Issue Brief

Scaling AI

Cost and Performance of AI at the Leading Edge

Author
Andrew J. Lohn

CSET CENTER for SECURITY and EMERGING TECHNOLOGY

December 2023

Executive Summary

Historically, the progress of artificial intelligence (AI) has been mercurial, alternating between surprising breakthroughs and periods of comparative drought. While that will continue to be true to some extent, the incentives and trends are clearer than they have been, and there is even a subfield dedicated to mathematically predicting how much progress to expect from additional investments.

These predictions suggest that further performance gains will come from increasing the scale of investment in the current approaches, but that there are sharply diminishing returns. For example, simply increasing the computing budget from $10 million to $100 million increased the pass rate for AI-generated computer programs from about 65% to about 75%. A billion-dollar version of the model would apparently only reach about 80%, and a trillion-dollar version only 90%. However, the record pass rates already exceed those numbers because users are more inventive in how they apply existing models. This highlights how researcher ingenuity can outperform large-scale investment.

Continued investment still has its place. Even shrinking jumps in performance from scaling may continue to justify the rising costs. It may also be that only relatively small improvements in performance "unlock" valuable or risky capabilities that justify policymaker intervention to either enable or avoid them.

There are initial signs, though, that these diminishing marginal returns are already dampening the drive for ever-larger models. The growth in compute to train the largest models appears to be slowing. Among publicly available models, users and developers prefer to download intermediate-scale models even when larger and more powerful ones are freely available from the same providers.

These trends raise questions about the types of models that will be most impactful and the relative importance of the compute, data, and algorithms that governments might hope to control. They also raise questions about the need for policymakers to intervene to promote or impede that progress and their opportunities to do so.

Introduction

The trajectory of AI progress is more contentious than ever, with both utopian and dystopian visions of where the current path leads. On the pessimistic side are arguments that AI is becoming an existential threat to be viewed alongside pandemics and nuclear war.[1] This leads to a suite of policy questions such as: Should companies agree to a six-month moratorium on further growth?[2] And should governments do more to block the sale of specialized hardware to rival nations?[3] More optimistically, others argue that AI will be a net economic and security boon. This leads to other questions such as whether nations should pay to develop the most powerful models rather than relying on companies.[4] And should governments do more to provide computing hardware to their own citizenry, either by provisioning a nationalized center or subsidizing computing hardware development?[5]

Many of these questions hinge on the idea that there will be substantial progress in AI by simply increasing the computing resources allocated to it. Recent history would certainly support that, but given the amount of investment required and the potential impact of the capabilities being discussed, it is worth deeper investigation. This report provides some baselines for anticipating future progress, as well as what shifts to watch for in the field of AI. It also introduces the tools and data for making these estimates.

This report starts by describing the current scale of AI models. It then describes how increases in scale to this point have been the driving force for recent progress but also shows that the growth in resources being allocated to the most compute-intensive models is slowing. That is likely because there are diminishing returns to further investment. This report first shows the increasing cost of progress based on the costs to train larger AI models, then illustrates the increasing cost to run those larger AI models. Furthermore, these costs may potentially drive users and companies toward smaller models even when larger ones could be created or are already available. This report's analysis suggests that users may already be choosing smaller models based on the download counts from a popular repository. Finally, it briefly discusses what these various economic and empirical trends might mean for future progress in AI, noting that scaling up resource allocation is not the only avenue to continued progress in AI.

What Makes a Model Large?

This report focuses on "Large Language Models" (LLMs) but there is also no official definition for what counts as "large." The "large" abstractly refers to any of several traits such as the number of parameters in the model, the amount of memory required to hold it, the amount of computation needed to train or run it, or the number of dollars needed to do those computations. And there is no clear threshold that separates "large" models from "small" models.

The small end of large

A rough division between large and small models is one billion parameters. GPT-2 (an openly available predecessor to the engine behind ChatGPT) comes in four sizes, the largest of which crossed this threshold with 1.5 billion parameters.[6] It was the first major model to be withheld for fears about its potential for misuse. However, aside from one billion being a convenient round number, models with many more parameters than that often do not fit on a single GPU, which makes them more cumbersome and expensive to use.[7]

The one billion parameter threshold is not a hard rule. Some would also consider models with fewer than one billion parameters to be "large," while others might only consider a model large if it requires multiple processors to train and operate.

The large end of large

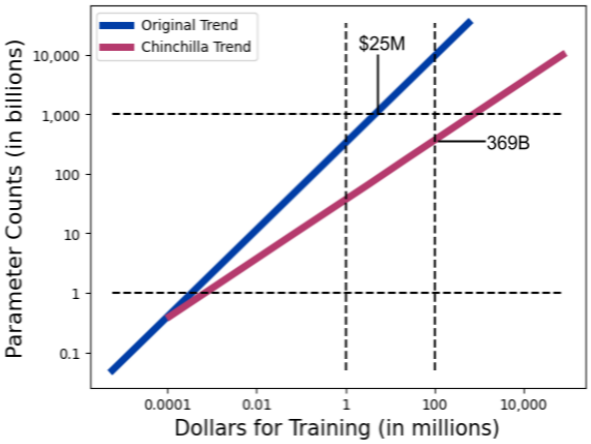
If one billion is the minimum threshold to be considered large, then what is the upper end of large? Although there is certainly an exact answer for how large the current largest model is, AI companies treat that information as proprietary. There are a variety of clues and recent trends to try to draw from, though.

Some models, known as mixture of experts (MoE), have over a trillion parameters, but they are effectively several smaller specialized models stitched together so that a query can be directed to the most appropriate smaller specialized model. The largest publicly known single models (referred to as "dense" models) in terms of parameter counts are in the range of Google's PaLM with 540 billion parameters and Nvidia's Megatron-Turing NLG with 530 billion parameters.[8] However, these are both proprietary models. Until recently, the largest openly available models were Meta's OPT-175B or Hugging Face's BLOOM 176B.[9] But they were not state of the art in

terms of performance because a smaller model can outperform a larger one if it is trained for longer on more data.

In 2021, Google created a model called Chinchilla that used only 70 billion parameters paired with more training data to achieve increased performance. Before the Chinchilla paper, it seemed that trillion parameter models were just around the corner. Now the top-performing models often have fewer parameters than ones from years prior. This shift makes models that are easier and less expensive to use. It also means that parameter counts are not a great way to compare the performance of models. A better metric is how much compute was used to train them. Figure 1 shows the original trend for parameter counts with respect to compute budgets and the updated one based on insights from the Chinchilla paper. The current guiding trends, which are not public, are likely below the Chinchilla trend.

Figure 1. Parameter Counts Correlate with Training Compute



Source. CSET.

According to the earlier trend in Figure 1, a trillion-parameter model would have cost about $25 million. For comparison, GPT-3 and BLOOM cost single-digit millions of dollars.[10] Using the Chinchilla approach, a trillion-parameter model would cost about $650 million to train but would be much more capable than a $25 million trillion-parameter model trained under the original trend.[11] In Appendix A, we estimate that the $100 million that OpenAI claims to have spent on GPT-4 would put it at between 369 and 520 billion parameters.[12] In practice, GPT-4 probably uses several models, but each is probably smaller than this estimate because smaller models are less costly to operate. The question is then, with their only somewhat larger parameter counts, how much better are today's models, and how much better again might the next generation be?

The Performance Benefits of Scaling

An entire subfield has developed to predict the performance benefits of scaling AI models, and they have found simple mathematical equations that describe how the models improve when additional resources are applied to training.[13] A common aspect of the model is its "loss," which describes how much it deviates from the desired answers, but there are many other more intuitive aspects such as pass rates for math or programming challenges.

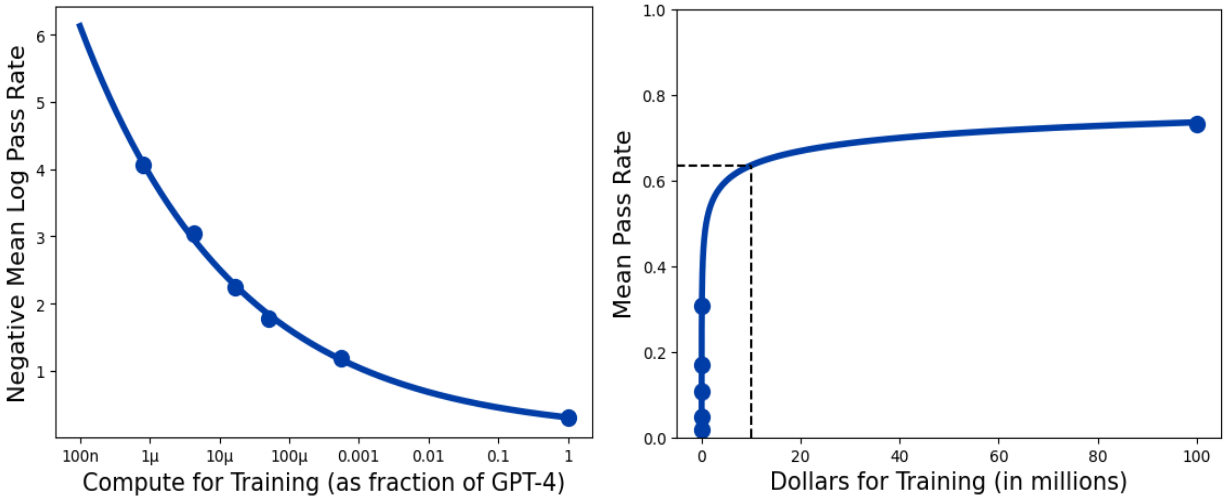
The equations are typically power laws, where the amount that a capability improves is related to the amount of some resource, such as compute or data, raised to an exponent. This report focuses mainly on compute as the resource and presumes that enough data is available for compute to be the limiting factor.

The existence of the power law implies that designers and funders can know some of the capabilities of a future system before it is made. They can make a series of small models and use those to predict the performance of a much larger model before deciding whether to build it.[14] This does not mean that all capabilities can be predicted. Power laws have not been found for all capabilities of interest, and there is conflicting evidence and debate as to what fraction of capabilities can be described by these equations even in theory. This report will return to that debate in a later section about emergence.

Diminishing marginal returns

As one specific example, OpenAI examined a range of models' abilities to solve programming problems.[15] As models grow, their performance improves according to the power law, but the OpenAI paper defines performance using a slightly complicated measurement called negative mean log pass rate.[16] Figure 2 reproduces the original graph from the OpenAI paper on the left side. The right side of the figure shows the exact same graph but changes the axes to be more intuitive by converting the x-axis to dollars spent on training (assuming $100 million for GPT-4) and converting the y-axis to the probability that each program it writes will succeed at solving the problem.[17]

Figure 2. Scaling GPT for Programming Has Diminishing Returns



Source. CSET analysis of data from the GPT-4 Technical Report.[18]

Converting to intuitive axes shows that there are diminishing marginal returns to increasing the computing budget. It suggests that a ten-million-dollar model would generate solutions that succeed about 64% of the time and this would increase to 74% for a one-hundred-million-dollar model.[19] Extrapolating beyond the data in Figure 2, a trillion-dollar model would increase those odds to just over 91%.

But simply scaling up the models to a trillion dollars would not lead to record performance. At the time of writing, the record on this programming challenge is 94.4%.[20] That top performer uses GPT-4 but adapts it for improved programming performance. Presumably, combining these adaptations with a model scaled beyond GPT-4 would lead to further performance gains, but currently, the gains are coming from being more inventive in using the models that already exist rather than simply scaling.[21]

Emergence

This predictability appears to be in contrast to what has been termed as emergent abilities. That is when scaling up models only leads to little performance improvement at first, but then appears to spike suddenly as models or compute budgets cross some threshold.[22] These emergent abilities have gained substantial interest but it appears that many of them are artifacts of the way that the ability was measured rather than

being a true property of increasing scale.[23] A common way for abilities to appear to emerge is when the answer has several parts and the model has to get them all correct. For example, in a multiplication problem, the model writes the answer one number at a time. If it gets any of those digits wrong then the whole answer is wrong. It does not get more credit for getting 80% of the digits than for getting 30% of them. As a result, the ability to multiply can appear to emerge suddenly once it is common to get all the digits correct, even if the ability to write each digit correctly only increases by a small amount.

It may be that many complex tasks can be suddenly "unlocked" by small increases in model performance. Writing a best-selling novel, or perhaps writing original malware, are complex tasks and may not be easy to predict based on the error rate of a model during training. But it may also be that even these "emergent" abilities are more predictable than they seem. Although important tasks are often complex, it may be possible to break many of them down into simpler subtasks that are predictable by power laws. More research should be dedicated to trying to decompose important but complex tasks into predictable subtasks. That decomposition could be used to identify the performance thresholds that might unlock valuable or risky capabilities. This could provide much needed guidance to justify either increasing investments or imposing restrictions.

Even if these models are powerful enough to justify the increasing costs for training, that is not the only cost they have to contend with. Larger models are also more expensive to run once they are trained.

Cost of Operation

Operational costs depend on many factors such as the length and number of the input queries and how quickly the responses need to be written. Following the Chinchilla trend, inference costs grow with approximately the square root of the training costs.[24] That means spending 100 times more on training leads to a model that costs 10 times as much to operate. In terms of scaling analyses, a square root is considered a slow growth in cost, so this is potentially good news for companies looking to market powerful models. Still, bigger models are more expensive to operate and that cost growth can be significant. In Appendix B, we use the Chinchilla trend to derive the square root relationship and show the result in Eq A2.

$$Cost_{Inference} \approx 2.35x10^{-5}\sqrt{Cost_{Training}} \qquad \text{Eq 1}$$

As a rough estimate, and presuming again that GPT-4 cost $100 million to train, this equation would suggest that outputs cost about $0.24 per thousand tokens.[25] If that is true then OpenAI might only be recovering half the cost of running it (they charge $0.12 per thousand tokens), which is consistent with early reporting that OpenAI was losing money with each output.[26] It is also consistent with Google's published inference costs.[27] However, there are strong economic incentives to reduce operational costs.

As mentioned earlier, companies probably make models that are smaller than the Chinchilla trend would project. That lends credence to the rumors that GPT-4 is actually a 220 billion parameter mixture of experts model rather than an approximately 500 billion parameter dense model as Chinchilla would imply.[28] Companies can also reduce costs by reducing the number of bits allocated to each parameter. Using these two effects brings operational costs down by a factor of six to about $0.04 per thousand tokens, which is consistent with Open AI CEO Sam Altman saying that each chat costs single-digit cents.[29]

Companies will continue to work hard to find ways to reduce operational costs while continuing to improve performance. They may need to continue to increase the model sizes to achieve those performance gains, but there is a strong financial incentive against it. If a smaller model can do the job, then it will be preferred.

Updating the Trends in Scaling

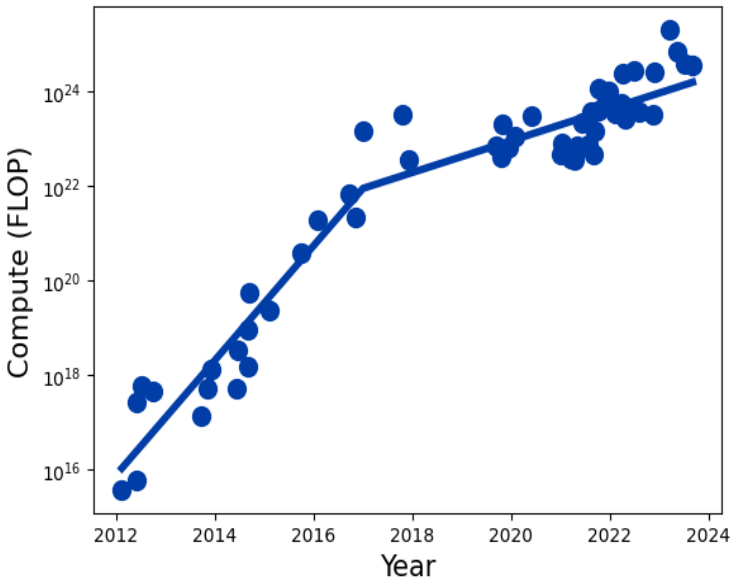
Plans for building the largest future models are closely guarded, but some trends and incentives can provide clues about what to expect.

Slowing growth

Investment in training state-of-the-art models has been growing rapidly for about ten years but it appears to be slowing down.[30] OpenAI's famous GPT series is an example of this trend. The time between the first and second GPTs was about 250 days. Then the third took about 500 days. GPT-4 came about 1,000 days later. Now OpenAI claims to not be training a GPT-5, which is believable given the level of investment required, the diminishing marginal returns, and this slowing historical trend.[31]

This slowdown can be seen most clearly from a dataset that attempts to track the compute demands of notable models. [32] Figure 3 shows only those models that used at least 10% as much compute as the previous record holder. Although compute demands continue to climb, the data matches far better to a kinked line with a slowdown than a straight one.[33]

Figure 3. Training Compute Is Slowing for the Largest AI Models



Source. CSET analysis of Epoch AI data.[34]

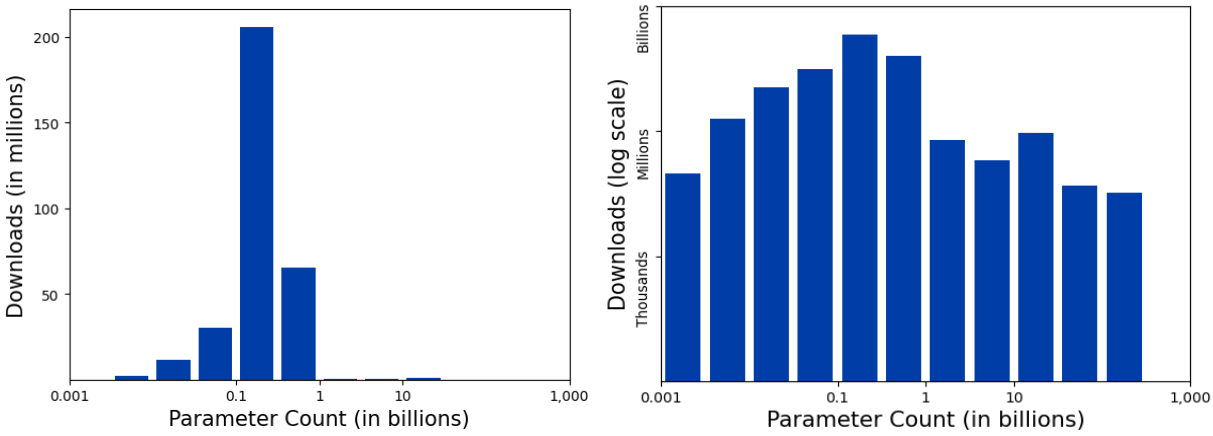
It is still possible for developers to train models that are substantially larger than the state of the art. It is unlikely, however, that anyone could train a model large enough that it could continue the trend observed for most of the 2010's.[35] It is also possible that the trend for all models, rather than just the most compute-intensive ones, is growing more smoothly, albeit at a lower total magnitude. That is a difficult question to test because it requires a dataset that is representative of all models.

User Preference for Smaller Large Models

There may also be less demand for model growth among users than previously anticipated. Hugging Face is a popular repository that has become the primary source for publicly available large models and they provide their download statistics for analysts to inspect.[36] Hugging Face offers powerful models up to hundreds of billions of parameters but the most capable models are proprietary, so the data may be biased against users who need the highest performing models. Hugging Face users tend to choose smaller models even when larger, more capable ones are available. As of July 2023, the two most popular language models had only 110 and 561 million parameters, a thousand times smaller than the state of the art, but each attracted over 40 million downloads per month.[37]

Overall, models below a billion parameters were far more popular both in terms of the number of models provided and the number of models downloaded per month. Figure 4 shows the total number of downloads for models grouped by their parameter counts. On the left, it is difficult to even see the download counts for models above a billion parameters. To make the download counts more visible, we reproduce the same graph but with a logarithmic y-axis on the right. Large models seem to be hundreds or thousands of times less popular, with download counts in the tens or hundreds of thousands compared to hundreds of millions for models just below a billion parameters.

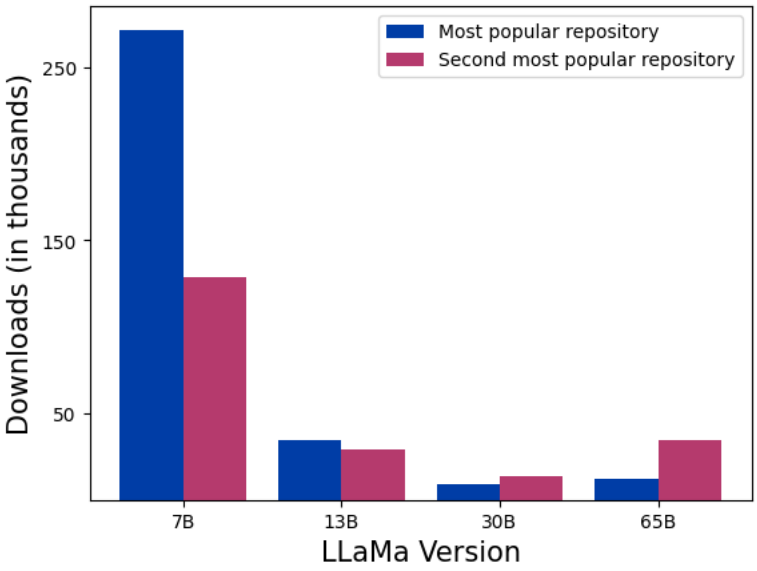
Figure 4. Small Hugging Face Models Are More Popular



Source. CSET analysis of Hugging Face data.[38]

As another example, Hugging Face offers all of the LLaMa models. The original LLaMa version 1 models came in four sizes are: 7B, 13B, 30B, and 65B, where B stands for billions of parameters. LLaMa2 comes in 7B, 13B, and 70B sizes. The smallest LLaMa model (7B) is by far the most popular, getting 25 times as many downloads as the 65B version, as shown in Figure 5. One possible reason is that the 7B parameter model performs adequately well, particularly for more narrow applications, and can readily run on a single GPU whereas the larger models typically require multiple GPUs. The larger models are more complex and expensive to operate. They can also be slower to run, which is not acceptable for all applications.[39]

Figure 5. Small LLaMa Models Are More Popular



Source. CSET analysis of Hugging Face data.[40]

Some Applications Do Not Benefit from Scaling

Another possible reason for the popularity of small models is that some applications do not benefit as much from scaling either because they are intrinsically less complex or because the constraints of their application cannot accommodate large models. This may be more the norm than the exception as seen in a recent survey.[41] The vast majority of researchers were working on tasks other than developing general-purpose language, imagery, or multimodal models that require such extreme scaling. Among top researchers across academia and industry, there was a wide range of compute demands over several orders of magnitude.

The smaller models may also be popular because of limited data availability. For example, in many translation applications, it seems that there is not enough data to justify training larger models.[42] This is especially true for what are called "low-resource" languages which have little written text for translation, but it also applies to high-resource languages. Model sizes and compute budgets have remained modest while the field has focused more on ways to get past the data limitations.[43] This is likely just one of many examples of a specialized task where a lack of data limits the value of scaling, but even for general-purpose AI, the demand for quality data might outstrip the internet.[44]

Distillation

The field might also trend toward smaller models because of a process called distillation. Distillation uses the capability of a larger model to train a smaller one to have similar performance, though perhaps for a narrower range of applications. A prominent example was when Stanford researchers used just a few hundred dollars' worth of outputs from ChatGPT to fine-tune the smallest LLaMa model (7 billion parameters). They called this fine-tuned model Alpaca and claimed that it performed comparably to ChatGPT according to human evaluators on a diverse range of tasks.[45]

It is still an open question how effective distillation can be. After the release of Alpaca, other researchers showed that it was learning the style of writing more than the content, so it was not actually performing as comparably as initially suspected.[46] But then further research improved on the technique for distilling content, closing the gap again.[47] This back and forth may continue for some time.

Ultimately, distillation of larger models to smaller ones is certainly possible to some degree and likely also has some limits. It may be that very broad capabilities such as general-purpose question answering require many parameters to store a lot of information. Those capabilities may be difficult to distill, but distillation may come with fewer sacrifices for more narrowly-scoped tasks.

To capitalize on the efficiency of distillation, companies might train a large, highly capable general-purpose model and then use it to train many smaller models that are each more narrowly scoped and less expensive to operate. A challenge though is that if smaller models can be trained using a relatively small amount of output, such as the few hundred dollars spent for Alpaca, then users or other companies may also make their own distilled models. These copycat versions could undercut the company that built the original model, leaving them holding the bill for the initial training runs.[48] In that way, distillation may help justify large initial investments to train a model, but it may also make those investments hard to protect. These dynamics create some uncertainty about the future of scaling in AI.

Expectations for the Future of Scaling

Increased scaling will certainly lead to increased performance. That is the historical trend, and it will be a continuing trend, but continued scaling is increasingly expensive. Among the very largest models, that expense has likely already changed the trajectory of the trend, slowing it considerably. Companies appear to be more reluctant to increase their spending, and users are showing some preference for smaller models even when larger ones are freely available.

The equations that predict how much progress to expect from additional investment suggest that there are sharply diminishing returns, at least in fundamental technical performance. That does not necessarily mean that the investments are not worthwhile. It may be that relatively small improvements in outputs are especially valuable. It does, however, mean that the proposed value or risk of that additional investment deserves extra scrutiny.

From a more holistic perspective, there is a question of whether doubling down on the current compute trajectories is the path forward for AI. Research dollars can be allocated to the computers that train today's models at tomorrow's scale, or they can be allocated to discovering better model designs or training procedures or more inventive ways of using the models that already exist. Growing today's models will certainly provide advances, but investment in fundamental research might lead to more efficient solutions. Fortunately, although there are many unknowns and uncertainties, these decisions can be at least partly guided by hard numbers.

Appendix A: Cost Estimates for Trillion Parameter Models and GPT-4

Full price A100 GPUs are $32.77 per hour for 8 or $11.57 per hour for 8 on three-year reserve. Then $100 million buys between 24.4 million and 69.1 million GPU hours. Assuming GPUs do 163 teraFLOPS each results in between 1.43E25 and 4.06E25 FLOP. Using the Chinchilla trend, that implies between 327 billion and 550 billion parameters. That is more or less consistent with Meta CEO Mark Zuckerberg saying that GPT-4 and PaLM are about ten times larger than LLaMA.[49] The largest LLaMa model has 65 billion parameters, so ten times that is about 650 billion parameters.

Appendix B: Derivation of Inference Costs

The number of parameters is related to the computing cost for training according to the optimization laws and the price of compute. We multiply the quantity of compute by the three-year reserve pricing (cheapest consistent option) to estimate the cost of compute for training. We then do a simple linear fit to the data from the Chinchilla optimization law for log of parameter counts (N) vs. log of compute cost for training (C_T) to get the following equation.[50]

$$log(N) = 7.77 + 0.498 log(C_T)$$	Eq A1
$$N = 10^{7.77} {C_T}^{0.498}$$	Eq A2

The cost of inference (C_I) is significantly more complicated. Inference can be limited by the act of conducting the computations, the act of loading the parameters into memory, or the communications between processors. Which of these is limiting depends on many variables related to the model's architecture, the number of tasks that can be run in the same request (batch size), the number of processors that share the task, and others.

Currently in practice, loading data into memory is often the most constraining step.[51] In general, each of the steps is likely to create similar delays because there is little to gain from accelerating one step if another is the bottleneck. For that reason, calculating the delays for one step is often a reasonable approximation for the others. Here we will calculate the delays (and therefore compute costs) for loading the parameters into

memory because it is often the primary bottleneck and because it involves conveniently few variables as shown in Eq A3.[52]

$$C_I = \frac{2N}{A_{mb}} \frac{p}{3600} \quad \text{Eq A3}$$

The time to load the parameters to memory is approximately the number of parameters (N) times two for the two bytes per parameter (in half precision), divided by the memory bandwidth (A_{mb}). That number is in seconds, so to convert to cost we divide by 3600 to convert it to hours and multiply by the price per hour for a single GPU (p). If multiple GPUs are used then the time would be divided by that number of GPUs but the price would also be multiplied by that number of GPUs, so the number of GPUs is dropped from the cost equation. We multiply the inference cost by 1,000 because pricing is typically given per thousand tokens generated.

Substituting Eq A2 into Eq A3 provides the inference cost as a function of the training cost.

$$C_I = 1000 \frac{2 * 10^{7.77} {C_T}^{0.498}}{A_{mb}} \frac{p}{3600} \quad \text{Eq A4}$$

Then we substitute $p = 11.57/8 = 1.446$ from Amazon's GPU pricing, and $A_{mb} \approx 2GB/s$ from Nvidia's documentation for the A100 GPU. And rounding the exponent 0.498 to 0.5,

$$C_I = 2.35x10^{-5}\sqrt{C_T} \quad \text{Eq A5}$$

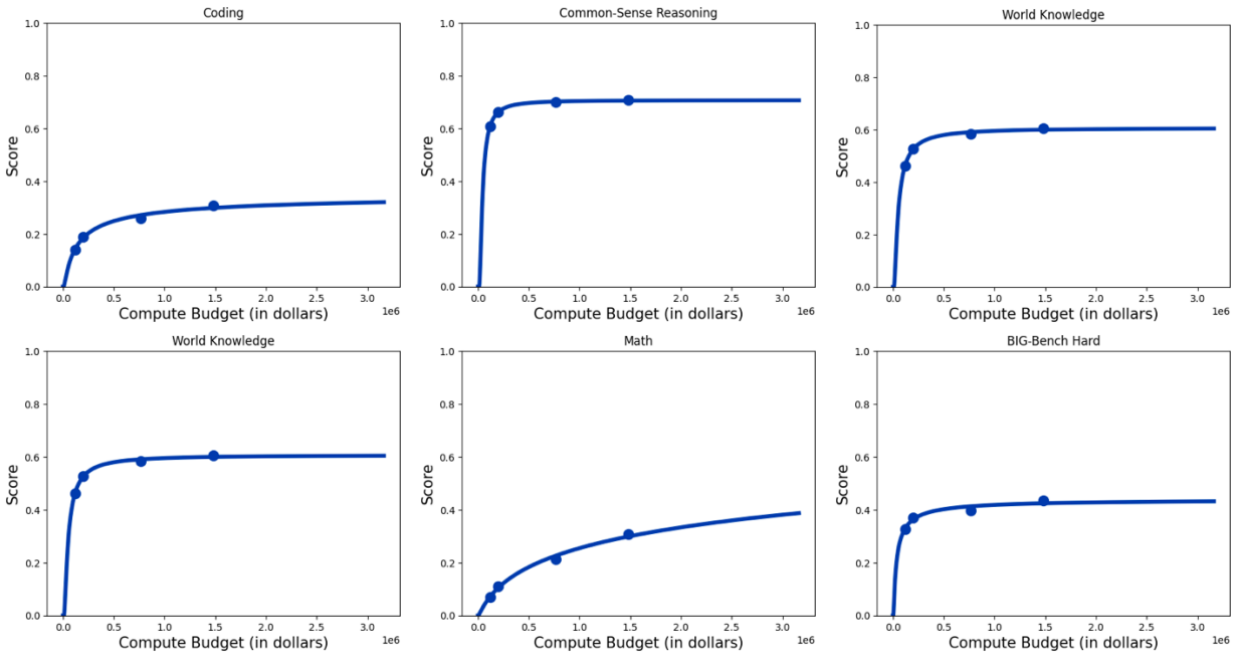
There are substantial incentives for model developers to reduce these inference costs so it is reasonable to expect the costs to decrease over time as developers find various cost reduction techniques. Some examples techniques are training smaller models than the Chinchilla optimization would imply, model quantization, and distillation.

Appendix C: Power Laws for LLaMa Benchmarks

Here we show the performance and fits to it for LLaMa1 across various benchmarks as reported in the LLaMa2 paper.[53] LLaMa2 is not used because they appear to have been inconsistent in their training for the 34B version. Excluding the 34B version leaves only three data points to fit three power law parameters. As it is, the four versions of LLaMa1 provide only one parameter more than the minimum for fitting. We expect that there is a lot of error in these fits which is why they are relegated to this appendix and not detailed in the body of the report.

The performance on some of these benchmarks should not be interpreted as the best that is possible. Further performance gains may still be possible where they appear to be saturated. For example, a task may be data limited and performance could improve with scale if the training set was adapted to have more data for that task.

Figure D1. LLaMa1 Across Sizes and Benchmarks



Source. CSET analysis of Meta data.[54]

Author

Andrew Lohn is a Senior Fellow at Georgetown University's Center for Security and Emerging Technology (CSET) and the Director for Emerging Technology on the National Security Council Staff, Executive Office of the President under an Interdepartmental Personnel Act agreement with CSET. Dr. Lohn completed this work before starting at the National Security Council. The views expressed are the author's own personal views and do not necessarily reflect the views of the White House or the Administration. During the preparation of this brief, Andrew Lohn also participated in the red-teaming of Meta's and OpenAI's large language models for which he was compensated.

Acknowledgments

The author would like to thank Tamay Besiroglu and Lennart Heim for their probing reviews. The author would also like to acknowledge Rebecca Gelles, James Dunham, and Maggie Wu for their help with data collection, analysis, and code review. The author also appreciates John Bansemer's direction and support.

Document Identifier: doi: 10.51593/2023CA006

Endnotes

[1] Center for AI Safety, "Statement on AI Risk," May 30, 2023, https://www.safe.ai/statement-on-ai-risk.

[2] Future of Life Institute, "Pause Giant AI Experiments: An Open Letter," March 22, 2023, https://futureoflife.org/open-letter/pause-giant-ai-experiments/.

[3] Emily S. Weinstein, Kevin Wolf, "For Export Controls on AI, Don't Forget the "Catch-All" Basics," CSET, July 5, 2023, https://cset.georgetown.edu/article/dont-forget-the-catch-all-basics-ai-export-controls/.

[4] Ben Wodecki, "Meet BLOOM: The Most Important AI Model of the Decade," AI Business, July 18, 2022, https://aibusiness.com/nlp/meet-bloom-the-most-important-ai-model-of-the-decade-#close-modal.

[5] National Artificial Intelligence Research Resource Task Force, "Strengthening and Democratizing the U.S. Artificial Intelligence Ecosystem," January 2023, https://www.ai.gov/wp-content/uploads/2023/01/NAIRR-TF-Final-Report-2023.pdf.

[6] Alec Radford, et. al., "Language Models are Unsupervised Multitask Learners," February 14, 2019, https://openai.com/research/better-language-models.

[7] There are many ways for models with more than a billion parameters to fit in a single GPU, especially GPUs that have large memory capacity. The models can use small input sizes or small batches or fewer bits per parameter for example.

[8] Aakanksha Chowdery, et. Al., "PaLM: Scaling Language Models with Pathways," arXiv:2204.02311, April 5, 2022, https://arxiv.org/abs/2204.02311; Mohammad Shoeybi, "Megatron-LM: Training Mulit-Billion Parameter Language Models Using Model Parallelism," arXiv:1909.08053, September 17, 2019, https://arxiv.org/abs/1909.08053.

[9] Susan Zhang, et. al., "Democratizing access to large-scale language models," Meta AI, May 3, 2023, https://ai.meta.com/blog/democratizing-access-to-large-scale-language-models-with-opt-175b/; BigScience Workshop, "BLOOM: A 176 Billion Parameter Open-Access Multilingual Language Model," November 9, 2022, https://arxiv.org/abs/2211.05100.

[10] BLOOM cost $7 million according to reporting in Kyle Wiggers, "A year in the making, BigScience's AI Language Model is Finally Available," https://techcrunch.com/2022/07/12/a-year-in-the-making-bigsciences-ai-language-model-is-finally-available/; Nvidia's blog (https://developer.nvidia.com/blog/scaling-language-model-training-to-a-trillion-parameters-using-megatron/) indicates that training GPT-3 would require 1024 A100 GPUs for a month. At $32.77 per hour for 8 A100 GPUs on AWS, that works out to about $3 million.

[11] The Chinchilla trend suggests 1.27E26 FLOP (See Table 3 in https://arxiv.org/pdf/2203.15556.pdf) for a 1 trillion parameter model. At $32.77 per hour for 8 A100 GPUs on AWS, and 163 teraFLOPS per GPU, that works out to about $649 million. This is for standard pricing; cloud services offer about a three times discount for long term contracts.

[12] OpenAI uses Microsoft's cloud which offers a 50% discount to customers who commit to three years: Microsoft, "Azure Machine Learning Pricing," https://azure.microsoft.com/en-us/pricing/details/machine-learning/.

[13] Tom Henighan, "Scaling Laws for Autoregressive Generative Modeling" arXiv:2010.14701, November 6, 2020, https://arxiv.org/pdf/2010.14701.pdf; See Eq 10 in Jordan Hoffman, et. al., "Training Compute-Optimal Large Language Models," arXiv:2203.15556, March 29, 2022, https://arxiv.org/pdf/2203.15556.pdf.

[14] In appendix C, we show how it applies to the LLaMa series of models for the benchmarks they evaluated.

[15] OpenAI, "GPT-4 Technical Report," arXiv:2303.08774, March 27, 2023, https://arxiv.org/pdf/2303.08774.pdf.

[16] The dataset of programming problems has 6 difficulty levels. They used the 23 problems in the third easiest difficulty level and had each model size generate many attempted solutions for each problem. They determined the fraction of attempts that succeeded at solving each problem then took the logarithm of that number. Finally, they averaged the result across the 23 problems.

[17] The GPT4 Technical Report uses units of fraction of the compute needed to train GPT4. We multiply those numbers by the estimated $100 million spent training GPT4 to convert those units to dollars.

[18] OpenAI, "GPT-4 Technical Report," arXiv:2303.08774, March 27, 2023, https://arxiv.org/pdf/2303.08774.pdf.

[19]The 74% for GPT-4 level investment is slightly higher than the 67% reported for GPT-4 on the benchmarking site Papers With Code. The discrepancy could be from Papers With Code using the entire programming dataset rather than just the fraction used in the technical report.

[20] Papers With Code, "Code Generation on HumanEval," https://paperswithcode.com/sota/code-generation-on-humaneval, retrieved Nov 6, 2023.

[21] To reach 94.4% pass rates by pure scaling along the trend from the GPT-4 technical report, assuming that GPT-4 cost $100 million, would cost over a quadrillion dollars.

[22] Jason Wei, et. al., "Emergent Abilities of Large Language Models," arXiv:2206.07682, June 15, 2022, https://arxiv.org/abs/2206.07682.

[23] Rylan Schaeffer, et. al., "Are Emergent Abilities of Large Language Models a Mirage?," arXiv:2304.15004, May 22, 2023, https://arxiv.org/pdf/2304.15004.pdf.

[24] Pablo Villalobos, David Atkinson, "Trading Off Compute in Training and Inference," Epoch, July 28, 2023, https://epochai.org/blog/trading-off-compute-in-training-and-inference.

[25] Language models have a vocabulary of tokens rather than words. The tokens can be words but they can also be many other things such as parts of words, numbers, or punctuation. The price per token can be very small, so prices are usually quoted in thousands of tokens instead.

[26] Will Oremus, "AI chatbots lose money every time you use them. That is a problem." *The Washington Post*, June 5, 2023, https://www.washingtonpost.com/technology/2023/06/05/chatgpt-hidden-cost-gpu-compute/.

[27] Reiner Pope, et. al., "Efficiently Scaling Transformer Inference," arXiv:2211.05102, November 9, 2022 https://arxiv.org/pdf/2211.05102.pdf; Google, "Cloud TPU Pricing," November 7, 2023, https://cloud.google.com/tpu/pricing; Google reports using 64 TPUv4 chips to generate 64 tokens in 1.9 seconds using the 540 billion parameter PaLM model. At the 3-year reserve pricing for TPUv4 of $1.449/hr, that works out to $0.76 per thousand tokens.

[28] Soumith Chintala [soumithchintala], https://twitter.com/soumithchintala/status/1671267150101721090, June 20, 2023.

[29] Will Oremus, "AI chatbots lose money every time you use them. That is a problem." *The Washington Post*, Jun 5, 2023, https://www.washingtonpost.com/technology/2023/06/05/chatgpt-hidden-cost-gpu-compute/.

[30] Andrew Lohn and Micah Musser, "AI and Compue: How Much Longer Can Computing Power Drive Artificial Intelligence Progress," CSET, January 2022, https://cset.georgetown.edu/publication/ai-and-compute/.

[31] Manish Singh, "OpenAI still not training GPT-5, Sam Altman says," TechCrunch, June 7, 2023, https://techcrunch.com/2023/06/07/openai-gpt5-sam-altman/.

[32] Jaime Sevilla, et. al., "Compute Trends Across Three Eras of Machine Learning," arXiv:2202.05924, February 11, 2022, https://arxiv.org/abs/2202.05924.

[33] Taking into account the statistical penalties for adding fitting parameters, there is a one in fifty million chance that a straight line is better than the kinked line according to the Aikike Information Criterion. According to Bayes Information Criterion instead, a value less than 2 is considered insignificant, between 2 and 5 is weak evidence, and between 5 and 10 is strong evidence. The value favoring a kinked line is 32.

[34] Jaime Sevilla, et al. "Parameter, Compute and Data Trends in Machine Learning," EpochAI, retrieved November 7, 2023, https://epochai.org/data/mlinputs.

[35] OpenAI, "AI and Compute," OpenAI Blog, May16, 2018, https://openai.com/research/ai-and-compute.

[36] About 10% of their models have parameter counts but those that do are especially popular, accounting for just over half of the total number of downloads.

[37] Bert-base-uncased has 110 million parameters and 44,626,861 monthly downloads. xlm-Roberta-large has 561 million parameters and 40,424,763 monthly downloads. The data was pulled from the Hugging Face website on July 7, 2023.

[38] Hugging Face, "Model API," retrieved on July 7, 2023, https://huggingface.co/api/models.

[39] Kyle Miller and Andrew Lohn, "Onboard AI: Constraints and Limitations," CSET, August 2023, https://cset.georgetown.edu/publication/onboard-ai-constraints-and-limitations/.

[40] Hugging Face, "Model API," retrieved on July 7, 2023, https://huggingface.co/api/models.

[41] Micah Musser, et al., "The Main Resource is the Human" A Survey of AI Researchers on the Importance of Compute," CSET, Apr 2023, https://cset.georgetown.edu/publication/the-main-resource-is-the-human/.

[42] See Appendix D in Kyle Miller and Andrew Lohn, "Onboard AI: Constraints and Limitations," CSET, August 2023, https://cset.georgetown.edu/publication/onboard-ai-constraints-and-limitations/.

[43] Angela Fan, et al., "Beyond English-Centric Multilingual Machine Translation," Meta AI, October 18, 2020, https://ai.facebook.com/research/publications/Beyond-English-Centric-Multilingual-Machine-Translation/.

[44] Pablo Villalobos, et al., "Will we run out of data? An analysis of the limits of scaling datasets in Machine Learning," arXiv:2211.04325, October 26, 2022, https://arxiv.org/pdf/2211.04325.pdf.

[45] Rohan Taori, et al., "Alpaca: A Strong, Replicable Instruction-Following Model," Stanford University, May 29, 2023, https://crfm.stanford.edu/2023/03/13/alpaca.html.

[46] Arnav Gudibande, et al., "The False Promise of Imitating Proprietary LLMs," arXiv:2305.15717, May 25, 2023, https://arxiv.org/pdf/2305.15717.pdf.

[47] Subhabrata Mukherjee, et al., "Orca: Progressive Learning from Complex Explanation Traces of GPT-4," arXiv:2306.02707, June 5, 2023, https://arxiv.org/pdf/2306.02707.pdf; Rishabh Agarwal, et al., "GKD: Generalized Knowledge Distillation for Auto-regressive Sequence Models," arXiv:2306.13649, June 23, 2023, https://arxiv.org/abs/2306.13649.

[48] Dylan Patel and Afzal Ahmad, "Google 'We Have No Moat, And Neither Does OpenAI,'" SemiAnalysis, May 4, 2023, https://www.semianalysis.com/p/google-we-have-no-moat-and-neither.

[49] Starting at minute 24 of Lex Friedman Podcast, "Mark Zuckerberg: Future of AI at Meta, Facebook, Instagram, and WhatsApp," June 8, 2023, https://www.youtube.com/watch?v=Ff4fRgnuFgQ.

[50] Jordan Hoffman, et al., "Training Compute-Optimal Large Language Models," arXiv:2203.15556, March 29, 2022, https://arxiv.org/pdf/2203.15556.pdf.

[51] Nvidia, "GPU Performance Background User's Guide," February 1, 2023, https://docs.nvidia.com/deeplearning/performance/dl-performance-gpu-background/index.html#gpu-arch.

[52] Kipply Chen, "Transformer Inference Arithmetic," Kipply's Blog, March 30, 2022, https://kipp.ly/blog/transformer-inference-arithmetic/.

[53] Hugo Touvron, et al., "LLaMa 2: Open Foundation and Fine-Tuned Chat Models," arXiv:2307.09288, July 19, 2023, https://arxiv.org/pdf/2307.09288.pdf.

[54] Hugo Touvron, et al., "LlaMa 2: Open Foundation and Fine-Tuned Chat Models," arXiv:2307.09288, July 19, 2023, https://arxiv.org/pdf/2307.09288.pdf.