Scaling transformative autoregressive models

Matthew Barnett[*], Tamay Besiroglu[*†]
[*]Epoch, [†]MIT FutureTech

February 2023

Abstract

We propose a simple and empirically grounded method for using neural scaling laws to bound the training requirements for machine learning models to reach human-level performance on generic tasks. We show that neural scaling laws can potentially be used to infer the training requirements for a model to copy human behavior on the task to a degree that would be broadly indistinguishable from real human behavior over a long time horizon, according to a trained expert. Combining this method with empirically derived scaling laws for economically valuable tasks may provide a means of upper bounding the compute requirements for training a model to automate those tasks. Our approach offers a simple interpretation of cross-entropy loss of autoregressive models that sheds light on both medium-term capabilities and provides a speculative yet educated guess for the computational resources needed for training highly advanced, potentially transformative, AI systems.

1 Introduction

Over the years, various methods of forecasting advanced AI have been pursued. For example, surveys of experts have been conducted (see Grace et al. 2018 and Grace 2022), and some have developed estimates based on extrapolating fractional progress towards human-level AI (see Bergal 2020).

One influential method for forecasting advanced AI involves anchoring the difficulty of AI development to biological milestones, under the assumption that AI will naturally improve with greater scale. This approach has roots in the work of Good 1966 and has been notably explored by Moravec 1998 and more recently by Cotra 2020. Despite the plausibility of this approach afforded by the fact that biological processes produced the human brain, this model has key weaknesses. Specifically, perhaps the state-of-the-art such model from Cotra 2020 is highly complex, with numerous free variables, resulting in a broad range of uncertainty regarding training requirements. By combining estimates from six biological anchors, the model generates a distribution of compute requirements spanning over 20 orders of magnitude.

Models anchored to biological milestones typically struggle with testability, as current models of this type tend to produce forecasts solely for human-level AI without addressing intermediate milestones. Consequently, current "biological-anchor" models are generally not suitable for back-testing or cross-validation, which limits the extent to which one can improve their methodology through empirical testing and verify the accuracy of the predictions.

1.1 Our framework at a high-level

Roughly speaking, to the extent that an autoregressive model, such as a large language model, can generate novel long outputs that approximate some target distribution extremely well, such a model will have acquired some mastery of the relevant underlying tasks. For example, we can be confident that language models that can generate novel and lengthy code bases that are similar in all relevant respects to those found on GitHub—including whether the code compiles successfully into a working software project—will have "learnt", at least in some functional sense, to perform basic programming tasks.

More speculatively, if such a model learns to generate novel long output sequences that cannot reliably be distinguished from the distribution by skilled observers, then we might be confident that such a model

We thank Tom Davidson, Nuño Sempere, Ege Erdil, Jaime Sevilla, Anson Ho, Pablo Villalobos for their helpful comments.

has, in some sense, achieved a performance at least as great as the humans that generated the relevant data. This assumption—that indistinguishability implies competence—underlies other behaviour tests for machine intelligence (notably Turing 1950's Imitation Game) and enables us to infer the performance requirements for human-level ability on arbitrary autoregressive tasks. Using basic statistical theory, we can infer the cross-entropy loss required for models to display sufficient accuracy and long-term coherence to achieve human-level performance.

Using empirically estimated scaling laws, we can apply these basic results to predict an upper bound on what training inputs will yield autoregressive models that achieve coherency over long sequences, assuming the empirically estimated scaling laws hold over many more orders of magnitude of compute. We also provide a speculative analysis forecasting the training inputs required to train so-called transformative AI (TAI) (Karnofsky 2016), which in this analysis we assume is enabled by training an autoregressive model that achieves coherence over sufficiently long sequences of natural text. To supplement this analysis, we also describe tests that may greatly reduce uncertainty in our estimate over training compute for TAI.

Although our analysis is speculative and currently relies on a low-resolution performance model, we believe it may still capture essential details of the process of training increasingly larger models, provide deeper insights into medium-term possibilities, and serve as an educated guess for the computational resources required for training TAI.

1.2 Scaling laws for autoregressive models

Recent scaling law papers, such as Hoffmann et al. 2022, have proposed a parametric form for cross-entropy loss directly as a power law in terms of data (D) and parameter count (N),

$$L(N, D) = E + \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}}, \tag{1}$$

where E denotes the irreducible loss, and A, B, α, β are constants. Their work, as well as related work on neural scaling laws, has shown that such relationships are fairly robust in that they persist across many orders of magnitude. Power-law scaling laws also seem ubiquitous: Henighan et al. 2020 demonstrated that such scaling laws for auto-regressive modeling are a good fit across a variety of modalities. In addition, power law scaling has been found in reinforcement learning (Neumann and Gros 2022), and has been investigated theoretically (for example, by Bahri et al. 2021).

Given a generic task, we could theoretically fit and apply a power-law scaling law to calculate what inputs are required (with current algorithms) in order to obtain arbitrary levels of performance on that task. Yet this function would only allow us to predict the cross-entropy loss of the model on the task. To solve for a desired *subjective* level of performance, we need some way of interpreting the cross-entropy loss of a model on data in a more common-sense fashion. This work aims to provide a partial answer for how we can interpret the cross-entropy loss of a model.

2 Estimating the difficulty of a task

In theory, if an ML model was trained to replicate human behavior on a task, and achieved perfect accuracy in the sense that it was literally impossible to detect any differences between the behavior of the model and the behavior of a human on that task, then that would almost certainly be sufficient to automate the task. More generally, a fundamental assumption of the approach taken in this report is that indistinguishability implies competence; that is, if a model is indistinguishable from the "real thing" then it must also be at least as competent as the real thing. The intuition for the assumption that indistinguishability requires performance capacity is one that underlies other behavioral tests for machine intelligence, notably Turing's Imitation Game (see e.g., Harnad 1992 for a discussion of this assumption).

Of course, we cannot expect to perfectly replicate human behavior on any real task. Equation (1) illustrates that as the data and parameter count increase, the cross-entropy loss of a model can only approach to E, the irreducible error, over time. This means that, if the scaling law were accurate, it would take infinite computation to get an ML model to replicate human performance. It is also worth stressing that while indistinguishability implies competence, the reverse is not true in general; e.g., a

model can be very competent at playing Go, and exhibit behavior quite distinguishable from a human.

Nonetheless, the main insight here is that, while future transformative models may not be trained to copy human behavior perfectly, we can still plausibly use the *difficulty of replicating human behavior* as an upper bound for the *difficulty of reaching human performance* on a given task. Moreover, there is a neat theoretical connection between a model's cross-entropy loss and how difficult it is to distinguish the model from human behavior. The next section will elaborate on this theoretical connection.

2.1 Distinguishability and k-performance

Consider the information-theoretic decomposition of cross-entropy as a sum of entropy and the KL-divergence,

$$H(p, q) = H(p) + D_{KL}(p\|q). \tag{2}$$

The KL-divergence, or $D_{KL}(p\|q)$ can be seen as a measure of how distinguishable distribution q is from p when using q as a model. Formally, it is defined as the expected surprise of a sample judged by q when sampling from p,

$$D_{KL}(p\|q) = \sum_{x \in X} p(x) \log \left(\frac{p(x)}{q(x)} \right). \tag{3}$$

Let us informally define the k-performance of a model on some task, relative to a judge (such as a trained human expert), as the average number of samples from the model required in order for the judge to reliably distinguish its behavior from a human expert performing the same task. To make the notion of k-performance stronger, we imagine that the judge can be arbitrarily strong; i.e., if specified, they may have various tools or unlimited time available to help them discriminate between models. While this definition is imprecise, it provides a starting point for understanding how we could predict when a model will perform "well enough" to replace human labor.

Consider, for example, a model whose k-performance on playing chess is 250, relative to a chess grandmaster. If a sample is taken to be a single move in the game, then by definition, on average, it takes 250 moves for a grandmaster to reliably distinguish between the model's play and play from that of a generic human grandmaster. Since the vast majority of chess games finish before 250 moves (Chessgames 2023), this model is likely highly difficult to distinguish from grandmaster play on any given game; thus, we should expect its competence at chess to be similar to or possibly greater than grandmaster play. Accordingly, if we can measure the difficulty of training a model to reach a k-performance of 250 at chess, we will obtain an upper bound for the difficulty of creating a grandmaster-level chess engine.

2.2 Ideal distinguishability

In order to make the concept of k-performance more precise, we can first consider how many samples are required, on average, for an ideal predictor to discriminate between models, ie., one that has access to the full distribution p and q, and can therefore calculate the Bayes factor in favor of p or q for each sample, up to some desired degree of confidence. This problem has been studied in the context of sequential probability ratio tests (SPRTs), going back to at least Wald 1945.[1]

Consider two hypotheses while sampling, corresponding to whether we are sampling from p_0 or p_1,

$$H_i : X_1, X_2, \ldots, X_n \overset{iid}{\sim} p_i, i = 0, 1. \tag{4}$$

If we start with a uniform prior over hypotheses (e.g., 50% for either), the question becomes how large n needs to be, on average, before we have reached a desired level of confidence (e.g., 99%) about whether we are sampling from p_0 or p_1. To answer the question, we first realize the likelihood ratio,

$$\Lambda_k = \prod_{i=1}^{k} \frac{p_1(X_i)}{p_0(X_i)}, k = 1, 2, \ldots \tag{5}$$

[1] In our exposition, we will follow these lectures notes of Robert D Nowak and derive an approximation for k-performance. Our derivation will vary only slightly from the derivation in the notes, as we will take into account prior probability in the analysis.

By Bayes' rule, we can find the posterior odds over our hypotheses by multiplying the likelihood ratio by our uniform prior odds. This is most easily seen in the case of the log-odds form of Bayes' rule,

$$\log\left(\frac{P(H_0|X_1,X_2,\ldots,X_k}{P(H_1|X_1,X_2,\ldots,X_k}\right) = \log\left(\frac{P(H_0)}{P(H_1)}\right) + \sum_{i=1}^{k}\log\left(\frac{p_0(X_i)}{p_1(X_i)}\right) \tag{6}$$

$$= \log\left(\frac{P(H_0|X_1,X_2,\ldots,X_k}{P(H_1|X_1,X_2,\ldots,X_k}\right) = \log\left(\frac{P(H_0)}{P(H_1)}\right) + \log\Lambda_k \tag{7}$$

$$= \log O(H_0) + \log\Lambda_k. \tag{8}$$

Thus, our procedure is as follows: we continue to sample from the distribution until our posterior odds for H_0 exceeds some specified threshold $\log(\gamma/(1-\gamma))$ or falls below $\log((1-\gamma)/\gamma)$. Here, γ refers to the confidence level at which our procedure will terminate.

For example, if our desired level of confidence is 99%, then $\gamma = 0.99$, and we will continue sampling (increment k) until either

$$\log O(H_0) + \log\Lambda_k \geq \log\left(\frac{0.99}{0.01}\right) = \log(99), \tag{9}$$

or

$$\log O(H_0) + \log\Lambda_k \leq \log\left(\frac{0.01}{0.99}\right) = -\log(99). \tag{10}$$

Let K^* indicate the stopping time for our procedure. The quantity we seek is $E_j[K^*]$, or the expected stopping time until we have reached a desired level of confidence, when the true distribution is j.

Notice that for a fixed k,

$$E_j[\log\Lambda_k] = E_j\left[\sum_{i=1}^{k}\log\frac{p_1(X_i)}{p_0(X_i)}\right] = \sum_{i=1}^{k}E_j\left[\log\frac{p_1(X_i)}{p_0(X_i)}\right] = \begin{cases} kD(p_1\|p_0), j=1 \\ -kD(p_0\|p_1), j=0 \end{cases} \tag{11}$$

What if we wanted to know $E_j[\log\Lambda_{K^*}]$? In this case, the derivation is slightly more complex, since K^* is a random variable and is also a function of $X_1, X_2, \ldots$. But by applying Wald's identity, we can arrive at the following expression for $E_j[\log\Lambda_K*]$, where j refers to the true distribution we are sampling from,

$$E_j[\log\Lambda_{K^*}] = \begin{cases} E_j[K^*]D(p_1\|p_0), j=1 \\ -E_j[K^*]D(p_0\|p_1), j=0 \end{cases} \tag{12}$$

To obtain $E_j[K^*]$ we will assume that when our procedure terminates, our posterior will be approximately equal to the threshold level of confidence. Under this assumption, it can be shown that we can express $E_j[K*]$ in a way that permits direct calculation:

$$E_0[K*] \approx \frac{P(D)_1\log(\frac{P(D)_0}{P(D)_1}) + (1-P(D)_1)\log(\frac{1-P(D)_0}{1-P(D)_1})}{-D(p_0\|p_1)}, \tag{13}$$

$$E_1[K*] \approx \frac{P(D)_0\log(\frac{P(D)_0}{P(D)_1}) + (1-P(D)_0)\log(\frac{1-P(D)_0}{1-P(D)_1})}{-D(p_1\|p_0)}. \tag{14}$$

This expression therefore enables us to express $E_j[K^*]$ as a function of $P(D)_j$ and the KL-divergence between p_0 and p_1, the derivation for which is provided in Appendix A.

2.3 Practical distinguishability

In the previous section, we proved the k-performance of a model relative to an ideal predictor. In practice, however, even the strongest experts will fall short of this ideal. One model for human discrimination abilities is to assume that humans update more slowly than the ideal predictor by some constant factor, which we can call the "slowdown" factor. Symbolically, we can write that for a human judge, their "modified log-odds Bayes' rule" would take the following form,

$$\log(\text{Posterior odds}) = \log(\text{Prior odds}) + \frac{\log(\text{Bayes factor})}{\text{Slowdown}}. \tag{15}$$

in which the term for the slowdown is always greater than or equal to 1, and controls the slowdown in human updating relative to an ideal discriminator.

Another model for human discrimination is to assume that humans only attend to some fraction of information that may aid in discriminating two distributions. For example, if given two images, one generated by a human artist, and one generated by a text-to-image model, a human may look for artifacts in the images in order to distinguish which image is which, as those might give away which image was produced by an imperfect model.

However, artifacts are not the only relevant pieces of information encoded in the images. In theory, even a single pixel difference between the images could provide strong evidence that one image was produced by a model. While this information could be employed by an ideal discriminator, humans are not fully aware of all relevant information. Therefore, we will posit the existence of some constant "attending fraction" of the data that humans will pick up on in order to discriminate between distributions.

Conveniently, both the "slowdown" model and the "attending fraction" model of human discrimination abilities yield identical predictions in expectation, with Slowdown = 1/(Attending fraction). Consider a model of human updating that takes on the form,

$$\log(\text{Posterior odds}) = \log(\text{Prior odds}) + \log(\text{Bayes factor})\mathbb{1}(A_i). \tag{16}$$

where $\mathbb{1}(A_i)$ is an indicator function on a random variable A_i, which is sampled from the uniform distribution on $[0, 1]$ for each piece of evidence, and equals 1 when $A_i >$ attending fraction, and 0 otherwise:

$$A_i \overset{iid}{\sim} U, \quad \mathbb{1}(A_i) = \begin{cases} 1 & \text{if } A_i < \text{ Attending fraction} \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

The expected update, $E[\log(\text{Bayes factor})\mathbb{1}(A_i)] = (\text{Attending fraction} * \text{Bayes factor} + (1 - \text{Attending fraction} * 0) = \text{Attending fraction} * \text{Bayes factor}$. For simplicity, we will assume in this document the "slowdown" model of human discrimination, rather than the "attending" model, safe with the knowledge that both models are fundamentally very similar.

Since the slowdown is a constant multiplied by the log-likelihood during update, k-performance is scaled by the inverse of the slowdown factor (see equation 11 in the previous section).

We can estimate the slowdown factor using experiments in which we get humans to try to discriminate between a model and real human performance, and calculate how many times more samples it takes a human on average compared to the ideal predictor. See Appendix B for an example experiment in which this factor could be estimated. Since the exact slowdown factor is an important source of uncertainty in this analysis which could be reduced, it is relatively promising to pursue these experiments.

2.4 Distinguishability of language modeling

In the previous sections, we assumed that we were drawing i.i.d. samples from some distribution. For autoregressive tasks, samples are drawn from a conditional distribution; for language modeling, that means each token is sampled from a distribution conditioned on all the text that came before.

Given the lack of independence, the likelihood ratio cannot be assumed to be a simple product as shown in equation (5). Instead, the likelihood ratio is given by,

$$\Lambda_k = \frac{p_1(X_{1:k})}{p_0(X_{1:k})}. \tag{18}$$

As is commonly done, we will model language as a stationary ergodic process for simplicity. While natural languages are not actually stationary and ergodic, as the probability of a word may sometimes depend on words arbitrarily far back in the text, this assumption may still provide a useful approximation (Jurafsky and Martin 2023). The generalized Shannon–McMillan–Breiman theorem (Algoet and Cover 1988) states that for any stationary ergodic process $X_{1:n} \sim p$,

$$H(p, m) = \lim_{n \to \infty} -\frac{1}{n} \log m(X_{1:n}). \tag{19}$$

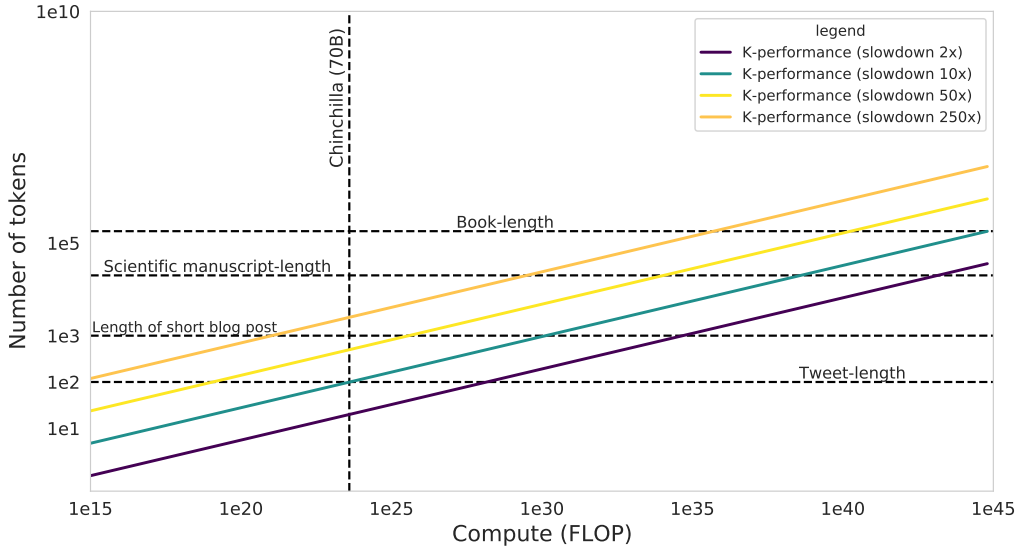


Figure 1. Relationship between k-performance and compute (FLOP). The predicted k-performance of a language model with respect to a hypothetical expert discriminator, as a function of compute, at 90% confidence. This illustrates the predicted amount of computation required to train so that a hypothetical expert discriminator needs at least that many tokens to be able to confidently discriminate its outputs from the relevant distribution at a 90% confidence level.

For long sequences, this equation provides an approximate value for $\log \Lambda_k$ as follows,

$$E_j[\log \Lambda_k] \approx -k(H(p_j, p_1) - H(p_j, p_0)) \tag{20}$$

$$= \begin{cases} kD(p_1 \| p_0), j = 1 \\ -kD(p_0 \| p_1), j = 0 \end{cases} \tag{21}$$

Since this expression matches (11), we can reuse the approximation for $E_j[K*]$ given in (24) and (25) for the case of language models.

Assuming the parametric scaling law in Hoffmann et al. 2022, we can determine the direct relationship between training compute and k-performance. Consider the scaling law of cross-entropy loss again in parameters (N) and data (D),

$$L(N, D) = E + \frac{A}{N^\alpha} + \frac{B}{D^\beta}. \tag{22}$$

The unique N_{opt} and D_{opt} that minimizes loss for a given compute budget (C), in which $\text{FLOPs}(N, D) = 6ND$,

$$\underset{N,D \text{ s.t. } \text{FLOPs}(N,D) \leq C}{\arg\min} E + \frac{A}{N^\alpha} + \frac{B}{D^\beta}, \tag{23}$$

is given by (see section 3.3 in Hoffmann et al. 2022),

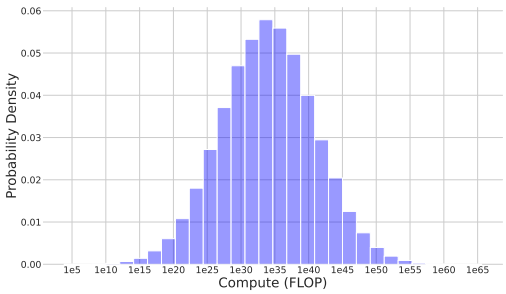
$$N_{\text{opt}}(C) = G \left(\frac{C}{6}\right)^a, D_{\text{opt}}(C) = G^{-1} \left(\frac{C}{6}\right)^b \tag{24}$$

$$\text{where } G = \left(\frac{\alpha A}{\beta B}\right), a = \frac{\beta}{\alpha + \beta}, \text{and } b = \frac{\alpha}{\alpha + \beta}. \tag{25}$$

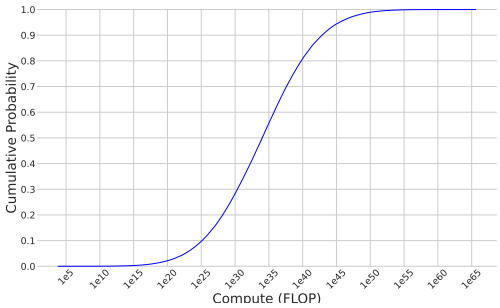
Putting these equations together with the approximation for k-performance given by (20), we can calculate k-performance as a function of training compute. Plugging in the empirical parameters revealed by Hoffmann et al. 2022, we arrive at figure 1, revealing a power law relationship (linear in log-log space) between compute and k-performance. It is worth noting that at very low levels of compute, the ideal predictor can easily distinguish two distributions using very few datapoints. Humans will fall far short of this ideal.

2.5 Speculative estimate for the training compute requirements of TAI

In this section, we provide a highly speculative estimate for the compute requirements of training transformative AI (TAI) (Karnofsky 2016).[2] The notebook is composed of two parts. The first part attempts to calculate a distribution over the maximum amount of compute required to train something like TAI, given estimates over the parameters of the model: namely, the human slowdown factor, and the k-performance required for transformative capabilities on language modeling. The second section uses this probability distribution to forecast the date that TAI is developed given estimates about algorithmic progress, hardware progress, and growth in willingness and ability to spend as a fraction of Gross World Product (GWP), among other uncertainties.



(a) Distribution over upper bounds of FLOP required for transformative levels.



(b) CDF over FLOP required for transformative levels of performance.

Figure 2. Distribution over the upper bounds of FLOP required for transformative levels of performance. For the slowdown factor parameter, we estimated a lognormal distribution with the 15th percentile at 9.85 and the 85th percentile at 289.05. For the requisite k-performance parameter, we very roughly estimated a lognormal distribution with the 15th percentile at 3129 tokens and the 85th percentile at 141,714 tokens.[3] The distribution over maximum compute required for TAI given manual parameter settings. Note that this distribution is technically a (soft) upper bound. It must be understood that these are highly preliminary results, given primarily for illustration of how this framework may produce relevant predictions.

The distribution above assigns non-negligible probability to compute levels that we have already observed in historical training runs (i.e. below $\sim$ 1e25 FLOP, see Sevilla et al. 2022). Because of this, it is necessary to update on the fact that we have yet to observe TAI. This update is explained in Appendix C.

2.6 Key sources of uncertainty

So far this analysis has assumed that the functional form, and the exact empirically-derived parameters of the Hoffmann et al. 2022 scaling law are correct. However, it is appropriate to have uncertainty about whether the functional form is correct, and about the parameters of the law itself. In addition, the empirically derived Hoffmann et al. 2022 scaling law only applies to the model and dataset used in that paper. In general, our understanding of scaling laws is somewhat limited, prohibiting a more confident analysis.

Yet an even larger uncertainty may be the uncertainty over the parameter k, standing for the requisite k-performance required to emulate a given task to a sufficient standard. Each task requires estimating a different k, and for many tasks, we may have only intuition to rely upon for reducing our uncertainty about this parameter. For some tasks, such as writing scientific papers, we can conceivably use heuristics, such as the length of an average scientific paper. However, making the right guess about k remains extremely speculative. The next section elaborates on the potential for making progress estimating the right k for various tasks.

[2] All relevant calculations can be provided in this Google Colab notebook.

[3] To obtain an estimate over the parameters, we elicited estimates from Epoch team members and averaged over these estimates using either an arithmetic mean or geometric mean, depending on whichever we thought was more appropriate.

2.7 How to improve these estimates

One advantage of using this approach to forecast transformative AI (Karnofsky 2016) relative to other approaches such as Bio Anchors, is that this uncertainty can be reduced substantially via experiment, ie., by several orders of magnitude. Appendix B describes one experiment that can reduce uncertainty over the slowdown factor. However, uncertainty must also be reduced over k, or the requisite k-performance parameter.

To get a better intuition about the right k, we can perform experiments on simple tasks, and calculate the expected upper bound compute distribution. For example, one could use this method to calculate the expected amount of compute required to reach human-performance on the popular games of Tic-tac-toe, Chess, and Go, and compare these estimates to the actual amounts of compute that were historically required to master performance on the task.

A key advantage of this method of forecasting AI is that it is task-agnostic, rather than task-specific. In other words, the method can be used, in theory, to forecast performance in any autoregressive task. It is our hope that with experience, and the right data, it should be possible to considerably reduce the uncertainty over the arrival of TAI, by getting a better sense as to what value of k-performance is appropriate on language modeling in order for transformative capabilities to emerge. And of course, ordinary forecasting work can help to reduce our uncertainty in, e.g., expected price-performance declines in computing, or expected algorithmic progress.

3 Conclusion

We have presented an approach to estimate an upper bound for the training compute necessary to achieve a specified level of performance in AI systems. Our method is grounded in the assumption that indistinguishability implies competence, a foundational concept in behavior tests for machine intelligence, such as Turing's Imitation Game. By employing basic statistical theory, we demonstrate that typical scaling laws can be interpreted as providing a characterization of how accuracy and long-term coherence improve with scale. Using this insight, we provide a plausible bound on the requirements for training autoregressive models to achieve human-level performance on generic tasks.

While the method has not yet been subjected to empirical tests, its results should be interpreted cautiously. Nevertheless, the fact that our approach is empirically testable represents a significant step towards creating falsifiable models of AI progress. Future research should aim to make this method more empirical, for instance, by investigating the relationship between cross-entropy loss and benchmark results. Overall, our approach offers a promising and empirically testable method for estimating the compute requirements of advanced AI systems to achieve human-level performance, representing a step towards creating falsifiable models of AI progress.

Appendix

Appendix A: Derivation of our expression for $E_j[K^*]$

To obtain $E_j[K^*]$ we will assume that when our procedure terminates, our posterior will be approximately equal to the threshold level of confidence. Under each hypothesis, there is some probability that the posterior will terminate on the correct threshold (detection, or $P(D)$) and some probability that it will terminate on the wrong threshold (false alarm, or $1 - P(D)$). We can therefore write,

$$E_j[\log \Lambda_{K^*}] \approx P(D)_j \log(\gamma) + (1 - P(D)_j)(-\log(\gamma)) \tag{26}$$

We can express $P(D)_j$ and $P(FA)_j$ as follows. To simplify notation, let $x = (X_1, X_2, \ldots, X_k)$ and let $p_j(x) = \prod_{i=1}^{k} p_j(x_i), j = 0, 1$. We can write $P(D)_0$ in terms of the decision set

$$R_j = \begin{cases} x : O(H_0) + \log(\Lambda_k) \geq \log(\frac{\gamma}{1-\gamma}), j = 1 \\ x : O(H_0) + \log(\Lambda_k) \leq \log(\frac{1-\gamma}{\gamma}), j = 0 \end{cases} \tag{27}$$

Let δ_j stands for the stopping threshold for R_j. Then $\delta_1 = O(H_1)\frac{\gamma}{(1-\gamma)}$ and $\delta_0 = \Lambda_k \leq O(H_1)\frac{(1-\gamma)}{\gamma}$,

$$R_j = \begin{cases} x : \Lambda_k \geq \delta_1, j = 1 \\ x : \Lambda_k \leq \delta_0, j = 0 \end{cases} \tag{28}$$

Thus, we have,

$$P(D)_0 = \int_{R_0} p_0(x)\,dx = \int_{R_0} \frac{p_0(x)}{p_1(x)} p_1(x)\,dx \tag{29}$$

$$= \int_{R_0} \Lambda_k p_1(x)\,dx = \delta_0 \int_{R_0} p_1(x)\,dx \tag{30}$$

$$= \delta_0 \int_{R_0} p_1(x)\,dx = \delta_0 P(D)_1 \tag{31}$$

Similarly,

$$P(D)_1 = \int_{R_1} p_0(x)\,dx = 1 - \int_{R_0} p_0(x)\,dx = 1 - \int_{R_0} \frac{p_0(x)}{p_1(x)} p_1(x)\,dx \tag{32}$$

$$= 1 - \int_{R_0} \Lambda_k^{-1} p_1(x)\,dx = 1 - \delta_1^{-1} \int_{R_0} p_1(x)\,dx \tag{33}$$

$$= 1 - \delta_0^{-1}(1 - P(D)_0) \tag{34}$$

From these relationships, we can derive the following formulas for $P(D)_0$ and $P(D)_1$,

$$P(D)_0 = \delta_1 \frac{1 - \delta_0}{\delta_1 - \delta_0} \tag{35}$$

$$P(D)_1 = \frac{1 - \delta_0}{\delta_1 - \delta_0} \tag{36}$$

Now, with these approximations, we're finally ready to express $E_j[K*]$ in a way that permits direct calculation,

$$E_0[K*] \approx \frac{P(D)_1 \log(\frac{P(D)_0}{P(D)_1}) + (1 - P(D)_1) \log(\frac{1-P(D)_0}{1-P(D)_1})}{-D(p_0 \| p_1)}, \tag{37}$$

$$E_1[K*] \approx \frac{P(D)_0 \log(\frac{P(D)_0}{P(D)_1}) + (1 - P(D)_0) \log(\frac{1-P(D)_0}{1-P(D)_1})}{-D(p_1 \| p_0)}, \tag{38}$$

thereby yielding the desired expressions. □

Appendix B: Measuring human slowdown and tests for model assumptions

In this section, I'll outline an experiment that could help estimate the human slowdown parameter described in section 2.3. At the same time, this experiment can also test the assumption that the rate of slowdown between humans and ideal discriminators is constant across different distributions.

The idea is to measure difference in performance between a trained human expert and the estimated performance of an ideal discriminator at the task of discriminating between real text and model-written text. Since we can calculate the performance of an ideal discriminator given only the KL-divergences, we only need the KL-divergences between the distributions, and human performance, which can be operationalized as how quickly humans are able to distinguish between samples from the distributions reliably.

Let us imagine a trial in which volunteers are given text sampled from either a natural distribution or a model distribution, that they must label either "artificial" or "human-written." The text is revealed incrementally (e.g., one word at a time), the rate of which is controlled by the user, for example, by pressing a key each time the volunteer wants to reveal a word. Each volunteer is instructed to signal when they have reached 90% confidence about the source of the text. They are given rewards both for quick discrimination, and for being well-calibrated, so their predictions should be close to 90% accurate. Additionally, the volunteers are expected to get better at this task over time.

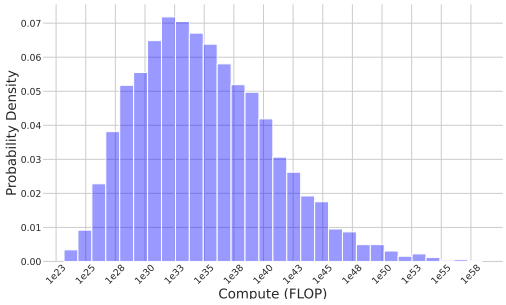
We can compare the behavior of an ideal discriminator (a quantity we can calculate) to the average

behavior of human volunteers, who have lots of experience on the task. We predict that there will be a constant slowdown factor (see section 2.3) that describes the difference between human and ideal performance. In addition to testing the assumption that there will be a constant slowdown factor between ideal and human performance, this experiment will also allow us to reduce our uncertainty over the slowdown parameter, and thus, reduce our uncertainty over the date when transformative AI may arrive.

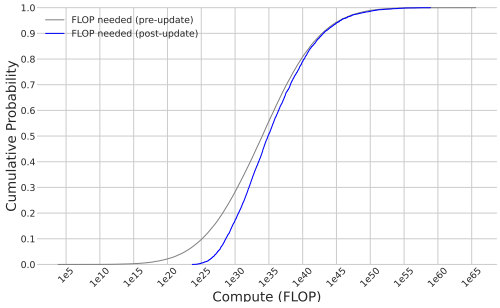
Appendix C: Updating the compute distribution to take into account our observations

The model outlined in section 3.5 yields a distribution over the maximum amount of compute required to train a transformative model. Yet, naively, the distribution compute contains a non-negligible probability on values that we have already observed from real-world training runs. This is problematic because it means that we are assigning some probability to transformative AI having already arrived.

One might think that we can simply perform a Bayesian update in order to make the distribution reflect these changes, but it is not obvious how one should do so. Naively, we could cut off all probability for values below the largest training run we have so far observed (which at the time of writing is about 1e25 FLOP) and re-normalize the distribution. However, this approach yields an unnatural distribution in which the probability density discontinuously increases from zero to some non-zero value. In practice, most people, upon observing that 1e25 FLOP was insufficient for anything near TAI, would want to update against values not very far above 1e25 FLOP too. However, it is unclear if we can do this in a principled way.



(a) Distribution over upper bounds of FLOP required for transformative levels of performance after updating on current AI not being transformative.



(b) CDF over upper bounds of FLOP required after updating on current AI not being transformative (blue) contrasted against our earlier CDF (gray).

Figure 2. Posterior distribution over upper bounds of FLOP required for transformative levels of performance after updating on current AI not being transformative.

In the absence of a principled method of doing the update, we have used the following model of the FLOP requirements. Assume that TAI arrival follows a piece-wise Poisson process between successive training runs in which the rate of arrival is determined by a power law in FLOP, with a constant and exponent parameter. Then, given a prior over these parameters, and given historical data on FLOP training runs, the time between them, and whether TAI arrived, we can update the parameters through Bayes' rule, which we can use to provide a distribution over the maximum FLOP requirement. The prior over the constant in the power law is determined by the prior over FLOP values for TAI. The power law is a free parameter which determines the smoothing for the update. The posterior distribution is obtained by the probability of TAI arriving within one year, according to the updated Poisson process.

References

Algoet, Paul H and Thomas M Cover (1988). "Asymptotic optimality and asymptotic equipartition properties of log-optimum investment". In: *The Annals of Probability*, pp. 876–898.

Bahri, Yasaman et al. (2021). "Explaining neural scaling laws". In: *arXiv preprint arXiv:2102.06701*.

Bergal, Asya (Sept. 2020). *Surveys on fractional progress towards HLAI*. URL: https://aiimpacts.org/surveys-on-fractional-progress-towards-hlai/.

Chessgames (2023). *Chessgames*. Accessed: 2023-04-25. URL: https://www.chessgames.com/chessstats.html.

Cotra, Ajeya (2020). *Draft report on AI timelines*. URL: https://www.alignmentforum.org/posts/KrJfoZzpSDpnrv9va/draft-report-on-ai-timelines.

Good, Irving John (1966). "Speculations concerning the first ultraintelligent machine". In: *Advances in computers*. Vol. 6. Elsevier, pp. 31–88.

Grace, Katja (Aug. 2022). *What do ML researchers think about AI in 2022?* URL: https://aiimpacts.org/what-do-ml-researchers-think-about-ai-in-2022/.

Grace, Katja et al. (2018). "When will AI exceed human performance? Evidence from AI experts". In: *Journal of Artificial Intelligence Research* 62, pp. 729–754.

Harnad, Stevan (1992). "The Turing Test is not a trick: Turing indistinguishability is a scientific criterion". In: *ACM SIGART Bulletin* 3.4, pp. 9–10.

Henighan, Tom et al. (2020). "Scaling laws for autoregressive generative modeling". In: *arXiv preprint arXiv:2010.14701*.

Hoffmann, Jordan et al. (2022). "Training Compute-Optimal Large Language Models". In: *arXiv preprint arXiv:2203.15556*.

Jurafsky, Dan and James H. Martin (2023). *Speech and Language Processing (3rd ed. draft)*, pp. 23–24.

Karnofsky, Holden (2016). *Some Background on Our Views Regarding Advanced Artificial Intelligence*. URL: https://www.openphilanthropy.org/research/some-background-on-our-views-regarding-advanced-artificial-intelligence/.

Moravec, Hans (1998). "When will computer hardware match the human brain". In: *Journal of evolution and technology* 1.1, p. 10.

Neumann, Oren and Claudius Gros (2022). "Scaling laws for a multi-agent reinforcement learning model". In: *arXiv preprint arXiv:2210.00849*.

Sevilla, Jaime et al. (2022). *Parameter, Compute and Data Trends in Machine Learning*. https://docs.google.com/spreadsh

Turing, Alan M (1950). *Computing machinery and intelligence*. Springer.

Wald, Abraham (1945). "Sequential tests of statistical hypotheses". In: *Breakthroughs in Statistics*. Springer, pp. 256–298.