Forecasting TAI with biological anchors

Part 3: Hypotheses and 2020 training computation requirements

Author: Ajeya Cotra
Date: July 2020

This report emerged from discussions with our technical advisors Dario Amodei and Paul Christiano. However, it should not be treated as representative of either of their views; the project eventually broadened considerably, and my conclusions are my own.

This is a work in progress and does not represent Open Philanthropy's institutional view. We are making it public to make it easier to gather feedback, to help inform others' thinking in the effective altruism community, and to allow for follow-on work outside of Open Phil. However, we may edit it substantially in the future as we gather feedback from a broader audience and investigate open questions. Accordingly we have not done an official publication or blog post, and would prefer for now that people not share it widely in a low-bandwidth way (e.g., just posting key graphics on Facebook or Twitter).

The report has been divided into four Google docs to load faster. ***This is Part 3; the first part is here, the second part is here, and the fourth part is here.*** *Additional materials (collected in this folder):*

- *Quantitative model: the Python notebook Biological anchor hypotheses for 2020 training computation requirements; a template spreadsheet When required computation may be affordable; and my best guess, conservative, and aggressive forecasts.*
- *Supplemental materials: a document containing various appendices; a folder of figures for the report; the spreadsheet Extrapolations of data and compute to train models; and the Python notebook Compute price trends, which draws on data in this folder.*

In Part 1, I provided an overview of the framework and estimates, provided definitions for key abstractions used in the model, and generated an estimate for the number of FLOP / subj sec of a transformative model. In Part 2, I reviewed theoretical and empirical evidence about training data requirements for a transformative model, introduced the concept of horizon length, and estimated how training data requirements may scale with parameter count for a transformative ML problem.

In this part, I will discuss each of the six biological anchors hypotheses in more detail, and combine them to generate my 2020 training FLOP requirements distribution:

- I will start with the Neural Network hypotheses which I place the most weight on (more).
- I will then cover the Evolution Anchor, Genome Anchor, and Lifetime Anchor hypotheses in less detail (more).
- Finally, I will describe in more detail how I update against low-end FLOP levels and assign probabilities to each hypothesis to generate my 2020 training FLOP requirements distribution (more).

Then in Part 4, I will explain how I generate my estimate for when the amount of computation required to train a transformative model may become available, and answer several questions and objections about the framework.

Neural network hypotheses

This family of hypotheses states that we should assume on priors that a transformative model would perform roughly as many FLOP / subj sec as the human brain and have about as many parameters as we would expect if we simply scaled up the architectures of the largest current neural networks (e.g. transformer architectures) to run on that many FLOP / subj sec.

In Part 1 I generated a probability distribution centered around **~1e16 FLOP / subj sec** for the amount of computation that a transformative model is likely to run on; this is 1 OOM larger than my central estimate for brain FLOP/s. This estimate will be used for the Neural Network hypotheses and the Genome Anchor hypothesis below.

It adjusts from the anchor point of human brain FLOP/s by a relatively modest constant factor to account for qualitative considerations about how sophisticated our architectures seem to be as of 2020, and estimate parameter count by assuming that a transformative model would have a similar ratio of computation to parameters as the most expensive neural networks we have trained so far. With this it extrapolates the amount of FLOP required to train such a model using an empirically-derived scaling law that expresses training data as a function of parameter count.

This results in a median estimate somewhere between ~1e31 FLOP and ~1e38 FLOP depending on the effective horizon length of the learning problem. In this section, I will:

- Generate an estimate for the number of parameters this model may require if its architecture is similar to existing neural networks (more).
- Discuss what a reasonable range of effective horizon lengths could be for a transformative ML problem (more).
- Generate subjective probability distributions for training FLOP requirements conditional on this family of hypotheses, broken into “short”, “medium”, and “long” effective horizon lengths (more).

Parameter count of a transformative neural network

I have focused first on inference FLOP / subj sec because computation feels somewhat more “fundamental” and universal than parameter count (it is more comparable across different model architectures and across the whole spectrum from “mostly classical programming” to “mostly pure ML”).

The more computationally powerful a neural network is, the more parameters it is likely to use, so we can try to derive an estimate of parameter count given our estimate for inference computation.

For example, in a simple feedforward neural network architecture, a model which performs twice as many FLOP per forward pass will straightforwardly have twice as many parameters, because each parameter performs roughly one multiplication and one addition in a feedforward architecture. Convolutional neural networks (CNNs) perform substantially more FLOP per parameter per forward pass than a generic feedforward neural network with the same number of parameters can, but also generally maintain a consistent ratio as they are scaled up. Other common architectures such as recurrent neural networks, transformers, etc also exhibit a broadly similar scaling behavior within their respective architecture class -- there is generally a fixed number of FLOP performed per parameter per forward pass.

For typical neural network architectures, this ratio is somewhere between ~1 and ~100 FLOP per parameter per forward pass.[1] How many forward passes would a transformative model need to perform per second?[2] I would expect that ~0.1-10 forward passes per second is sufficient:

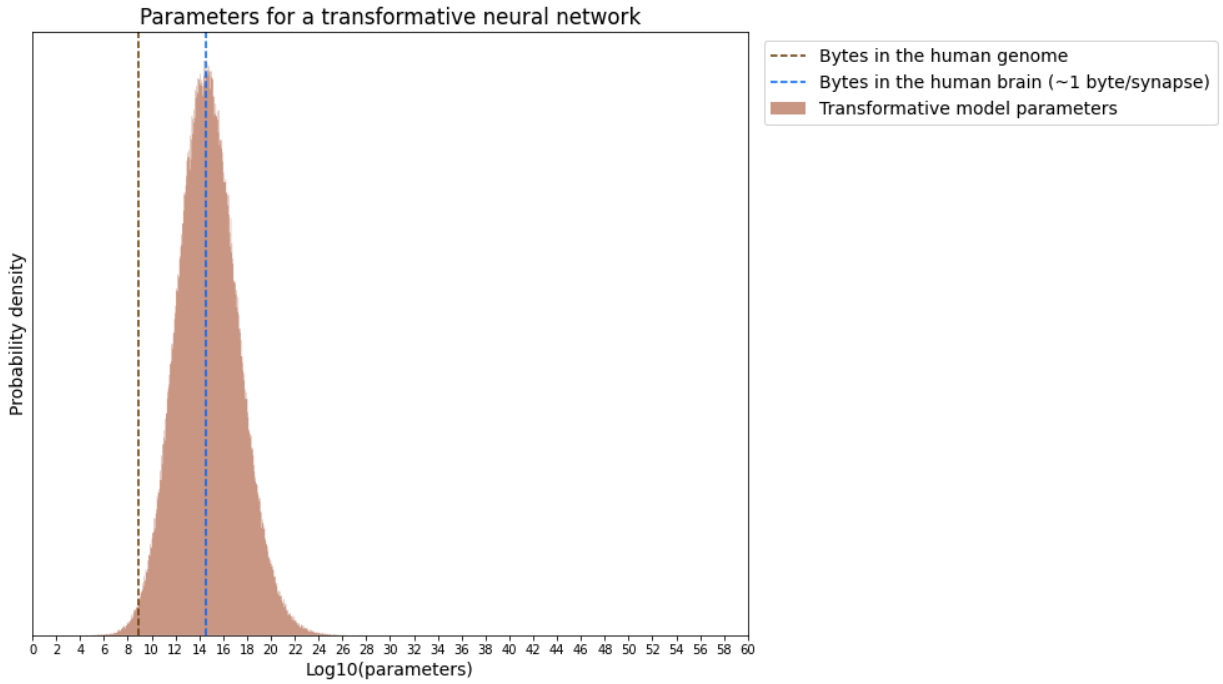
- According to Joe's report, each of the ~1e14-1e15 synapses in the human brain experiences a firing event roughly ~0.1-2 times per second. If we treat the parameters of a transformative model as broadly analogous to the synapses in the brain, that would suggest that each parameter should be used ~0.1-2 times per second.
- Average English speaking speed in conversation works out to about 2.5 words per second, while average reading speed works out to about 4 words per second and a good typing speed is slightly more than one word per second. If we imagine that a transformative model is primarily language-based, processing or generating one token of text with each forward pass, then matching human speaking, reading, and writing speeds would suggest performing a few forward passes per second.
- Average human reaction time is about 250 milliseconds, while especially skilled humans such as fighter pilots or competitive video game players reach reaction times closer to 100 milliseconds; if each forward pass of a transformative model constitutes an "action" and the model would need to match human reaction times in order to have a transformative impact, that would imply that it would need to perform 4-10 forward passes per second.

These assumptions imply that a transformative neural network should perform (1 to 100) * (0.1 to 10) = 0.1 to 1,000 FLOP per parameter per second, meaning that a model running on ~1e16

[1] Models which reach ratios higher than this are usually pure image processing models, which I think are likely less representative of a transformative model than language models or RL models.

[2] Note that researchers are often able to control how many forward passes they run per second -- "forward passes per second" is not an inherent feature of a neural network, unlike "parameter count" or "FLOP per forward pass." For example, I expect that it would be necessary to perform thousands of forward passes per second to make the Long Horizon Neural Network hypothesis work, because otherwise training would take too much wall-clock time. In this section, though, I am assuming that the model is running at "human speed."

FLOP / subj sec would require ~1e13 to ~1e17 parameters. I expect that researchers would attempt to increase this ratio if possible, since reducing parameter count reduces data requirements, so I expect higher values to be somewhat more likely than lower values. For simplicity, I have assumed that the parameter count distribution for a transformative neural network is identical to the FLOP / subj sec distribution, but shifted to the left by 1.5 OOM (i.e. a factor of ~30), resulting in a distribution **centered around ~3e14 parameters**, a value slightly smaller than the number of synapses in the human brain:



Note that this ratio is simply based on a rough empirical relationship between model FLOP per forward pass and model parameters. it is certainly possible in theory to design a very computationally intensive architecture with a very small number of parameters. For example, according to Wikipedia, Deep Blue (generally considered a "classical computer program" rather than a "machine learning model") employed some number of learned parameters representing the relative value of particular positions in its evaluation function.[3] Conversely, it is also possible to design models which only use a small fraction of their parameters in each forward pass, such as the mixture-of-experts model GShard.

The chart above marks a point for the estimated number of bytes in the human genome; the Genome Anchor hypothesis (discussed below) assumes that the number of parameters

[3] "Deep Blue's evaluation function was initially written in a generalized form, with many to-be-determined parameters (e.g. how important is a safe king position compared to a space advantage in the center, etc.). The optimal values for these parameters were then determined by the system itself, by analyzing thousands of master games."

required to characterize a transformative model is in this range, and therefore that the ratio of FLOP per parameter per forward pass is much more extreme.

Effective horizon length of a transformative ML problem

Recall that I am assuming we can estimate training data requirements with the following functional form:

Train FLOP = (F FLOP / subj sec) x (H subj sec / effective horizon) x (KP^{α} effective horizons)

In the previous two sections, I generated probability distributions for F and P for the Neural Network family of hypotheses; in Part 2 I generated probability distributions for the scaling parameters α and K.

The effective horizon length H of the Neural Network hypothesis is currently my biggest source of uncertainty. I currently think all values in the range from 1 subjective second to multiple subjective years are plausible. In this section, I will:

- Discuss why neural networks trained on ML problems with short effective horizon lengths (e.g. a few subjective minutes) could likely do useful work of various kinds (more).
- Explain why a transformative ML problem may nonetheless require a long effective horizon length (e.g. a few subjective *years*), because having a transformative impact may require replicating cognitive abilities that natural selection may have optimized with multi-year generation times (more).
- Sketch out one form that a long horizon transformative model training run could take (more).
- Briefly describe several strategies that researchers may be able to use to reduce the effective horizon length of an ML problem which may naively seem to be long horizon (more).
- Discuss how we might estimate the effective horizon length of an ML problem which consists of many sub-skills learned over varying time horizons (more).

Short horizon models can probably do a lot of economically valuable work

It seems likely to me that a non-trivial fraction[4] (>5%) of economically valuable labor currently[5] done by humans could be made a lot more efficient or automated entirely by neural networks trained on short horizon ML problems using techniques very similar to current ML techniques.

[4] My guess is that this is probably true for most possible ways of defining "fraction of human labor", e.g. weighting by hours, wages, value-added measures, etc.

[5] Note that automating X% of the labor humans do as of 2020 will likely result in substantially less than X% reduction in labor force participation because of complementarity: technological innovations can create new work. My understanding is that historical waves of automation have tended to have little to no long-term effect on the fraction of humans who are employable -- e.g., technology has already automated away an outright majority of the human labor done in 1500 (mostly farming).

For example, here's a potential hypothesis to automating the handling of customer service calls (which have already been automated to a large degree using simpler computer programs over the last decade or two):

1. Train a generative language model on a large, generic corpus of text scraped from the internet, such that it is able to write basically grammatical, superficially sensible English in response to a broad range of different contexts.
2. Fine-tune the model to predict customer service representatives' responses in transcripts of recorded customer service calls.
3. Feed the model real-time transcripts of customers' speech as new calls come in, and have it predict responses. Convert the model's text to speech in real time to carry on a conversation with the customer, potentially allowing a human to jump in when necessary.
4. Continually improve the model using RL from human preferences: elicit quantitative ratings of the model's quality of service from call transcripts from human overseers or customers themselves, train a reward model to predict humans' quality ratings from call transcripts, and train the model to optimize predicted human score with reinforcement learning.

This is likely to be a relatively short horizon learning problem. Conservatively, the horizon length might be the length of an average customer service phone call, which are often quite short (e.g. a few minutes long). However, my best guess is that it would be substantially shorter than the average call length. Simply learning to generate idiomatic English that makes even vague sense in context is likely to be a significant fraction of the work of training the whole model, and the horizon length for "grammatical and vaguely sensical English" is probably closer to 1 token than multiple minutes. Additionally, it seems likely that in a lot of cases, the appropriate response to a statement by the customer can more-or-less be determined locally, without explicitly looking ahead to the end of the entire interaction (for example, a single interaction may involve dealing with multiple separate issues).

A similar process could be done for generating code. As of the time of writing in July 2020, language models are already being trained to predict what code humans will type next:

- TabNine, trained in 2019, is a code autocompletion service based on the open-source version of GPT-2 (~1.5B parameters) and fine-tuned on ~2 million GitHub files.[6]
- Microsoft's IntelliCode Compose (paper released in May 2020) is a code completion service based on a generative model GPT-C (~0.37B parameters) trained on ~1.2 billion lines of code.
- Very soon after, OpenAI (in a partnership with Microsoft) revealed an improved code-prediction model in a demo at the May 2020 Microsoft Build conference.

[6] "Deep TabNine is trained on around 2 million files from GitHub. During training, its goal is to predict each token given the tokens that come before it….Deep TabNine is based on GPT-2, which uses the Transformer network architecture. This architecture was first developed to solve problems in natural language processing. Although modeling code and modeling natural language might appear to be unrelated tasks, modeling code requires understanding English in some unexpected ways." July 2019 blog post Autocompletion with deep learning.

Such models seem to already be a little bit useful for speeding up routine coding, and potentially helping to catch some mistakes. They could likely be further improved with short horizon reinforcement learning, using a reward model trained on signals constructed from a variety of cues such as whether the code compiles/runs without errors, whether it passes unit tests (which may sometimes be partially written by a version of the model itself), human reviewer ratings, what edits are suggested by a human collaborator, etc.

Other categories of work that seem as if they could likely be done substantially or entirely by short horizon neural networks include:

- **Customer service and telemarketing.** Each interaction with a customer is brief, but ML is often required to handle the diversity of accents, filter out noise, understand how different words can refer to the same concept, deal with customization requests, etc. This is currently being automated for drive-thru order taking by the startup Apprente (acquired by McDonald's).
- **Personal assistant work:** This could include scheduling, suggesting and booking good venues for meetings such as restaurants, sending routine emails, handling routine shopping or booking medical and dental appointments based on an understanding of user needs, and so on.
- **Research assistant work:** This could involve things like copy-editing for grammar and style (e.g. Grammarly), hunting down citations on the web and including them in the right format, more flexible and high-level versions of "search and replace", assisting with writing routine code or finding errors in code, looking up relevant papers online, summarizing papers or conversations, etc.
- **Repetitive but dextrous manual labor** done by humans in factories, such as tying knots, painting, glueing, folding, etc. My understanding is that these types of work have been difficult to automate using classical robotics because there are tiny variations across different instances of the task that make it difficult to program a simple trajectory for robotic actuators. Such models could potentially be trained partially in simulation, as as the OpenAI Rubik's cube robot was, to cut down on physical capital costs.

The general theme is that work that is naturally broken up into short, fairly repetitive chunks for which we can clearly define either a training data distribution (as in medical diagnostics or customer service) or a training environment (as in factory work) can likely be cast as a short horizon ML problem. Note that it might not currently be economically or logistically feasible to train some of these models -- the only claim I am making is that it's likely we could *design a short horizon ML problem* such that a sufficiently large model trained on that problem using an amount of data/experience in line with what we would expect from existing training runs would successfully automate the relevant type of work.

We may need long horizons for meta-learning or other abilities that evolution selected for

Training a model with SGD to solve a task generally requires vastly more data and experience than a human would require to learn to do the same thing. For example, esports players generally train for a few years to reach professional level play at games like StarCraft and DOTA; on the other hand, AlphaStar was trained on 400,000 subjective years of StarCraft play, and the OpenAI Five DOTA model was trained on 7000 subjective years of DOTA. GPT-3 was trained on 300 billion tokens,[7] which amounts to about 3000 subjective years of reading given typical human reading speeds; despite having seen many times more information than a human about almost any given topic, it is much less useful than a human for virtually all language-based jobs (programming, policymaking, research, etc).

I think that for a single model to have a transformative impact on its own, **it would likely need to be able to learn new skills and concepts about as efficiently as a human, and much more efficiently than hand-written ML algorithms like SGD.** For a model trained in 2020 to accelerate the prevailing rate of growth by 10x (causing the economy to double by ~2024), it seems like it would have to have capabilities broadly along the lines of one of the following:

- Automate a wide swathe of jobs such that large parts of the economy can ~immediately transition to a rate of growth closer to the faster serial thinking speeds of AI workers, or
- Speed up R&D progress for other potentially transformative technologies (e.g. atomically precise manufacturing, whole brain emulation, highly efficient space colonization, or the strong version of AGI itself) by much *more* than ten-fold, such that once the transformative model is trained, the relevant downstream technology can be developed and deployed in only a couple of additional years in expectation, and then *that* technology could raise the growth rate by ten-fold. For AI capable of speeding up R&D like this, I picture something like an "automated scientist/engineer" that can do the hardest parts of science and engineering work, including quickly learning about and incorporating novel ideas.

Both of these seem to require efficient learning in novel domains which would not have been represented fully in the training dataset. In the first case, the model would need to be a relatively close substitute for an arbitrary human and would therefore probably need to learn new skills on the job as efficiently as a human could. In the second case, the model would likely need to efficiently learn about how a complex research domain works with very little human assistance (as human researchers would not be able to keep up with the necessary pace).

Humans may learn more efficiently than SGD because we are able to use sophisticated heuristics and/or logical reasoning to determine *how* to update from a particular piece of information in a fine-grained way, whereas SGD simply executes a "one-size-fits-all" gradient

[7] "All models were trained for a total of 300 billion tokens." Brown et al 2020, pg 8.

update step for each data point. Given that SGD has been used for decades without improving dramatically in sample-efficiency, I think it is relatively unlikely that researchers will be able to hand-design a learning algorithm which is in the range of human-level sample efficiency.

Instead, I would guess that **a transformative ML problem would involve meta-learning** (that is, using a hand-written optimization algorithm such as SGD to find a model which itself uses its own internal process for learning new skills, a process which may be much more complex and sophisticated than the original hand-written algorithm).

My best guess is that human ability to learn new skills quickly was optimized by natural selection over many generations. Many smaller animals seem capable of learning new skills that were not directly found in their ancestral environment, e.g. bees, mice, octopi, squirrels, crows, dogs, chimps, etc.

The larger animals in particular seem to be able to learn complex new tasks over long periods of subjective time: for example, dogs are trained over a period of months to perform many relatively complex functions such as guiding the blind, herding sheep, assisting with a hunt, unearthing drugs or bombs, and so on. My understanding is that animals trained to perform in a circus also learn complex behaviors over a period of weeks or months. Larger animals seem to exhibit a degree of logical reasoning as well (e.g. the crow in the linked video above), which seems to help speed up their learning, although I'm less confident in this.

This makes me believe it's likely that our brain's architecture, our motivation and attention mechanisms, the course of brain development over infancy and childhood, synaptic plasticity mechanisms, and so on were optimized over hundreds of millions of generations for the ability to learn and perhaps reason effectively.

The average generation length was likely several months or years over the period of evolutionary history that seems like it could have been devoted to optimizing for animals which learn efficiently. I consider this a *prima facie* reason to believe that the effective horizon length for meta-learning -- and possibly for training other cognitive abilities which were also selected over evolutionary time -- may be in the range of multiple subjective months or years. It could be much lower in reality for various reasons (see below), but anchoring to generation times seems like a "naive" default.

Here I am not saying we should expect that training a transformative model would take as much computation as natural selection (that view is represented by the Evolution Anchor hypothesis which I place substantially less weight on than the Neural Network hypotheses). I am instead saying:

1. A transformative model would likely need to be able to learn new skills and concepts as efficiently as a human could.
2. Hand-written optimization algorithms such as SGD are currently much less efficient than human learning is, and don't seem to be on track to improve dramatically over a short

period of time, so training a model that can learn new things as efficiently as a human is likely to require meta-learning.

3. It seems likely that evolution selected humans over many generations to have good heuristics for learning efficiently. So naively, we should expect that it could take an amount of subjective time comparable to the average generation length in our evolutionary history to be able to tell which of two similar models is more efficient at learning new skills (or better at some other cognitive trait that evolution selected for over generations).

My understanding is that meta-learning has had only limited success so far, and there have not yet been strong demonstrations of meta-learning behaviors which would take a human multiple subjective minutes to learn how to do, such as playing a new video game.[8] Under this hypothesis -- assuming that training data is not a bottleneck -- the implicit explanation for the limited success of meta-learning would be some combination of a) our models have not been large enough, and b) our horizons have not been long enough.

This seems like a plausible explanation to me. Let's estimate the cost of training a model to learn how to play a new video game as quickly as a human can:

- **Effective horizon length:** Learning to play an unfamiliar video game well takes a typical human multiple hours of play; I will assume the effective horizon length for the meta-learning problem is one subjective hour.
- **Model FLOP / subj sec and parameter count:** Even if our ML architectures are just as good as nature's brain architectures, it seems plausible that models much smaller than the size of a mouse brain aren't capable of learning to learn complex new behaviors at all -- my understanding is that we have some solid evidence of mice learning complex behaviors,[9] and more ambiguous evidence about smaller animals. According to Wikipedia, a mouse has about ~1e12 synapses in its brain, implying that its brain runs on ~1e12 FLOP/s.[10] I will assume we need a model larger than the equivalent of a bee but smaller than the equivalent of a mouse (say at least ~3e9 parameters and 1e11 FLOP / subj sec) to perform well on the "learning to learn new video games" ML problem.

[8] While GPT-3 displays impressive within-context few shot learning at a wide variety of natural language processing tasks, my impression is that so far it has not performed tasks which would require the typical *human* to learn something new, or would otherwise require humans to think over a period of multiple subjective minutes. For the most part, these seem to be tasks that a human would know how to do very well from the description alone, and which would take a human <1 minute to execute, consistent with the effective horizon length of GPT-3 being relatively short on downstream tasks.

[9] The behaviors I have seen mice learn seem substantially shorter horizon and less complex than learning new video games, but the mice are managing to do this despite having been optimized for survival and reproduction in the mouse ancestral environment rather than directly for "learning new tricks", so I would expect a mouse-sized meta-learned model to be substantially better than mice at learning, while being worse at various other mouse-survival-and-reproduction-relevant behaviors.

[10] That is, assuming that mice neurons fire about the same number of times per second on average as human neurons, and that the conversion from "synaptic firing events" to "FLOP" is the same for both species.

If the scaling behavior follows the estimate generated in Part 2, the amount of computation required to train a model that could quickly master a new video game should be (3600 subj sec) * (1e11 FLOP / subj sec) * (1700 * 1e11^0.8) = 2e25 FLOP. At ~1e17 FLOP per dollar,[11] that would cost $200 million, which makes it unsurprising this hasn't been successfully demonstrated yet, given that it is not particularly valuable.

Note that while meta-learning seems to me like the single most likely way that a transformative ML problem could turn out to have a long horizon, there may be other critical cognitive traits or abilities that were optimized by natural selection which may have an effective horizon length of several subjective months or longer.

What might a long horizon transformative ML problem look like?

A particularly salient vision for training a long horizon transformative model involves multiplayer self-play: the idea is to create a population of several instances of the agent who must compete and cooperate with one another to accumulate reward on a wide variety of rich and complex simulated environments over the course of episodes lasting multiple subjective years.[12] The agent is trained over many episodes with traditional reinforcement learning and/or evolutionary methods to maximize the total reward it gets in an average long horizon self-play episode.

This is intended to create a selection pressure toward "general intelligence" similar to the selection pressure over evolutionary history. The idea is that the need to compete with and cooperate with other humans to successfully navigate a complicated physical environment resulted in substantial fitness benefits accruing to humans who were slightly more intelligent than their peers (e.g. they could acquire more allies by being more persuasive and more useful, they could more easily deceive and defeat enemies, etc), resulting in a "general intelligence arms race" and an explosion of increasingly sophisticated strategies and counter-strategies.

Most zero-sum games are trained with self-play. OpenAI's "Emergent Tool Use from Multi-Agent Autocurricula" is a very small-scale application of this idea to more than two players -- two teams with at least two agents per team play hide-and-seek in a simple simulated environment, and spontaneously generate strategies and counter-strategies that they weren't explicitly taught over the course of many games.

Note that we don't strictly need to have a competitive, interactive environment for this hypothesis to work; we could instead train a model on much the same types of problems as in the short horizon hypothesis (e.g. writing code, generating text, manipulating robots) while receiving feedback at much longer intervals. For example, the model might spend a few

[11] This is optimistic for a well-optimized training run in 2020, and definitely too optimistic for earlier time periods.

[12] Note that because each effective horizon must be multiple subjective years long, it would likely take too much wall-clock time to run the model in real time and train it using a lot of interaction with humans; it would be necessary to speed up the model relative to real time and rely on human interaction for only a tiny proportion of its data.

subjective years (perhaps corresponding to a few hours in wall-clock time) attempting to prove a certain theorem or write a novel, and a human overseer could provide feedback on the result of this attempt. This approach could also be combined with self-play.

Reasons to think effective horizon length may be shorter

Despite the *a priori* argument given above that solving a transformative task with ML is likely to involve long horizon training, there are reasons to think researchers could devise transformative ML problems with shorter effective horizon lengths:

- **Human feedback or demonstrations could provide short horizon proxies for long-term value:** Even if the policy's actions have to be optimized to maximize expected reward over the next several minutes, we may be able to design the reward signals themselves to reflect much longer-term consequences.
    - One option is imitation learning. A short horizon model might be trained to generate behaviors that are hard to distinguish from behavior transcripts generated by human experts.
    - Another option is RL from human feedback. If human overseers have a good understanding of what immediate behaviors are likely to result in good long-term outcomes, they can directly reward those behaviors. As an analogy, small children are generally myopically optimizing for adults' short-term approval, but the adults in their lives can provide approval for behaviors (e.g. homework or tooth-brushing) which will pay off over much longer timescales.

  Imitation and human feedback can be particularly powerful if learning a reward model (learning to accurately predict how much humans would approve of an action) or a discriminator (learning to accurately predict how much an action deviates from what a human would do in the same context) requires many fewer parameters than learning a policy (learning how to *actually perform* actions that humans would approve of and/or that are hard to distinguish from humans' actions). In that case, a relatively small amount of human feedback or small number of behavior transcripts can be used to train a small reward model or discriminator, which can then be called a much larger number of times to train a large policy.
- **We might be able to train composable short horizon reasoning operations:** When humans make decisions optimizing for our long-run goals (e.g. pursuing a line of research or deciding whether to sell a company), we generally use some amount of explicit verbal planning and reasoning -- in other words, our decisions are governed by an internal structure, and we could explain how they were built up from shorter steps of reasoning if we needed to.[13] We may be able to train models to execute the kinds of short reasoning steps we use in our explicit verbal thinking, and compose those steps over and over again (e.g. in a long chain of deductive logic or a large tree search) to

[13] e.g., "I decided to sell my company because they made a good offer. I think their offer is good because I don't expect my company is actually worth that much. I don't think it's worth that much because I would need to expand my user base a lot to justify that valuation, and I don't have any good ideas for expanding my user base…."

make a complex decision. We could then update the model to directly imitate the ultimate output of this large, slow process. In this way, what seems like a long horizon problem (making a decision based on long complex reasons) can be learned as a short horizon problem. This is essentially the same idea as Christiano's Iterated Distillation and Amplification, Anthony et al's Expert Iteration algorithm, and the algorithm used to train DeepMind's MuZero agent and earlier AlphaZero and AlphaGoZero agents.

- **Some short horizon behaviors might naturally generalize to longer horizons:** For some types of problems, it might be the case that the most "natural" way[14] to perform well on shorter instances of the problem is to discover a strategy that generalizes robustly to larger instances. For example, it seems likely to me that a sufficiently large model trained on a distribution of arithmetic problems of increasing size (one-digit, two-digit, … N-digit addition, subtraction, multiplication, and division) would eventually start to perform perfectly on arbitrarily large (>> N-digit) arithmetic problems, because it seems likely that implementing the underlying rules of arithmetic is the most natural way to perform well on arithmetic problems of a wide variety of sizes. We may be able to design a transformative ML problem which has this property -- or has certain important subproblems with this property which reduce the effective horizon length of the whole problem. On priors, we should expect this to be the case to some extent, since humans are capable of acting to optimize outcomes over periods of time substantially longer than one generation.
- **Longer-horizon behaviors may require a much smaller number of examples to learn:** For some ML problems, achieving strong performance might requiring learning *some* very long horizon behaviors, but those behaviors might be very simple -- representable by a small number of parameters and therefore requiring a very small number of long horizon data points to learn. As an artificial example, consider a problem in which the RL agent must learn to control many complicated actuators in its body in order to move (e.g. a much larger version of QWOP). When the agent can get moving at all, it has the option of moving left or right. Let's say that moving left for 10,000 timesteps will result in a reward of +1, and all other behavior sequences result in a reward of 0. The subproblem of "Figure out which direction to move" has a horizon length of 10,000 but the optimal policy ("Keep going left") is very simple to represent; on the other hand, the subproblem of "Figure out how to move" is extremely complex, but has a much shorter horizon length. The effective horizon length of the entire problem might be something like $(N \times 1 + M \times 10^4) / (N + M)$, where N is much larger than M; this could end up being much closer to 1 than 10,000. It is unclear how much this toy example is representative of real-world RL training runs because it is very difficult to understand what "fraction" of an RL agent's "capacity" is being directed to certain sub-skills (or whether that's a reasonable ontology in the first place). It is plausible to me that it is somewhat representative -- for example, while a typical game of DOTA takes ~20 minutes to play, the OpenAI DOTA bot achieved strong play with a horizon length of only ~2 minutes, largely because its very strong "micro" play (i.e. skill at fast-paced

[14] That is, the way that is most likely to be discovered by the optimization algorithm (e.g. SGD).

combat tactics) made up for its mediocre "macro" play (i.e. whole-game strategy). It achieved *decent* "macro", but did not end up having to learn its intricacies really well in order to defeat humans at DOTA; it is plausible that a smaller fraction of its parameters were devoted to macro and it therefore needed a smaller number of long horizon examples than short horizon examples. The relatively small information content of the human genome may be a hint that there is relatively little to learn over horizons of multiple subjective years; see the discussion of multiple horizon lengths below and the Genome Anchor hypothesis further down for more detail.

- **Some key skills may naturally have a somewhat shorter effective horizon length:** Some economically valuable behaviors may have a "natural" horizon length closer to the range of "several hours or days" rather than "several years" because ground-level feedback may be easier to collect. For example, a significant fraction of the role of a venture capitalist might be well-described as a series of roughly independent "episodes" in which the VC learns about a new company before deciding whether to invest in it and if so how much; a model that was superhuman at this aspect of venture capital might add a lot of value and complement the skills of human VCs, even if it is not able to e.g. carry on long-running professional relationships. Similarly, law, consulting, IT, security, and medicine have some "episodic" elements of quickly orienting to a fresh problem and providing advice or making decisions about it on a limited timeframe. Particularly if we are looking for ways that models can *complement* humans rather than replace them one-for-one, many important aspects of many valuable roles could potentially be automated without necessarily requiring the model to have abilities that might have required long horizon meta-learning to acquire.
- **We may not need to run the agent the entire length of the task horizon:** For some tasks, we may need to see the consequences of an agent's actions play out in the world over a long period of time to estimate the reward, but the agent itself may only need to take actions for a relatively small portion at the beginning of each episode, after which the consequences take their own course until the agent's reward can be determined. Because I expect running the environment to be substantially cheaper computationally than running the agent, this could substantially reduce the effective horizon length.

These possibilities are not mutually exclusive. We may first select types of work that seem to have "natural horizon lengths" of several hours or days rather than months or years. It might then turn out to be the case that most of the behaviors learned over a horizon of 5 minutes generalize naturally to the timescale of days, and many of the remaining longer-horizon behaviors are simple enough that they can be learned with a very small number of day-long data points, not much impacting the average horizon length. We might then tackle the remaining difficult long horizon behaviors through a combination of human feedback, demonstrations, decomposition / Expert Iteration, and other techniques. If there are some important long horizon behaviors that we are still not able to train with an affordable amount of data and computation, we might figure out ways for humans to make up for the weaknesses of the model while still extracting a huge amount of economic value from the model that might count as "transformative" impact.

I expect that exploiting these possibilities fully will require building up a lot of infrastructure and know-how, such that it is more plausible that tactics like these could be used to bring effective horizon lengths down to a few subjective minutes in 2030 or 2040 than it is in 2020. A more accurate version of this model would incorporate this by increasing the probability of shorter effective horizon lengths over time.

What might effective horizon length be if a mixture of horizons is required?

Reinforcement learning problems and generative modeling problems tend to involve simultaneously learning a number of sub-skills that play out over varying timescales. For example,

- A real-time strategy game like StarCraft will involve exercising strategic skills (which play out over the course of a whole game), tactical skills (which come up several times per game), and reflex-based skills (exercised continuously throughout a game).
- Language modeling involves modeling very local correlations between words (e.g. various grammatical rules or common phrases and idioms) as well as regularities that hold across much longer chunks of text (e.g. the fact that character names and descriptions must remain consistent over a novel).

In general, I expect that a model would have stronger performance on skills that it has had more opportunity to exercise in the training distribution -- e.g., I expect that language models will be more likely to make errors in logical consistency across a long narrative than grammatical errors, and I expect models trained to play real-time strategy games to be much more impressive at aspects of the game that require fast and precise reflexes than aspects which require good strategy.

If excellent performance at short horizon skills combined with lower performance at longer horizon skills is sufficient for overall success at the relevant task -- or if short horizon skills transfer very well to longer horizon skills -- then the total cost of training will be dominated by the cost of learning the skills which play out over short timescales, and the effective horizon length of the overall problem will be short; if not, the effective horizon length may be several orders of magnitude higher, dominated by the longest-horizon skills in the distribution.

It is likely that a transformative ML problem would also involve learning different behaviors over different timescales. How can we model the way these different timescales would average out into an overall effective horizon length? I have heard of three high-level perspectives, although many more are possible:

- **Short horizon training (especially generative modeling) will dominate:** Under this perspective, almost all of the difficulty and expense of training most practically valuable behavior with ML will come from generative modeling (i.e. training the model to predict what it will see next), which has a very dense reward signal. According to this perspective, generative training will cause the model to build up useful concepts and

representations that can then be transferred to a wide variety of downstream skills, such that the model can achieve human-level performance on those skills with very little additional fine-tuning.

- **Long horizon training (especially meta-learning) will dominate:** Under this perspective, a non-trivial fraction p (e.g. 1% or 10%) of the training data must be focused on key long horizon behaviors -- particularly the behavior of efficiently learning complex new skills over subjective months of years -- for the model to learn those behaviors adequately enough to have a transformative impact. The overall effective horizon length is therefore at least equal to p times the horizon length of the long horizon behaviors, a term which will likely dominate over the short horizon portion of training. Note that a perspective which expects training data to be distributed log-uniformly across several orders of magnitude of effective horizon length would be equivalent to this perspective, because roughly $1/n^{th}$ of the data would have a horizon length of 10^n. The Genome Anchor hypothesis, explored below, could be framed as a variant of this hypothesis which uses the information content of the human genome as an estimate of how large the "hard core" of long horizon behavior may be.
- **The number of data points at each timescale will decay exponentially:** Under this perspective, some fraction q of the training data can be focused on the shortest-horizon behaviors, and q of the remaining data can be focused on the second-shortest-horizon behaviors, and q of the remaining data can be focused on the third-shortest-horizon behaviors, and so on. This is one very simple way to model the idea that shorter-timescale skills may transfer to longer-timescale skills, but only partially or imperfectly; this could be how curriculum learning might turn out to work. The overall effective horizon length on this perspective is highly sensitive to the value of q. If we assume a range from 1 subjective second to 1 billion subjective seconds (~32 subjective years), setting $q = 0.2$ will result in an effective horizon length of ~4.4 subjective years, whereas setting $q = 0.8$ will result in an effective horizon length of 15 subjective minutes. This spreadsheet implements the exponential decay model of horizon lengths, allowing the user to choose different values of q.

I was personally most intuitively sympathetic to the second perspective when I began this investigation, but after reflecting and discussing more I am ultimately very uncertain which of these high-level stories will turn out to be most correct. Differing intuitions about this question seems to be an important driver of disagreement about TAI timelines for the set of people I have talked with.

Training FLOP distributions for neural network hypotheses

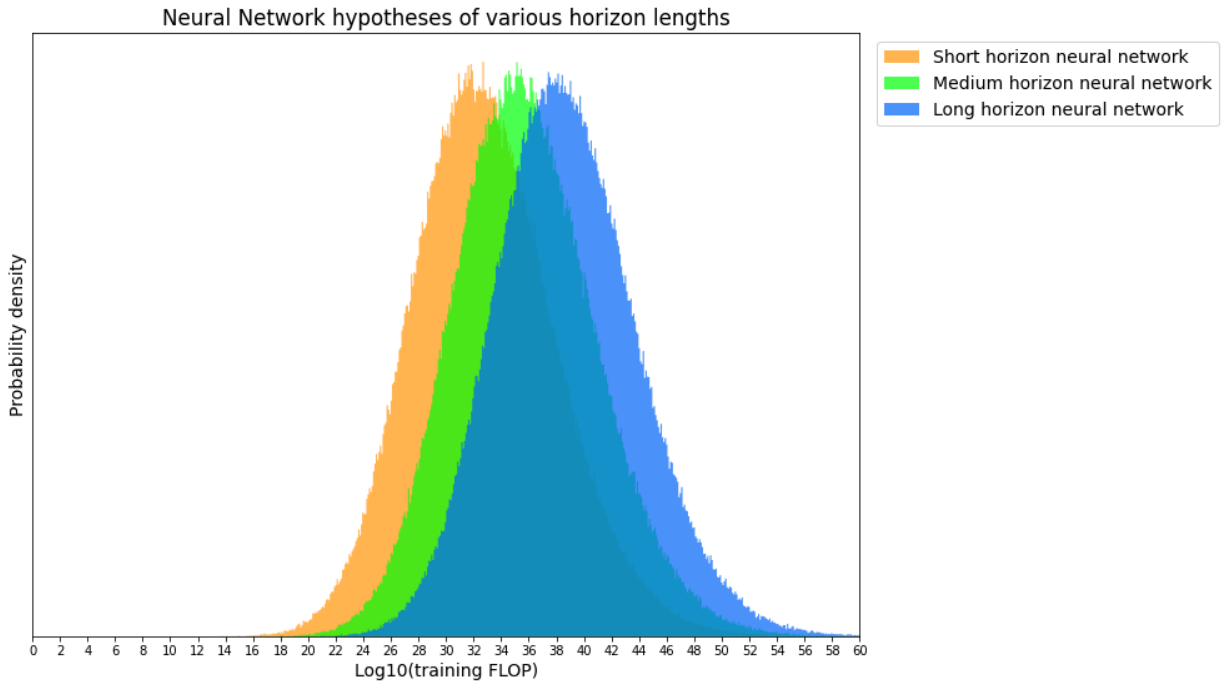
I divide the spectrum of possible horizon lengths somewhat arbitrarily into three categories:

1. **Short Horizon:** a log-uniform distribution between 1 subjective second and 1000 subjective seconds (~17 subjective minutes), with a median of **~32 subjective seconds**.

2. **Medium Horizon:** a log-uniform distribution between 17 subjective minutes and 1,000,000 subjective seconds (~1.7 subjective weeks), with a median of **~8.8 subjective hours**.
3. **Long Horizon:** a log-uniform distribution between ~8.8 subjective hours and 1,000,000,000 subjective seconds (~32 subjective years), with a median of **~1 subjective year.**

The categories were chosen to span an equal range in log-space (3 OOMs). The low end of the short horizon category was chosen because it is close to the effective horizon lengths of the shortest-horizon problems we solve today. The high end of the long horizon category was chosen because it is more than the amount of time that it would have taken for someone to grow up and successfully reproduce in the human ancestral environment (i.e., it is longer than the typical "horizon length" used in human evolution).

Here are my subjective distributions over the amount of computation it would take to train a transformative model according to each of these three hypotheses:



The median 2020 training FLOP requirements predicted by the Short Horizon hypothesis is **~1e31 FLOP**; the Medium Horizon hypothesis predicts a median of **~3e34 FLOP**; and the Long Horizon hypothesis predicts a median of **~1e38 FLOP.**

Other biological anchor hypotheses

I have thought most deeply about the three Neural Network hypotheses, and they collectively drive my views the most. In this section I describe my briefer investigations into three alternative biological anchor hypotheses:

- The Genome Anchor hypothesis, which anchors to the amount of information in the human genome and arrives at a median estimate of ~3e33 FLOP for 2020 training computation requirements (more).
- The Lifetime Anchor hypothesis, which anchors to the amount of computation done over the course of a human lifetime, and arrives at a median estimate of ~1e27 FLOP before updating against low-end FLOP levels, which pushes the median to ~1e28 FLOP (more).
- The Evolution Anchor hypothesis, which anchors to the amount of computation done over the course of evolution from early neurons, and arrives at a median estimate of ~1e41 FLOP (more).

Genome Anchor hypothesis

This hypothesis states that we should assume on priors that a transformative model would run on roughly as many FLOP / subj sec as the human brain *and have about as many parameters as there are bytes in the human genome (~7.5e8 bytes)*, and that we can extrapolate the amount of FLOP required to train such a model using an empirically-derived scaling law that expresses training data as a function of parameter count. It adjusts from the anchor points of human brain FLOP/s and human genome parameters by a relatively modest constant factor to account for qualitative considerations about how sophisticated our architectures seem to be as of 2020.

Like the Neural Network hypotheses, this hypothesis anchors on human brain FLOP/s to estimate model FLOP / subj sec. I will be using the same probability distribution over FLOP / subj sec derived in Part 1, centered around ~1e16 FLOP / subj sec, which is ~1 OOM higher than my estimate for human brain FLOP/s. Like the Long Horizon Neural Network hypothesis, this hypothesis assumes that **we need to train a transformative model to have some critical cognitive ability (such as sample-efficient learning) that evolution optimized for,** and that the ML problem of finding a model with this ability would have a long effective horizon length, comparable to the generation length over evolutionary history.

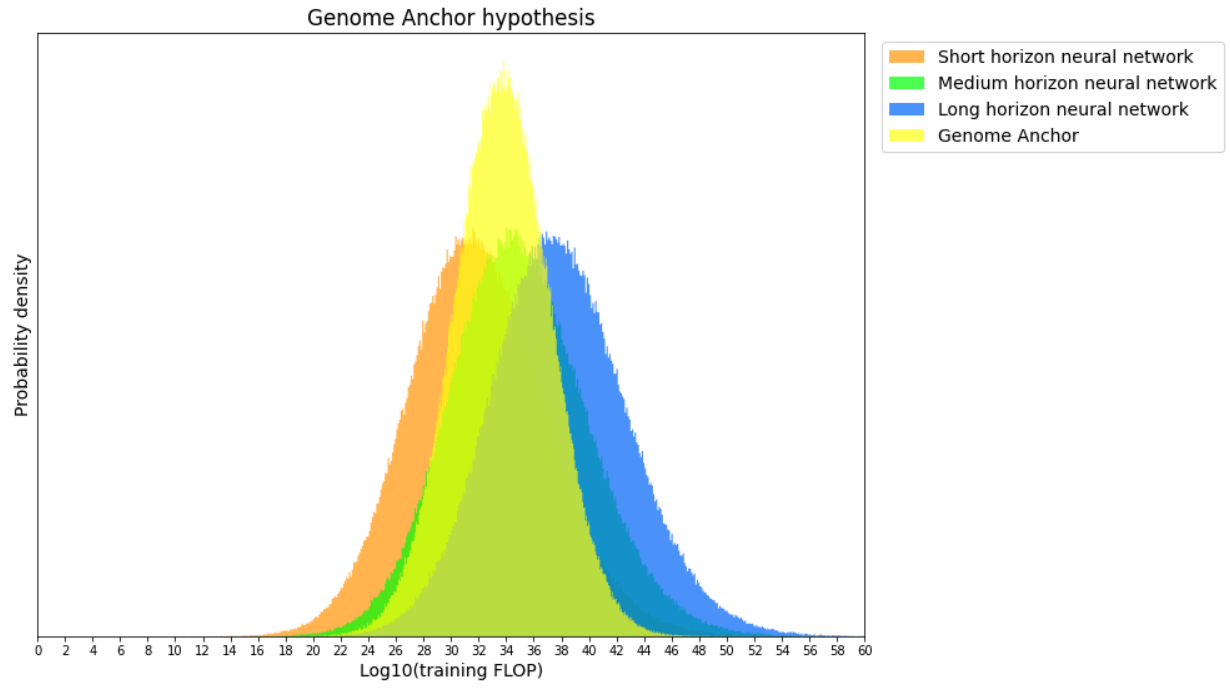
The key difference is that this hypothesis assumes that researchers could execute a long horizon training run like the one described above using many orders of magnitude fewer parameters than the Long Horizon Neural Network hypothesis assumes, anchoring parameter count to the amount of information we believe can be stored in the human *genome*. The motivating intuition is that evolution performed a search over a space of small, compact genomes which coded for large brains rather than directly searching over the much larger space

of all possible large brains, and human researchers may be able to compete with evolution on this axis.

According to Wikipedia, the human genome contains 750 megabytes (7.5e8 bytes) of information; my impression is that this figure is not very uncertain, and the conception of "information in the genome" is much less conceptually fraught than the concept of "brain FLOP/s", but I have not dug into this question. For simplicity, I centered the distribution of parameter count for this hypothesis around 7.5e8 rather than attempting to think about how I would shift it,[15] and assumed ~3 OOM uncertainty on either side to match the amount of uncertainty I assumed when estimating how much larger or smaller a transformative model would be compared to the brain.

I assumed that the effective horizon length was log-uniform between ~3e7 subjective seconds (~1 subjective year) and 1e9 subjective seconds (~32 subjective years), for an average of ~5 subjective years. I also assumed that the scaling behavior is the same as the one derived in Part 2 and used in the Neural Network hypotheses.

The yellow distribution in the image below is the subjective distribution over the amount of FLOP it would take to train a transformative model that runs on ~1e16 FLOP / subj sec under these assumptions:



[15] A more careful accounting of this hypothesis would likely need to consider both how human-designed architectures would compare to the genome and what "fraction" of the genome is likely devoted to relevant features of brain development.

This distribution is narrower than the corresponding distributions for the Neural Network hypotheses primarily because the parameter count distribution is narrower since I took a simple point estimate for the number of bytes in the human genome rather than the wide distribution I used for the human brain FLOP/s anchor. Secondarily, uncertainty over the sample complexity scaling exponent doesn't translate into quite as much uncertainty over the number of total data points required because the parameter counts in question are similar to the number of parameters in existing models.

How plausible is this hypothesis? There are at least two distinct ways to interpret this hypothesis:

- One possibility is that the hypothesis is claiming that researchers in 2020 would be able to design an architecture for a transformative model which only contains ~7.5e8 parameters given ~2-5 years of trying (per the definition of technical difficulty in Part 1).
- Another possibility is that the Genome Anchor hypothesis is not directly making a claim that researchers will quickly create a genome-like architecture, but simply using the genome as a way to bound what fraction of data points would need to be long horizon when training an ordinary neural network. This would make it a more specific version of the "Long horizon training will dominate" view described above, which takes a view on the exact number of long horizon data points.

I am quite skeptical of the first interpretation, that researchers could quickly design a genome-like architecture given the current state of ML algorithms:

- Such an architecture is very different from typical architectures we use today, which perform a much smaller number of FLOP per parameter per timestep (e.g., ~1-100 rather than millions). A normal neural network in which each parameter performs a small number of operations per timestep is a very agnostic architecture that doesn't require researchers to understand how the brain works much; reducing parameter count by a factor of over 1 million from that agnostic starting point is a substantial step toward ordinary programming on the spectrum from "pure ML" to "pure programming", and seems like it would require much more specific knowledge on the part of researchers.
- If it were easy to achieve impressive results in language modeling, games, image classification, and so on using such a small ratio of parameters to computation per timestep -- while still training on a small number of data points per parameter -- researchers likely would have been doing this already because it would have saved a lot on training computation. The fact that they haven't suggests that this version of the hypothesis would require that researchers either come to understand how the human genome codes for the development of the human brain and design an architecture that imitates this mapping, or else develop a new architecture at least as effective as that genotype-to-phenotype mapping despite not understanding the biological mechanisms in detail and not having found a similarly efficient architecture so far.

- Even if researchers could discover a mapping from ~7.5e8 parameters to a space of models running on ~1e16 FLOP / subj sec which contains a transformative program, we have very little evidence about whether current local optimization techniques like SGD could effectively search through that space. The optimization hypothesis may need to be more winding, or we may need to train much closer to convergence to achieve the desired level of performance, or we may not be able to use gradients, all of which would likely make sample complexity scale more poorly with parameter count than it does for today's architectures.

I find the second possibility more compelling, but am still somewhat skeptical on net. It seems like if either version of this hypothesis is correct, we should have seen more impressive results in meta-learning so far. Above, I argued that if a model needs to be at least 10% the size of a mouse brain to be able to learn new video games as well as a human, then training a typical neural network of that size on that problem would cost ~$200M in 2020. However, if the cost of meta-learning would be dominated by the cost to train as many parameters as there are in the mouse *genome*, this cost would be much lower.

Mice and humans have essentially the same number of base pairs in their genome,[16] implying 7.5e8 parameters would be required to represent a mouse. Assuming as above that the model could run on ~1e11 FLOP / subj sec and we need about one 1-subjective-hour-long data point per parameter, this would cost (1e11 FLOP / subj) * (3600 seconds / data point) * (7.5e8 data points) = $2.25 million; the fact that we do not yet have models that are capable of learning complex new skills such as games over the course of several subjective minutes despite various attempts feels like substantial evidence against this hypothesis.

Lifetime Anchor hypothesis

This hypothesis states that we should assume on priors that training computation requirements will resemble the amount of computation done by a child's brain over the course of growing to be an adult, because we should expect our architectures and optimization algorithms to be about as efficient as human learning. It anchors to this estimate and adjusts from this anchor by a relatively modest constant factor to account for qualitative considerations about how sophisticated our architectures and algorithms seem to be as of 2020.

[16] "All eutherian mammals have genomes of essentially the same size. It is instructive to consider the size of the mammalian genome in terms of the amount of computer-based memory that it would occupy. Each base pair can have one of only four values (G, C, A, or T) and is thus equivalent to two bits of binary code information (with potential values of 00, 01, 10, 11). Computer information is usually measured in terms of bytes that typically contain 8 bits. Thus, each byte can record the information present in 4 bp. A simple calculation indicates that a complete haploid genome could be encoded within 750 megabytes of computer storage space." Mouse Genome Informatics, Chapter 5 section 1.

Suppose it takes on average about 1 billion seconds (~32 years) for an intelligent human to go from an infant[17] to their peak level of productivity. If a human brain performs ~1e15 FLOP/s, that would be (1e15 FLOP/s) * (1e9 seconds) = 1e24 FLOP, only about 1 OOM larger than the amount of computation used to train AlphaStar. I am quite skeptical that this is the most appropriate anchor (see below), but here I will do my best to condition on the assumption that this is the best biological anchor to work with and training FLOP will be somewhere in this region, and generate the most plausible version of the hypothesis given that assumption.

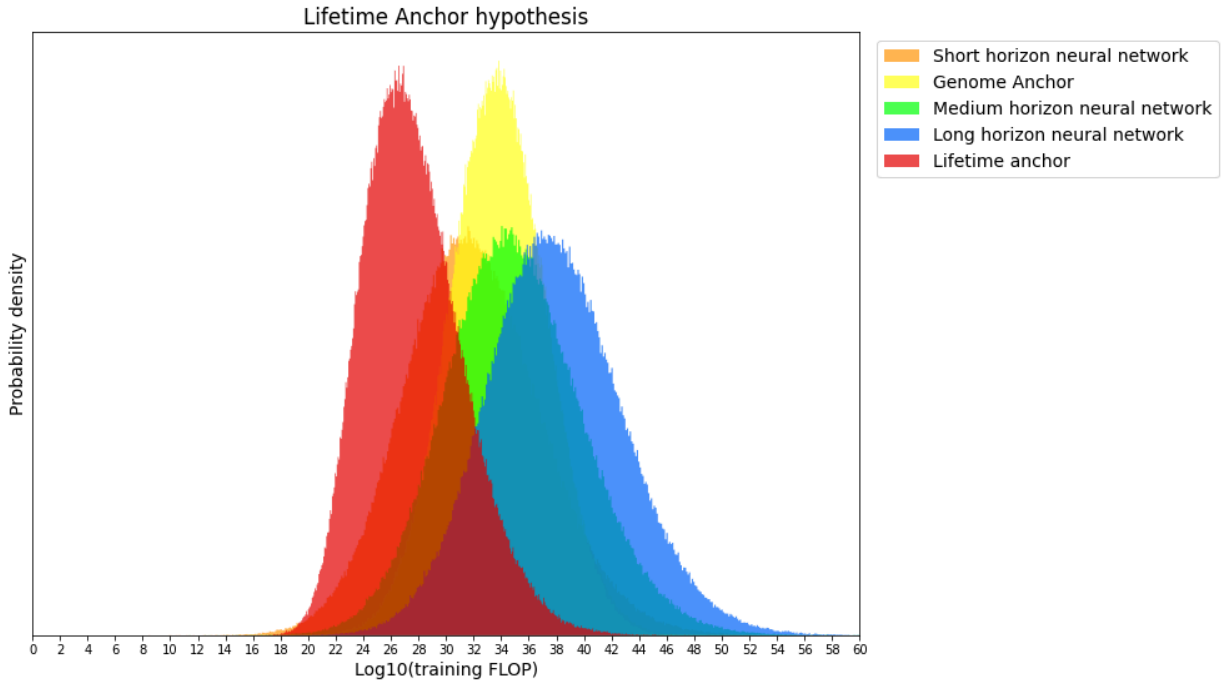
If ~1e24 FLOP is the most relevant biological *anchor*, where should we expect our training computation requirements would fall *relative* to that anchor? For the Neural Network and Genome Anchor hypotheses, I assumed that a transformative model would need to run on about ~1 OOM more FLOP / subj sec than the brain; I expect that if we are anchoring training FLOP on the human lifetime, we would need to shift the distribution by *more* than 1 OOM to the right. This is because:

- Many models we are training currently already require orders of magnitude more data than a human sees in one lifetime. For example, GPT-3 was trained on 300 billion tokens,[18] which amounts to about 100 billion subjective seconds or ~100 subjective lifetimes of experience given human reading speeds. AlphaStar was trained on 10,000 subjective lifetimes of experience, and the OpenAI Five DOTA model was trained on 200 subjective lifetimes. These models are sub-transformative, and scaling to TAI would likely not involve a major *reduction* in training data requirements, even if trends suggesting steep *increases* in data requirements are too pessimistic.
- In the neural network hypotheses, model FLOP / subj sec is correlated with model parameters, and model parameters determine how much data the model needs to train on. This means that the model FLOP / subj sec distribution essentially impacts the training FLOP distribution *twice*, because it also partially determines how many seconds of training the model requires. A shift of 1 OOM in model FLOP / subj sec translates into a shift of about ~2 OOM in training FLOP.
- Brain FLOP/s seems to me to be somewhat more analogous to "ongoing energy consumption of a biological artifact" while lifetime FLOP seems to be more analogous to "energy required to *manufacture* a biological artifact"; Paul's brief investigation comparing human technologies to natural counterparts, which I discussed in Part 1, found that the manufacturing cost of human-created artifacts tend to be more like ~3-5 OOM worse than their natural counterparts, whereas energy consumption tends to be more like ~1-3 OOM worse.
- Human babies may be born with various "baked-in priors" from evolution that make learning faster -- for example, they may have an intuitive understanding of what faces look like, or an intuitive grasp of the structure of human languages, intuitive priors about the visual world (for example that objects will be separated from other objects with

[17] The appropriate starting point for the purposes of this exercise may be conception, given that the work of translating from the genetic code into a brain development plan begins there, but it wouldn't make a material difference to the estimate.
[18] "All models were trained for a total of 300 billion tokens." Brown et al 2020, pg 8.

"edges") or an intuitive understanding of physics. Even if it is relatively trivial to somehow imbue ML models with these same priors (such that the human lifetime is still the most appropriate biological anchor to use), it may require somewhat more computation -- either because researchers may need to do some trial-and-error to determine how to "hard-code" these priors into a learning setup or because the model may need to do a constant amount of extra learning at the beginning to "catch up."

Somewhat arbitrarily, I settled on a median of ~3 OOM larger than the anchor. Here, I have generated a subjective probability distribution over the amount of computation that would be required to train a transformative model conditional on the Lifetime Anchor hypothesis:



To generate the compute requirements distribution, I multiplied the distribution over brain FLOP/s by 1e9 seconds, and multiplied the result by a skew-log-normal distribution with a median of ~3 OOM, a standard deviation of ~5 OOM, and a right skew (recall that I chose ~5 OOM as the spread based on my beliefs about algorithmic progress; see this section from Part 1). This expresses the view that there is a ~50% chance researchers can train a transformative model using at most ~1000 times the amount of computation done by the human brain from birth to age 32 -- and if that doesn't work, there is a long tail of larger levels of computation that might turn out to be necessary.

Note that this hypothesis does not make any direct assumption about the model size, instead directly anchoring on a certain value for total training FLOP. We can see that if a transformative model would require ~1e15 parameters and run on ~1e16 FLOP / subj sec, the Lifetime Anchor hypothesis predicts that the amount of computation required to train it is orders of magnitude smaller than the amount of data we would expect to need based on the

extrapolation from current models, even if we allow for an extremely short (<1 second) horizon length.

I think the most plausible way for this hypothesis to be true would be if a) it turns out we need a smaller model than I previously assumed, e.g. ~1e11 or ~1e12 FLOP / subj sec with a similar number of parameters, *and* b) that model could be trained on a *very* short horizon ML problem, e.g. 1 to 10 seconds per data point. Condition a) seems quite unlikely to me because it implies our architectures are much more efficient than brain architectures discovered by natural selection; I don't think we have strong reason to expect this on priors and it doesn't seem consistent with evidence from other technological domains. Condition b) seems somewhat unlikely to me because it seems likely by default that transformative ML problems have naturally long horizon lengths because we may need to select for abilities that evolution optimized for, and possible measures to get around that may or may not work.

The second way this hypothesis could turn out to be true would be if we need to use a large model (e.g. ~1e15 parameters) but manage to vastly outperform ML sample complexity trends on at least one transformative ML problem within ~2-5 years of effort. This also seems unlikely to me -- I don't think there are compelling reasons to believe that transformative ML problems would naturally have more favorable sample complexity scaling than current problems, and I am not aware of any plausible paths to dramatically improving sample complexity quickly.

Another major reason for skepticism is that (even with a median ~3 OOM larger than the human lifetime) this hypothesis implies a substantial probability that we could have trained a transformative model using less computation than the amount used in the most compute intensive training run of 2019 (AlphaStar at ~1e23 FLOP), and a large probability that we could have done so by spending only a few OOMs more money (e.g. $30M to $1B). I consider this to be a major point of evidence against it, because there are many well-resourced companies who could have afforded this kind of investment already if it would produce a transformative model, and they have not done so. See below for the update I execute against it.

More broadly, the Lifetime Anchor hypothesis contradicts the efficient-market hypothesis, implying that AI companies and the inputs to AI research are radically mispriced. It is also in tension with the general pattern observed across domains that technological progress tends to be broadly continuous -- it implies that a transformative model could be trained in the very near term, increasing the annual economic value-added of ML models from hundreds of billions of dollars to hundreds of trillions of dollars in a very short period, and taking the world from a ~3% growth rate to a ~30% growth rate all at once without first passing through intermediate levels of growth. While I don't believe that markets are always fully efficient or that technological progress is always continuous, I would still expect AI to be adding a lot more economic value than it is today, and to be priced much higher on the market, if this hypothesis were actually feasible.

Evolution Anchor hypothesis

This hypothesis states that we should assume on priors that training computation requirements will resemble the amount of computation performed in all animal brains over the course of evolution from the earliest animals with neurons to modern humans, because we should expect our architectures and optimization algorithms to be about as efficient as natural selection. It anchors to this estimate and adjusts by a relatively modest factor to account for qualitative considerations about how sophisticated our architectures and algorithms seem to be as of 2020.

Like the Long Horizon Neural Network hypothesis and the Genome Anchor hypothesis, this hypothesis assumes that we need to train a transformative model to have some critical cognitive ability (such as sample-efficient learning) that evolution optimized for. Unlike those hypotheses, it assumes that human-designed architectures and optimization algorithms have done very little to reduce the amount of computation that meta-learning would take compared to a baseline of simulating natural selection from a very primitive starting point such as a randomly-connected nerve net with dozens of cells.

As with the Lifetime Anchor hypothesis, I am skeptical that this is the most appropriate biological anchor to lean on in generating our 2020 compute requirements distribution, but in this section I will try to condition on the assumption that it is the most informative anchor.

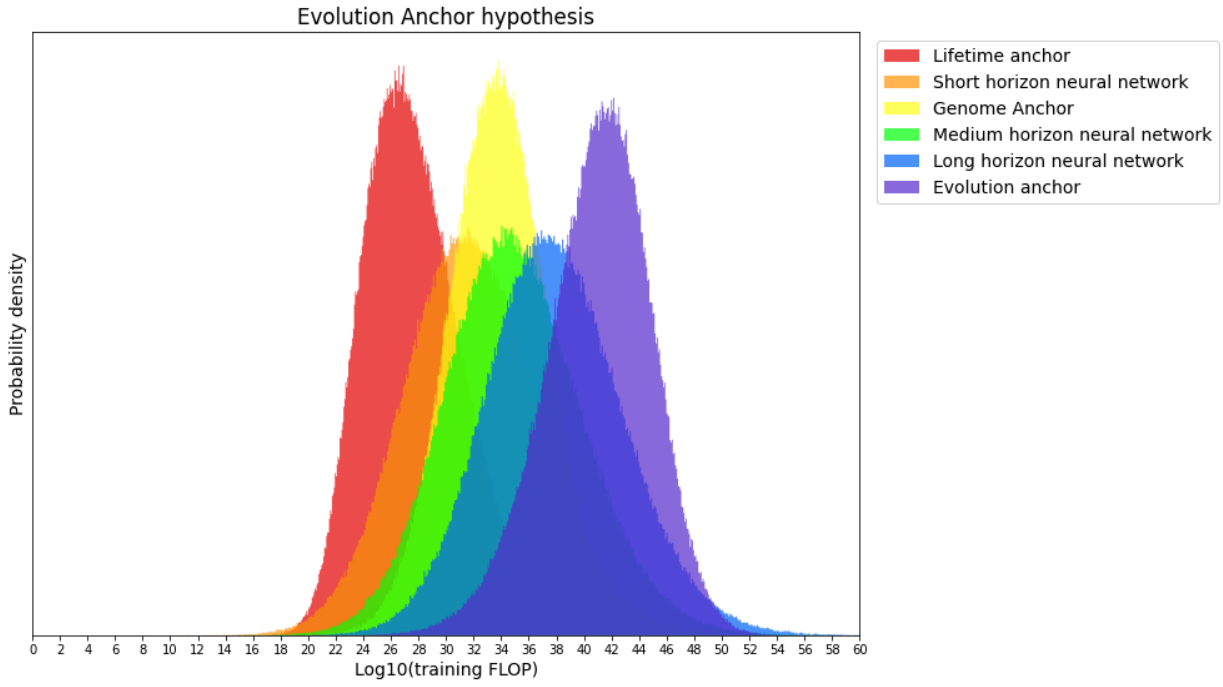
The amount of computation done over evolutionary history can roughly be approximated by the following formula: (Length of time since earliest neurons emerged) * (Total amount of computation occurring at a given point in time). My rough best guess for each of these factors is as follows:

- **Length of evolutionary time:** Virtually all animals have neurons of some form, which means that the earliest nervous systems in human evolutionary history likely emerged around the time that the Kingdom *Animalia* diverged from the rest of the Eukaryotes. According to timetree.org, an online resource for estimating when different taxa diverged from one another, this occurred around ~6e8 years ago. In seconds, this is ~1e16 seconds.
- **Total amount of computation occurring at a given point in time:** This blog post attempts to estimate how many individual creatures in various taxa are alive at any given point in time in the modern period. It implies that the total amount of brain computation occurring inside animals with very few neurons is roughly comparable to the amount of brain computation occurring inside the animals with the largest brains. For example, the population of nematodes (a phylum of small worms including *C. Elegans*) is estimated to be ~1e20 to ~1e22 individuals. Assuming that each nematode performs ~10,000 FLOP/s,[19] the number of FLOP contributed by the nematodes every second is ~1e21 * 1e4 = ~1e25; this doesn't count non-nematode animals with similar or fewer numbers of

[19] *C. Elegans* is estimated to have ~7,500 synapses.

neurons. On the other hand, the number of FLOP/s contributed by humans is (~7e9 humans) * (~1e15 FLOP/s / person) = ~7e24. The human population is vastly larger now than it was during most of our evolutionary history, whereas it is likely that the population of animals with tiny nervous systems has stayed similar. This suggests to me that the *average* ancestor across our entire evolutionary history was likely tiny and performed very few FLOP/s. I will assume that the "average ancestor" performed about as many FLOP/s as a nematode and the "average population size" was ~1e21 individuals alive at a given point in time. This implies that our ancestors were collectively performing ~1e25 FLOP every second on average over the ~1 billion years of evolutionary history.

This implies that the total amount of computation done over the course of evolution from the first animals with neurons to humans was (~1e16 seconds) * (~1e25 FLOP/s) = **~1e41 FLOP**. Here, I generated a distribution over training computation requirements for the Evolution Anchor hypothesis in purple:



To generate this distribution,

- I first generated a distribution for the amount of computation done in evolutionary history, incorporating my uncertainty about the length of the evolutionary period, the amount of computation that might be occurring in a nematode's nervous system, the average ancestor population size, and how much more or less computation we might need compared to the evolution anchor (using the same distribution generated in Part 1 for how much better or worse ML architectures might be compared to brain architectures) -- I have not investigated these estimates very thoroughly and I expect they could easily be improved with more research.

- I then multiplied that distribution by a skew-log-normal distribution with a median of ~0 OOM, a standard deviation of ~5 OOM, and a left skew. This expresses the view that that researchers could train a transformative model using as much computation as the amount of computation done in evolutionary history, and there is a substantial chance that smaller amounts would be sufficient as well, with the probability decaying slowly at smaller and smaller levels of computation.

Note that depending on the particular spirit of the hypothesis we are conditioning on (that the anchor of "FLOP done over the course of evolution" is the most appropriate biological anchor to use), I think we could argue for shifting the distribution further to the left rather than centering it exactly where the evolution anchor is centered -- just as I argued above that the Lifetime Anchor hypothesis distribution should be centered ~3 OOM to the right of its biological anchor. Below, I discuss some reasons to expect that we should be able to train a transformative model with substantially less computation than evolution. However, I have chosen to leave it alone in this context because:

- I wanted to simplify the report somewhat, and this hypothesis is the one I have spent the least time thinking about.
- There are plausible arguments that I have underestimated true evolutionary computation here in ways that would be somewhat time-consuming to correct: for example, it may be that we should somehow attempt to incorporate the evolution of precursors to true neurons, or that for early ancestors with very few neurons the amount of computation it would take to appropriately simulate their DNA would dominate the amount of computation to simulate their nervous systems. Not shifting the Evolution Anchor hypothesis distribution further to the left -- despite believing that there are relatively straightforward ways we could improve upon evolution, such as reducing population sizes or crafting less noisy fitness signals -- feels like an easy fudge to attempt to make up for this.
- The Long Horizon Neural Network hypothesis (in blue above) makes a number of reasonable assumptions, and I consider to be an attractive "naive default" view -- and it is actually relatively close to the Evolution Anchor hypothesis already. Shifting the Evolution Anchor hypothesis multiple OOMs to the left based on inside-view considerations of how we might improve on natural selection would cause it to be centered to the *left* of the Long Horizon Neural Network hypothesis. This feels somewhat strange because the spirit of this hypothesis involves believing that our optimization algorithms will be *less* efficient than SGD.
- I expect that some ML researchers would want to argue that we would need substantially *more* computation than was performed in the brains of all animals over evolutionary history; while I disagree with this, it seems that the Evolution Anchor hypothesis should place substantial weight on this possibility.

Like the Human Lifetime Anchor hypothesis, this hypothesis makes no assumptions about the size of the model, estimating training FLOP directly from a biological anchor. This could be achieved by training a very large model (e.g. one with ~1e18 FLOP / subj and a similar number

of parameters) using an amount of data that is in-line with extrapolations from existing models, or by training a smaller model using a lot more data than would be predicted from these extrapolations, or by training a model that grows in size over the course of the training run as animals' brains did over the course of evolutionary history, etc.

I think it is very likely that this hypothesis or some cheaper hypothesis would work out given 2020 architectures and algorithms; I discuss this more below.

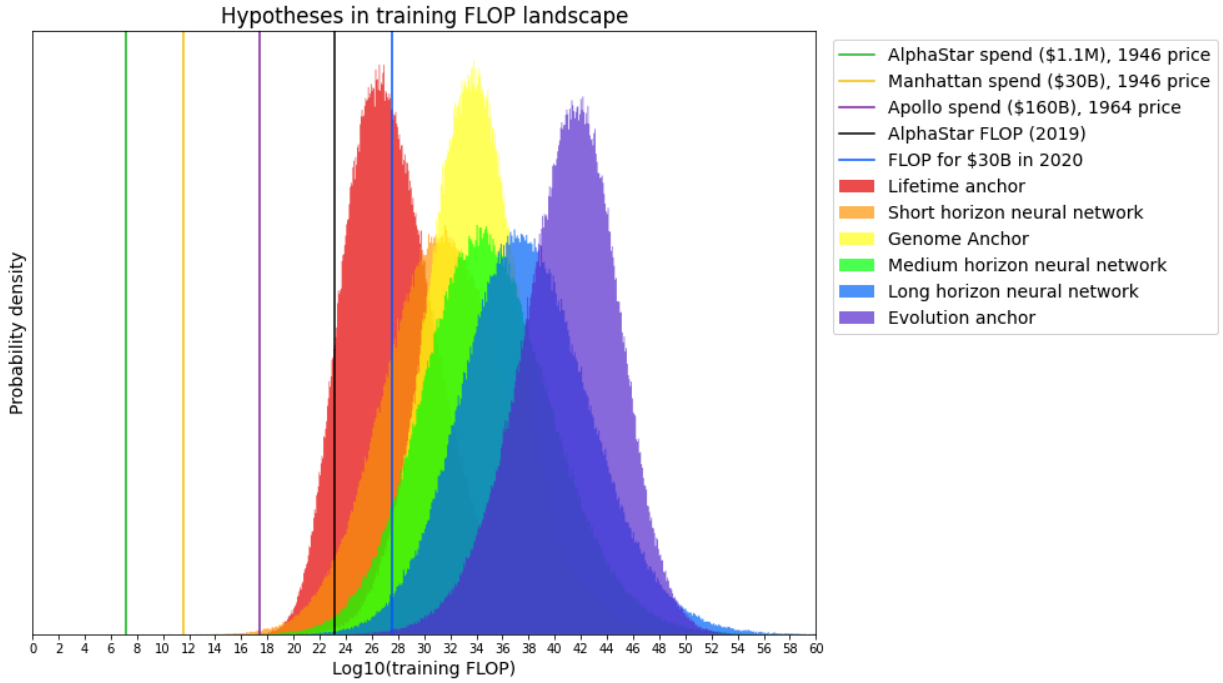
2020 training computation requirements distribution

In this section, I will attempt to use the hypothesis distributions generated in the previous sections to construct an overall 2020 compute requirements distribution. I will:

- Adjust the hypothesis distributions so that they assign a lower probability to the possibility that very low levels of computation (e.g. as much computation as models today are trained with) would be sufficient (more).
- Attempt to estimate the probability the amount of computation that would be required to train a transformative model in 2020 is much larger than any hypothesis would predict (more).
- Assign weights to each of the hypothesis distributions based on the qualitative discussions of the plausibility of each hypothesis given in the previous sections (more).
- Generate a probability distribution by creating a weighted mixture of the different biological anchor hypothesis distributions and the hypothesis representing "none of the above" (more).

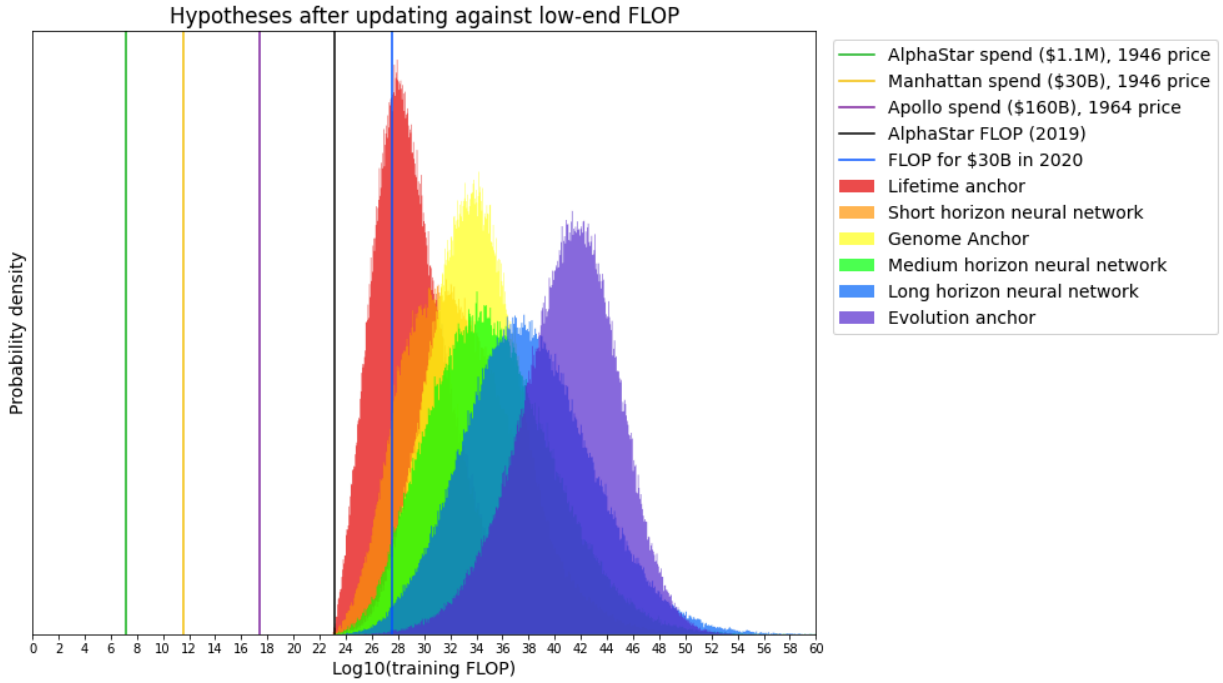
Updating against levels of computation that are already affordable

As I mentioned above, some hypotheses -- most notably the Lifetime Anchor hypothesis -- assign non-trivial probability to levels of computation small enough that we can already afford to do training runs of that size. To see this more clearly, we can display the hypotheses within the context of a landscape of FLOP levels that would have been attainable for various prices in the present or past:



The dotted lines in this landscape mark levels of computation that could theoretically have been purchased at various points in the past with various levels of investment (as well as markers for the median estimate of lifetime computation and evolution computation); see this appendix for details on that landscape. The black dotted line is the amount of computation that the AlphaStar training run used (~1e23 FLOP); the grey dotted line further to the right is the amount of computation that I estimate could be purchased for ~$30B in 2020.

I think it is unlikely that the amount of computation that would be required to train a transformative model is in the range of AlphaStar or a few orders of magnitude more -- if that were feasible, I would expect some company to have already trained a transformative model, or at least to have trained models that have already had massive economic impact. To loosely approximate a Bayesian update based on the evidence from this “efficient markets” argument, I truncate and renormalize all the hypothesis probability distributions:



Here, I have assumed that the probability that the required amount of compute is smaller than the amount of compute used to train AlphaStar (~1e23) should be set to ~0. Additionally, I assume that we have some amount of evidence that it's unlikely to be between 1e23 and 1e27, with the strength of that evidence getting weaker for higher values.[20] AlphaStar cost around ~$1M to train, and it seems likely that AI companies would have been willing and able to spend much more than that if they had reason to expect substantial profit. I assume that there is no particularly strong evidence against values larger than ~1e27 purely from an "efficient markets" standpoint.

To the extent that the prior (untruncated) distribution of a hypothesis assigned probability to levels of computation that we have updated against, we should consider that hypothesis less credible overall. The truncation removed ~30% of the probability mass in the Lifetime Anchor distribution and ~10% of the mass from the Short Horizon Neural Network distribution; the others were virtually unaffected, losing <5% probability mass.

Note that this loss of probability mass is not shown visually in the plots above, because the plotting software I use automatically renormalizes the histograms so that the total area under their curve adds up to 1; this means that when a distribution gets *narrower* -- as the Lifetime Anchor hypothesis did when low end values were cut off -- it also gets *taller.* Instead, I

[20] Specifically, I assumed that there was no evidence against values greater than 1e27 FLOP, that there was perfect evidence against values less than 1e23 FLOP, and interpolated log-linearly between those two values.

take this into account below when I assign weights to different hypotheses and combine them into a mixture distribution.

Probability that the required FLOP is larger than all hypotheses predict

What is the probability that 2020 training computation requirements for a transformative model are larger than the amount predicted by the Evolution Anchor hypothesis -- that is, that none of the biological anchor hypotheses would work out in any form[21] (even given access to the requisite data and environments)?

I would estimate a ~10% probability that 2020 compute requirements are larger than the Evolution Anchor hypothesis. From my current perspective, I think it could be defensible to assign a probability closer to 25% instead, but I don't currently see how someone would defend a probability of 50% or more (although I may be missing some relevant arguments). My reasoning is as follows:

- ML models have already made substantial progress on several tasks analogous to the types of behaviors animals were under strong natural selection pressure to perform well (e.g. object recognition and motor control) using vastly less training computation than it took to evolve those abilities. For example, if our ancestors collectively performed ~3e24 FLOP per second, then the amount of computation performed in the entire AlphaStar training run (~1e23) is the equivalent of less than one second of evolutionary history, implying that AlphaStar should only be as capable as the earliest animals with neurons (believed to be tiny jellyfish-like creatures).
- There are also some specific ways it seems that we could improve upon the "simulate natural selection" baseline, *prima facie*. For example, population sizes are a consequence of the carrying capacity of an ecological niche rather than being tuned to minimize the amount of computation used to evolve intelligent animals; it seems likely that they were far too large from a computational-efficiency standpoint. Additionally, the genetic fitness signal in the natural environment was often highly noisy, whereas we could plausibly design artificial environments which provide much cleaner tests of precisely the behaviors we are looking to select for.
- The Evolution Anchor hypothesis posits that training a transformative model would require computation levels almost *twenty orders of magnitude* larger than anything we could reasonably afford today; as mentioned above, this allows for room to train a model more than *eight orders of magnitude* larger than models we are training today, *and* to train it using much more data than extrapolations from current models would predict. I think there is little reason to expect that even domain experts would be justified in

[21] I intend to construe the hypotheses broadly here, meaning that if researchers come up with any training run which ends up using an amount of computation that is squarely in-distribution for hypothesis X, that would count as "hypothesis X worked out." It does not necessarily have to look like the sketches I presented in this section.

making a confident judgment that such an unprecedentedly huge training run couldn't accomplish a transformative task; most experts most of the time are not entertaining this possibility. If the Evolution Anchor hypothesis would work out, this is easily compatible with the views of experts who are highly pessimistic about machine learning-based transformative AI.

Assigning probabilities to each of the biological anchor hypotheses

If there is a ~90% probability that the Evolution Anchor hypothesis or something cheaper would work out given 2020 architectures and algorithms, how should we distribute that credence across hypotheses? Which hypothesis distribution is most likely to represent the *minimum* required amount of computation given 2020 architectures and algorithms?

This is more debatable and dependent on intuitive priors that may differ from person to person than the question of whether the Evolution Anchor hypothesis would work out. I have given some qualitative commentary in the sections above on how plausible each of the different hypotheses seems to me; here I will attempt to translate that into rough quantitative weights.

Long Horizon Neural Network as a "naive" default view

I consider the Long-Horizon Neural Network hypothesis to be a "naive default" view that errs conservative. This hypothesis assumes that we would need a model roughly ~1 OOM larger than the human brain to solve a transformative task, and then makes fairly standard or somewhat conservative assumptions about every other parameter:

- It assumes that a transformative model would look similar to contemporary architectures, and so would perform about ~1-100 FLOP per parameter per subjective second.
- It assumes that a transformative ML problem would likely involve meta-learning of some kind, and that meta-learning would have an effective horizon length of multiple subjective months or years.
- It assumes that the amount of data required to train this model will scale in a similar way to existing models -- that the number of instances of "trying to learn a new skill over the course of multiple subjective months" that the model will need to observe will scale close to linearly with its parameter count (an exponent of ~0.8).
- It assumes that there is no "shortcut" to training the relevant long horizon cognitive skills using decomposition, human feedback, and so on.

If it is the case that the computation that would be required to train a transformative model in 2020 is larger than the amount estimated by this hypothesis, then the reason is likely some combination of:

- **Our optimization techniques are fundamentally limited:** It won't be possible to use SGD (or other optimization techniques researchers could design in a few years of effort) to find a transformative model from a search space that contains one -- perhaps because

there would be astronomically many bad local minima in the loss landscape for every transformative model, and any optimization technique researchers try would likely get stuck in one of those local minima.
- **The quantitative estimates are too small:** Contemporary optimization techniques *would* work for finding a transformative model at *some* scale, but the amount of computation required would be larger than the amount estimated by any hypothesis distribution -- perhaps because we would actually need a model many orders of magnitude larger than the human brain (or the human brain is much more powerful than we think), or because the amount of training experience needed would be orders of magnitude larger than ten data points per parameter.

Both of these are possible, but I don't see a strong affirmative reason to be confident in either of these possibilities:

- Simple variants of SGD have been able to produce models capable of a wide variety of useful behaviors -- and some of these behaviors, most notably language modeling, seem like important sub-skills for many possible transformative tasks.
- There are some skills that seem important to many transformative tasks that our models have not yet displayed clearly, notably quickly learning new skills. However, as I argued above, it is plausible (and consistent with the biological anchors framework) that this is because models are not large enough for meta-learning to work well rather than because of a fundamental limitation.
- More generally, this framework predicts that machine learning models today should be substantially less "capable" than mice and only somewhat more capable than bees; this does not seem obviously inconsistent with what we in fact observe.

I would estimate an ~80% probability that the Long-Horizon Neural Network hypothesis or a cheaper hypothesis would work out given 2020 algorithms and architectures. I can see how someone would assign a probability closer to ⅓, but I don't currently see how someone could arrive at a probability much lower than that.

How likely is it that more aggressive hypotheses are right?

Hypotheses which predict that smaller amounts of computation will be sufficient are committed to denying one of the "naive" assumptions laid out above:

- The other Neural Network hypotheses assume that models can have a transformative impact without being able to plan over long horizons, or else that we can train models to plan adequately over long horizons without actually giving them direct experience with many long horizon episodes.
- The Genome Anchor hypothesis assumes that the amount of training data required to learn long horizon behaviors can best be estimated by anchoring to the amount of parameters in the human genome, rather than the amount of parameters in a typical neural network running on ~1e16 FLOP / subj sec.

- The Lifetime Anchor hypothesis does not make specific statements about training methods, but the most likely ways it could turn out to be true is if a) the model size required to solve a transformative task will be substantially smaller than the human brain *and* very short horizons will be sufficient, or b) researchers can relatively easily design a large model which learns some transformative task as efficiently as a human baby could (which may involve "hard-coding" a number of the priors such as intuitive physics, intuitive psychology, or proclivity for learning languages that human babies may have acquired over the course of natural selection).

On the other hand, the Evolution Anchor hypothesis relaxes the "default assumptions" further, allowing for the model size to be much larger than the human brain, and/or for sample complexity to scale much more poorly than for existing ML problems, and/or for horizons to be much longer than one subjective year.

For each of these six hypotheses, I'll assign a probability estimate to the event that the *minimum* amount of computation required to train a transformative model is distributed roughly as that hypothesis would predict:

- **Lifetime Anchor hypothesis:** I consider the Lifetime Anchor hypothesis to be unlikely. As we saw above, there is almost a factor of ~2 update against this hypothesis from the efficient markets argument; it also seems to be out of line with other evidence we have about ML systems -- for example, we have substantial evidence that training a model to solve a task consistently requires many more examples than teaching a human an analogous skill. I'll assign a ~5% probability to this hypothesis.
- **Short Horizon Neural Network hypothesis:** I think that there is a substantial possibility that effective horizon length may be much shorter than the conservative assumption made by the Long Horizon Neural Network hypothesis from some combination of proxy reward signals, decomposition, and short horizon sub-skills transferring to longer horizons (see above). In particular, several researchers I have spoken to consider it plausible that learning the skill of "predict what will happen next" will dominate the cost of training. I'll assign a ~20% probability to this hypothesis.
- **Genome Anchor hypothesis:** While I think it is unlikely that researchers could quickly discover an architecture which structurally behaves like the genome while having a similar scaling behavior to neural networks, I do find the genome anchor to be somewhat plausible as a way to think about what fraction of the training data for a neural network may need to be long horizon if a mixture of horizon lengths is required (although I don't currently think of it as strongly privileged over other ways of thinking about that question). I'll assign a ~10% probability to this hypothesis. This makes the total probability assigned to the three lowest-computation hypotheses ~35%.
- **Medium-Horizon Neural Network hypothesis:** I currently consider this hypothesis to be the most likely. It feels as if we should be able to solve substantially more impactful tasks with an effective horizon length in the range of subjective hours or days compared to subjective seconds or minutes. If the "naive" effective horizon length of meta-learning is multiple subjective months, it seems highly plausible that some combination of the

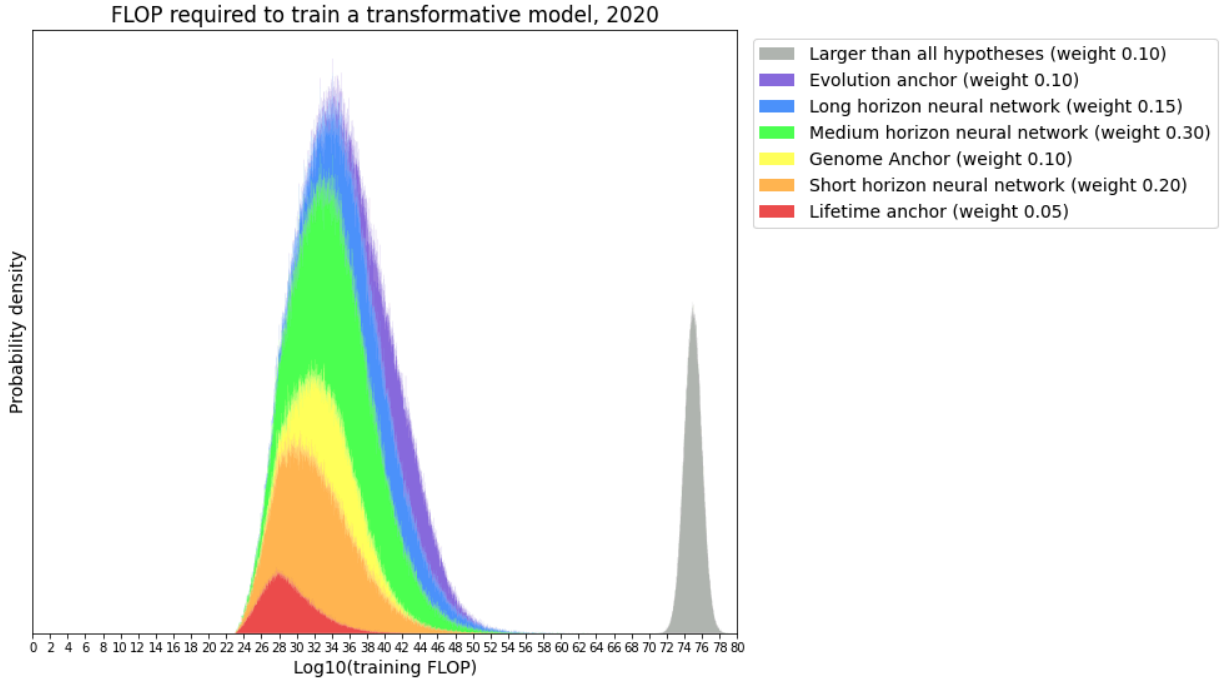
reasons described above may bring that down by ~2-3 OOM into the Medium Horizon range (as opposed to ~5-6 OOM into the Short Horizon range). I'll assign a ~30% probability to this hypothesis.

- **Long Horizon Neural Network hypothesis:** The above implies that the probability assigned to this hypothesis is ~15%, for a total of ~80% probability assigned to all the hypotheses which estimate lower training computation requirements than Evolution Anchor.
- **Evolution Anchor hypothesis:** The above implies that the probability assigned to this hypothesis is ~10%, for a total of ~90% probability assigned to all of the biological anchor hypotheses combined.

The weighted mixture of biological anchor hypotheses

To generate a combined 2020 compute requirements distribution from the hypotheses, I need to first somehow represent the possibility that the minimum amount of computation required to train a transformative model given current architectures and algorithms is larger than the Evolution Anchor hypothesis.

Because that possibility is outside the biological anchors framework, I don't have a way to estimate exactly *how* much larger the required amount of computation would be in that world. As a stand in, I simply generate an arbitrary "dummy distribution" to represent this possibility, and create a weighted sum of the seven different distributions using the weights described above:



This dummy distribution plays no role in the calculation of timelines and its location is completely irrelevant**;** it is purely a visual aid which allows the total size of the combined distribution containing the six hypotheses to grow or shrink based on the total probability assigned to “at least one hypothesis is correct” (otherwise the plotting software would automatically enforce that the total density is 1).

In the Part 4, I describe how I generate a probability distribution over when the amount of compute required to train a transformative model may become affordable; in that calculation, I make the somewhat conservative assumption that **if 2020 compute requirements are larger than the Evolution Anchor, we will not develop TAI anytime in the rest of this century.**

Again, it is important to emphasize that the distribution above is conditioned on 2020 architectures and algorithms; compute requirements distributions for future years will put more probability mass on low levels of computation due to algorithmic improvement, and may have probability mass on amounts of computation even smaller than ~1e23.