OpenAI's GPT-3 Language Model: A Technical Overview

CHUAN LI
JUNE 3, 2020 • 14 MIN READ

by Chuan Li, PhD

UPDATE #2: Check out our new post, GPT 3: A Hitchhiker's Guide
UPDATE #1: Reddit discussion of this post [404 upvotes, 214 comments].

OpenAI recently published GPT-3, the largest language model ever trained. GPT-3 has 175 billion parameters and would require 355 years and $4,600,000 to train - even with the lowest priced GPU cloud on the market.[1]

GPT-3 Key Takeaways

NLP tasks that it has never encountered. That is, GPT-3 studies the model as a general solution for many downstream jobs **without fine-tuning**.

- The cost of AI is increasing exponentially. Training GPT-3 would cost over **$4.6M** using a Tesla V100 cloud instance.
- The size of state-of-the-art (SOTA) language models is growing by at least a factor of 10 every year. This outpaces the growth of GPU memory. For NLP, the days of **"embarrassingly parallel" is coming to the end**; model parallelization will become indispensable.
- Although there is a clear performance gain from increasing the model capacity, it is not clear what is really going on under the hood. Especially, it remains a question of whether the model has learned to do **reasoning, or simply memorizes** training examples in a more intelligent way.

Diving into the Model

GPT-3 comes in eight sizes, ranging from 125M to 175B parameters. The largest GPT-3 model is an order of magnitude larger than the previous record holder, T5-11B. The smallest GPT-3 model is roughly the size of BERT-Base and RoBERTa-Base.

All GPT-3 models use the same attention-based architecture as their GPT-2 predecessor. The smallest GPT-3 model (125M) has 12 attention layers, each with 12x 64-dimension heads. The largest GPT-3 model (175B) uses 96 attention layers, each with 96x 128-dimension heads.

GPT-3 expanded the capacity of its GPT-2 by three orders of magnitudes without significant modification of the model architecture — just more layers, wider layers, and more data to train it on.

Understanding the Data

following datasets:

Dataset	# Tokens (Billions)
Common Crawl (filtered by quality)	180.4
WebText2	55.1
Books1	22.8
Books2	23.65
Wikipedia	10.2

Notice GPT-2 1.5B is trained with 40GB of Internet text, which is roughly 10 Billion tokens (conversely assuming the average token size is 4 characters). So GPT-3 175B has a lower data compression ratio 300 / 175 = 1.71 in comparison to GPT-2 1.5G 10 / 1.5 = 6.66. This raises the question that, with this amount of parameters, whether the model functions by memorizing the data in the training and pattern matching in inference. We will have more discussions later in this article.

One novel challenge GPT-3 has to deal with is data contamination. Since their training dataset is sourced from the internet, it is possible that the training data will overlap with some of the testing datasets. Although GPT-2 has touched this topic, it is particularly relevant to GPT-3 175B because its dataset and model size is about two orders of magnitude larger than those used for GPT-2, creating increased potential for contamination and memorization.

gram overlap with anything in the training set". They then evaluate GPT-3 on these cleaned testing datasets and compare to the scores of the original "un-cleaned" datasets.

The comparisons show that the performance on most benchmarks changed negligibly. However, there are a few tasks that were significantly impacted by the data clean process. OpenAI flagged these tasks for further review.

Training the Model

GPT-3 is trained using next word prediction, just the same as its GPT-2 predecessor. To train models of different sizes, the batch size is increased according to number of parameters, while the learning rate is decreased accordingly. For example, GPT-3 125M use batch size 0.5M and learning rate of 6.0×10^{-4}, where GPT-3 175B uses batch size 3.2M and learning rate of 0.6×10^{-4}.

We are waiting for OpenAI to reveal more details about the training infrastructure and model implementation. But to put things into perspective, GPT-3 175B model required 3.14E23 FLOPS of computing for training. Even at theoretical 28 TFLOPS for V100 and lowest 3 year reserved cloud pricing we could find, this will take 355 GPU-years and cost $4.6M for a single training run. Similarly, a single RTX 8000, assuming 15 TFLOPS, would take 665 years to run.

Time is not the only enemy. The 175 Billion parameters needs $175 \times 4 = 700$GB memory to store in FP32 (each parameter needs 4 Bytes). This is one order of magnitude larger than the maximum memory in a single GPU (48 GB of Quadro RTX 8000). To train the larger models without running out of memory, the OpenAI team uses a mixture of model parallelism within each matrix multiply and model parallelism across the layers of the network. All models were trained on V100 GPU’s on the part of a high-bandwidth cluster provided by Microsoft.

In fact, The size of SOTA language model increases by at least a factor of 10 every year: BERT-Large (2018) has 355M parameters, GPT-2 (early 2019) reaches 1.5B, T5 the end, and model parallelization is going to be indispensable for researching SOTA language models.

Running Inference

This is where GPT models really stand out. Other language models, such as BERT or transformerXL, need to be fine-tuned for downstream tasks. For example, to use BERT for sentiment classification or QA, one needs to incorporate additional layers that run on top of the sentence encodings. Since we need one model per task, the solution is not plug-and-play.

However, this is not the case for GPT models. GPT uses a single model for **all downstream tasks.** Last year, OpenAI already showed GPT-2's potential as a turn-key solution for a range of downstream NLP tasks without fine-tuning. The new generation, GPT-3, uses a more formatted approach for running inference, and demonstrate even superior performance.

It uses a paradigm which allows zero, one, or a few examples to be prefixed to the input of the model. For example, in the few-shot scenario, the model is presented with a task description, a few dozen examples, and a prompt. GPT-3 then takes all this information as the context and start to predict output token by token. The situation is similar to zero-shot and one-shot; only the number of examples are reduced.

Let's use the task of English to French translation as a concrete example: the task description can be the sentence "Translation English to French." The few dozen examples may include text such as "sea otter => loutre de mer" and "peppermint => menthe poivree" etc. The prompt is the Enligsh word to be translated, for example, "cheese => ." Then the model is expected to output the French word for cheese, which is "fromage."

Results

Next, we briefly discuss the performance of GPT-3 using some of the downstream tasks.

model is asked to generate the rest of the story in a word by word fashion.

More precisely, GPT-3 is presented with a title, a subtitle, and the prompt word "Article: ." It then writes short articles (~200 words) that fools human most of the time. According to OpenAI's user study, "mean human accuracy at detecting articles that were produced by the 175B parameter model was barely above change at ~52%". Meaning humans will make random guesses while asking to detect GPT-3 generated articles. In contrast, the mean human accuracy at detecting articles produced by the smallest GPT-3 model (125M) is 76%.

This can be a big deal — "simply" increasing the size of the model by three orders of mangnitude is able to change something that is half-working into something non-distinguishable from human work. In plain English, this empirically shows that the number of model parameters, the FLOP/s-days and the number of training examples needs to grow according to a power function of the improvement of the model.

Of course, GPT-3 may still produce non-factual content (such as suggesting the popular U.S. TV program "The Tonight Show" is hosted by Megyn Kelly instead of Jimmy Fallon), nor did OpenAI claim the model is ready for writing the last two books of "A Song of Ice and Fire." Nonetheless, getting closer to the finishing line of the Turing test for writing short articles is significant, and will no doubts have its great impact on our social media.

General NLP Tasks

Although writing a new article is cool, the killer feature of GPT-3 is the ability to be 're-programmed' for general NLP tasks without any fine-tuning. This is where OpenAI's real ambition lies: having a model to do just about anything by conditioning it with a few examples.

The paper showed a dozen of downstream tasks, ranging from the usual players such as machine translation and question and answer to the unexpected new tasks such as arithmetic computation and one-shot learning of novel words. Instead of

What is the role of the few examples that fed to GPT-3 model before it makes predictions? Do more examples improve the results?

One way to think about these examples is that they "teach" the model how to do reasoning. It would be amazing if this is really the case, because it shows that the model can indeed be reprogrammed for new tasks very quickly. However, it is not clear how such a reprogramming process works under the hood. GPT-3 tends to perform better on language modeling tasks and less well on reasoning tasks. For example, GPT-3 really thrives in the task of machine translation, especially when the target language is English. It even beats the fine-tuned SOTA in the tasks of WMT Fr->En and WMT De->En. On the other hand, GPT-3 performs significantly less well than fine-tuned SOTA in SuperGLUE's BoolQ task. This task asks the model to read a short passage and then answer a related True/False question. The 15% gap between the fine-tuned SOTA and GPT-3 few shots seems to suggest that model isn't particularly strong in terms of conducting reasoning based on a passage that was not seen in the training.

Another interesting view is that these examples function as "filters" that let the model search for highly relevant context or patterns from the dataset. This is possible because the dataset is practically compressed into the weights of the network. Examples that have a strong response to the filters are then "interpolated" to produce the output of the model. Obviously, the more examples you give to the model, the more precise the filter becomes, and in consequence, the better the results.

At this stage, I found the second explanation probably makes more sense. Language models are designed to generate readable texts. They do not have a deep "understanding" of the physical world, nor are they trained to do sophisticated reasoning. Think about how we get to understanding the world — reading newspapers and novels is not enough. Otherwise, there will be no need to study math, physics, engineering, etc.

100000 * 100000 = 10 Billion different combinations for five digits addition. Every example takes at least five tokens (the two input numbers, the plus sign, and the equal sign and the output number). So there it requires least 5 Billion tokens to store 10% of the examples. The entire training dataset has 300 Billion tokens. So to argue the network is purely memorizing the training data, there should be at least 5 / 300 ≈ 1.7 % of the training data are five-digits addition. I honestly don't think we see five-digits addition appear that often in my daily life. This indicates the network, at a certain degree, was learning to work with numbers instead of memorizing their combinations.

The Next Level

GPT-3 has generated a lot of discussion on Hacker News. One comment I found particularly intriguing compares human brain with where we are with the language models: A typical human brain has over 100 trillion synapses, which is another three orders of magnitudes larger than the GPT-3 175B model. Given it takes OpenAI just about a year and a quarter to increase their GPT model capacity by two orders of magnitude from 1.5B to 175B, having models with trillions of weight suddenly looks promising.

If GPT-2 was "too dangerous to release," and GPT-3 almost passed the Turing test for writing short articles. What can a trillion parameter model do? For years the research community has been searching for chatbots that "just works," could GPT-3 be the breakthrough? Is it really possible to have a massive pre-trained model, so any downstream tasks become a matter of providing a few examples or descriptions in the prompt? At a broader scale, can this "data compilation + reprogram" paradigm ultimately lead us to AGI? AI safety needs to go a long way to prevent techniques like these from being misused, but it seems the day of having truly intelligent conversations with robots is just at the horizon.

Footnotes

over the past two years.

3. With the exception that GPT-3 use alternating dense and locally banded sparse attention patterns in the layers of the transformer, similar to the Sparse Transformer used in "Generating long sequences with sparse transformers", Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever, 2019

4. The accuracy measures how successfully human can detect machine generated articles. Lower detection rate means better performance of the GPT model. The theoretical best case would be human are making random guess at a 50% successful rate, as he/she can not distinguish what is real from what is fake.

5. GPT-3 shows that it is possible to improve the performance of a model by "simply" increasing the model size, and in consequence the dataset size and the computation (TFLOP) the model consumes. However, as the performance increases, the model size has to increase more rapidly. Precisely, the model size varies as some power of the improvement of model performance.

6. We double V100's theoretical 14 TFLOPS of FP32 to get its theoretical 28 TFLOPS of FP16. Notice **optimized** ML models can leverage V100's Tensor Cores (112 TFLOPS) to further speed up the training. However, we do not expect to see 8x speedup from Tensor Core. Because the real speed up from mixed precision training in comparison to FP32 is usually shy of 2x for **image models**, and upto 4x for **language models** that have high sparsity, even with implementations that highly optimized for Tensor Cores.

[SHARE ARTICLE]

in 𝕏 ✉

{ FOOTER }

AI FACTORIES

//FOR EVERY MISSION

SUPERINTELLIGENCE

ENTERPRISE

GOVERNMENT

STARTUPS AND RESEARCHERS

//FOUNDATIONS

AI INFRASTRUCTURE

TRUST AND SECURITY

CUSTOMER STORIES

PRODUCTS

//PRODUCTS

SUPERCLUSTERS

1-CLICK CLUSTERS

INSTANCES

//DOCS

DOCUMENTATION

BLOG

RESEARCH

//FEATURES

AI INFRASTRUCTURE

ORCHESTRATION

LAMBDA STACK

TRUST AND SECURITY

COMPANY

//INSIDE LAMBDA

ABOUT

CAREERS

LEADERSHIP

INVESTORS

//RESOURCES

RESEARCH

CUSTOMER STORIES

BLOG

PARTNERS

BRAND GUIDELINES

PRIVACY POLICY

TERMS OF SERVICE