Week 8 Reading Summaries: GPT-3 and Chinchilla

Amogh Patankar
University of California, San Diego
3869 Miramar Street Box #3037, La Jolla, CA- 92092
apatankar@ucsd.edu

Abstract

With these readings, we dive deep into the most famous generative model in recent years, GPT-3, as well as a paper from DeepMind looking at the training strategies for compute-optimal LLMs. The OpenAI paper specifically trains GPT-3, a 175B parameter model, and tests its performance in a few-shot environment; the testing is done in many scenarios. Chinchilla, a compute-optimal model from DeepMind, is tested on a range of downstream evaluations, and outperforms Gopher, a 280B parameter model, reaching a SOTA level on a famous benchmark, Massive Multitask Language Understanding (MMLU).

1 GPT-3

1.1 Introduction

In recent memory, ML has continued to trend in the direction of NLP systems, specifically, models with large quantity of parameters, being fine-tuned for challenging tasks like Q+A, comprehension, etc. However, for these large models, the quantity of data often limits performance, and pre-training and fine-tuning struggles with a narrowed training distribution. The solution, is meta-learning- models learning a wider range of abilities, and applying specific ones at inference. This paper uses a 175B parameter model (GPT-3), and evaluates on many novel tasks, achieving SOTA performance on most evaluations.

1.2 Approach

For this model, they identify four techniques to improve their power- fine-tuning, zero-shot, one-shot, and few-shot learning. They test 0S, 1S, and FS learning as opposed to traditional finetuning.

1.2.1 Model + Architecture

The paper utilizes the same architecture as GPT-2, but uses dense and sparse attention layers in an alternating pattern; they also test eight different models, all with different sizes and hyperparameters.

1.2.2 Training Dataset

They train the model on a variety of datasets- primarily though, the Common Crawl dataset, which is roughly 1 trillion words. However, they filter it, using high quality corpora, perform deduplication, and add higher quality corpora to CommonCrawl.

1.2.3 Training Process

The authors use gradient noise scale to guide their batch size choices, and use a mixture of model parallelism, inter- and intra- network, while using V100 GPUs.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

1.2.4 Evaluation

They perform a variety of evaluations for GPT-3 for few-shot learning, they evaluate by drawing random samples and use to condition the model. For multiple choice tasks, they use K random samples, with context with correct completion, then calculate LM likelihood. For binary classification tasks, they give more meaningful options, and use multiple choice framework, while with free-form responses, they use beam search, and score using F1 scoring. Then, for model size and learning settings, they use a developmental set for the model size(s).

1.3 Results

With respect to GPT-3, the authors note that language modeling performance is restricted by a power-law, especially with regards to training compute. In the various sections, they evaluate various evaluation frameworks and tests.

1.3.1 Language Modeling, Cloze, and Completion Tasks

For language modeling, they calculate 0S perplexity on the PTB dataset, setting a new SOTA, and because PTB is a traditional dataset, it doesn't have clear example separation. LAMBADA is intended to test long-range dependency modeling, and is the main usage of FS modeling- and hits the SOTA again, by over 18%. Testing HellaSwag and StoryCloze is meant for few-shot ending prediction, and GPT is close but doesn't exceed SOTA.

1.3.2 Closed Book Q + A

The authors measure GPT-3 against factual knowledge Q + A; usage of 0S, 1S, and FS modeling is even stricter than previous work, and fine-tuning isn't allowed. GPT-3 achieves matches or exceeds SOTA for TriviaQA, and WebQS, while coming close for NQs. This suggests a limitation of GPT-3's knowledge capacity.

1.3.3 Translation

For GPT-3, the authors note that the training is 93% in English, and 7% other, meaning that GPT-3 learns from a mixture of languages. This results in GPT-3 performing at a level close to SOTA, but not quite equivalent, for obvious reasons, such as another model being specifically trained on 50%/50% English and French, for ENG to FRE translation, for example.

1.3.4 Winograd-Style Tasks

Winograd tasks, referring to word determination, were used as evaluators for GPT-3, and as such, GPT-3 trails fine-tuned SOTA in other commonsense reasoning challenges.

1.3.5 Common Sense Reasoning

For common sense reasoning, they evaluate on PIQA, where GPT-3 achieves SOTA accuracy for 0S, 1S, and FS; But on ARC (easy, challenge), and OpenBookQA, it falls short of fine-tuned SOTA. Overall, they show mixed results for commonsense reasoning by GPT-3 vs SOTA.

1.3.6 Reading Comprehension

For reading comprehension tasks, GPT-3 is best on freeform datasets, and worst on structured dialog acts with respect to interaction based data. This trend follows for discrete reasoning datasets like DROP too.

1.3.7 SuperGLUE

For the superGLUE benchmark, GPT samples a new set of examples in context for each problem; GPT varies in performance fro the varous benchmarks in SuperBLUE.

1.3.8 Natural Language Inference

For NLI tasks, i.e. understanding the relationship between sentences, GPT is much better in settings that aren't few-shot; GPT-3 performs better than smaller models, but still has room to improve for NLI tasks.

1.3.9 Synthetic + Qualitative Tasks

The authors also test GPT-3 on tasks to perform computations, such as arithmetic, word-scrambling and manipulation, SAT analogies, etc. Across these tasks, GPT-3 is able to perform excessively well, beating SOTA and achieving human level average scores in some cases like SAT analogies.

1.4 Measuring and Preventing Memorization of Benchmarks

The authors also ensure that GPT-3 doesn't simply memorize information that exists in train sets, and regurgitate it during test time/inference time, and this is backed by previous literature. GPT-3 doesn't overfit by a significant amount, meaning the memorization is limited; they also use a clean version of the dataset(s) that removes all leaked examples that match (under 13 grams). Performance in the following: reading comprehension, German translation, reversed words, PIQA, Winograd, and language modeling, only marginally decreases after cleaning. The authors do work for data contamination and clean datasets in order to have a fair, performant model.

1.5 Limitations

GPT-3 has a few limitations. Namely, it struggles with text synthesis and NLP tasks; it seems to repeat itself semantically, and has difficulty with in-context learning. Similarly, it has constraints with respect to algorithm and architecture. Since it's an autoregressive model, it doesn't use bidirectionality, and it may explain mediocre performance in tasks. Other limitations of GPT-3 include sampling efficiency, ambiguity of memorization vs learning, and compute cost and practicality.

1.6 Broader Impacts

The authors discuss the misuse of language models, fairness, bias, and representation, and energy usage. They note that language models can be misused as text generation from GPT is very human-like, and that threat actors pose high danger. They analyze gender, race, and other biases that GPT-3 characterizes biases by, and discuss energy-intensive compute, and note efficiency.

1.7 Conclusion

The authors present a 175B parameter generative language model that shows SOTA performance on many benchmarks and evaluations, while nothing the flexibility of the model.

2 Chinchilla Scaling Law

2.1 Introduction

The authors introduce the problem- that LLMs have large parameter counts (in excess of 500B), and as such, the compute and energy rises. While literature exists showing power law relationship for paraemters and performance, model size and training tokens should be scaled in proportions. The authors look at FLOPs budgets, and try to model pre-training loss to work backwords to find compute numbers. That is, in this paper, the authors come up with a compute-optimal model, Chinchilla, that outperforms Gopher with reduced size and cost.

2.2 Related Work

The authors discuss large language models, scaling behavior for those LLMs, hyperparameter judgment, and improved model architectures. They note that the advent of LLMs has introduced a need for high quality data, and they analyze scaling properties for LLMs. Moreover, they note that tuning hyperparameter combinations and heuristics to determine those result in good performance.

2.3 Estimating The Optimal Parameter/Training Tokens Allocation

The authors look at three different approaches in order to solve the question of model size and training token tradeoff.
With approach 1, they fix the model sizes, as well as the # of training steps required to train the model, with varying training sequences (# of training tokens). In doing so, they find the model size that achieves the lowest loss, after tuning FLOP count, learning rate, etc.
With approach 2, the authors vary the model size for varying FLOP counts, and compute loss. For each FLOP count, they train diverse model sizes; they utilize a power law between FLOPS, model size, and # of training tokens to find the right values for α and β.
With approach 3, the authors model loss as a function of parameter count and seen tokens, then minimize the Huber loss, then plotting as contours, and finding optimal α and β values.

2.3.1 Optimal Model Scaling

With those approaches, they find that the predictions all seem similar- that as compute increases, model size and training data must also increase proportionally. The amount of training data projected is more than currently used for LLMs, meaning that datasets are the big changes to see engineering improvements.

2.4 Chinchilla

The authors discuss Chinchilla, their newer model after GOpher, already a compute-optimal model; Chinchilla is more performant, and has a smaller memory and inference footprint.

2.4.1 Model + Training details

Chinchilla is trained on similar data as compared to Gopher; the distribution, optimizer, tokenizer, and quantization of the weights is different, just as the model architecture is.

2.4.2 Results

- Language Modeling: Chinchilla outperforms Gopher on all evaluations of the Pile dataset, and is more performant than existing models. However, when comparing Chinchilla to other models, train/test leakage affects results, similar to the GPT-3 model from the previous paper.
- MMLU: Regarding the MMLU task, Chinchilla peroms GOpher by roughly 8%, and greater accuracy than other existing models in specific tasks.
- Reading Comprehension: Chinchilla outperforms other models on LAMBADA.
- BIG-bench: Chinchilla outperforms Gopher and other models by roughly 10%.
- Common sense: Chinchilla is able to reach performant metrics for 0, 5, and 10-shot on TruthfulQA dataset/evaluation.
- Closed-book Q + A: Chinchilla also achieves closed-book SOTA performances on accuracies.
- Gender bias + toxicity: Regarding gender and toxicity, the authors conduct a thorough evaluation for Chinchilla. They note that for gender, Chinchilla is better at gender discrimination/categorization than its predecessor, Gopher. Similar performance is noticed for toxicity recognition.

2.5 Discussion and Conclusion

The authors suggest that based on the results of their performant solution, Chinchilla, that LLM trends will move towards optimally setting model size, training duration, and # of training tokens. They note that predicting the scaling of LLMs has limitations, such as intermediate scaling performance, and that high-quality data is required for scaling laws to apply. All in all, a large variety of factors is required for scaling laws to apply correctly for LLMs, as seen by their analysis of Chinchilla, Gopher, and all the other models they evaluated.