Face
    - camel-ai/biology · Datasets at Hugging Face
- Writing/Storytelling: story-generation: take only "wp" (todo)
- Coding: (select one or mix)
    - sahil2801/CodeAlpaca-20k · Datasets at Hugging Face
    - GPTeacher subset - Code Instruct: teknium1/GPTeacher: A collection of modular datasets generated by GPT-4, General-Instruct - Roleplay-Instruct - Code-Instruct - and Toolformer (github.com)
    - camel-ai/code · Datasets at Hugging Face
    - https://huggingface.co/datasets/vikp/code_with_explanations
- **Reasoning/Logic:**
    - QingyiSi/Alpaca-CoT · Datasets at Hugging Face