Efficient LLM Supervised Fine Tuning

Proposal by Bilge Acun and Mostafa Elhoushi

Motivation

What problem does this project try to solve?

Large language models (LLMs) have shown strong capabilities in knowledge extraction and reasoning. One of LLMs' main capabilities is in-context learning — learning during inference from a group of example question-answer pairs and then answering a new question. However, one of the main challenges of in-context learning is the long context consumed by the examples that lead to large memory requirements. The main approach to long-context windows has been to train models from scratch with long context length, possibly with attention optimizations or approximations (e.g., sparsity-based approaches, Performer, Longformer, etc). In this project, we would like to explore other alternatives to in-context learning.

Who cares? If you are successful, what difference will it make?

The solution coming out of this project could augment or wrap existing LLMs to make them more robust on:

- long dialogue conversations,
- answering questions about large PDF files, or
- auto-completion of code with knowledge of a large repo,

all without the need for expensive re-training from scratch or fine-tuning on large datasets.

Problem

Given a prompt that consists of multiple examples, followed by a question, find the most efficient way to learn from those examples.

Approaches

If long documents could be fine-tuned instead of prefilled into the model, that would remove the limitation on context length that comes from the pretrained model. In prior work, few-shot fine-tuning was compared to in context learning (ICL): **Few shot Fine tuning vs In Context Learning** (https://aclanthology.org/2023.findings-acl.779.pdf). This approach compares fine-tuning to in-context learning as alternative strategies for task adaptation. And it shows that fine-tuning achieves comparable results to in-context learning (which is semantically similar to prefilling in the long document case).

In different work by Anthropic, **"Context Distillation"** was proposed as a better way of fine-tuning on prompts: https://arxiv.org/abs/2112.00861. In context distillation, the model (p_0) is fine-tuned for parameters θ with a loss based on KL divergence between $p_0(X|C)$ and $p_\theta(X)$, where C is the prompt and X is the data that the model originally was trained on. Their approach avoids overfitting when training on a tiny dataset (i.e. prompts) and aims to close the semantic gap between fine-tuning and prompting.

The idea we propose in this project is to apply the context-distillation method to fine-tuning of the natural language inference (NLI) classification task and compare it to the ICL approach.

Benchmarks

What are the common datasets and benchmarks?

Code and benchmarks are open-source and available at https://github.com/uds-lsv/llmft

Models: Depending on the compute resources available different model sizes can be selected. Baselines are available from opt-125m, opt-350m, opt-1.3b, opt-2.7b, opt-6.7b, opt-13b, opt-30b models.

Scope

The goal of the project is to **implement a different fine-tuning approach** from the one that's used in the state-of-the-art "Few shot Fine tuning vs In Context Learning (ICL)" paper. The alternative fine-tuning approach can use "Context Distillation" based fine-tuning proposed by Anthropic, for example. Students could also propose alternative fine-tuning techniques. The current fine-tuning approaches that's included in the paper and repository are:

- **Vanilla fine-tuning** with a randomly initialized classification head on top of the pre-trained decoder.
- **Pattern-based fine-tuning (PBFT)** leveraging the pre-trained language modeling head for classification.

Both of these approaches can be combined with the following parameter-efficient methods:

- BitFit (https://arxiv.org/abs/2106.10199)
- LoRA adapters (https://arxiv.org/abs/2106.09685)

Metrics: Compare **in-domain accuracy** and **out-of-domain accuracy** as shown in the papers. Approaches should also be compared in terms of system resource requirements such as **execution time and memory capacity**.

This is a medium difficulty project that may require 3-4 people to work on.

Resources

- Are there any open datasets the students can train with?
  - Yes, information about datasets is available in the github repo: https://github.com/uds-lsv/llmft

- What are the computing resources required to compute a baseline? (CPU/GPU days)
    - A single GPU can work for evaluating small models (125m & 350m sizes).
    - For larger models of 30b size, at least 4 GPUs are required.
    - LoRA adapters can be used for fine-tuning for memory efficiency.
- What are the computing resources required to compute a SoTA model?
    - The approaches being explored are either to fine-tune samples for a few iterations or just infer them to perform in-context learning. So it does not require full training resources.

Contact

The authors of this proposal are happy to collaborate and help mentor. You can reach them at:

- Bilge Acun-Uyan: acun@meta.com
- Mostafa Elhoushi: melhoushi@meta.com