In the long run, only the fundamentals matter

Before I argue for this hypothesis, I'll try to motivate the line of reasoning I'll be taking.

One line of reasoning I've heard recently is that you should "just believe straight lines on a graph", i.e. that empirical progress made consistently in the past will continue. I think this is a bad way of making predictions about AGI.

Here's an anecdote: When I joined OpenAI in 2020, I thought scaling up LLMs might be on the path to AGI. There seemed to be new interesting empirical capabilities emerging from GPT-1 -> GPT-2 -> GPT-3, as well as very predictable scaling laws, so I thought GPT-7 could be something like AGI. But in hindsight this is a very silly line of reasoning -- not only will you run out of data well before GPT-7, but even GPT-1000 would still lack important capabilities due to the fundamental limitations of transformers and autoregressive modeling. You didn't need to follow the scaling laws to the end and spend billions of dollars to realize this (though it's worth doing for other reasons); you could have just reasoned about it from first principles.

In the short run, straight lines always stay straight, but in the long run fundamentals are the only thing that matters.

Position summary

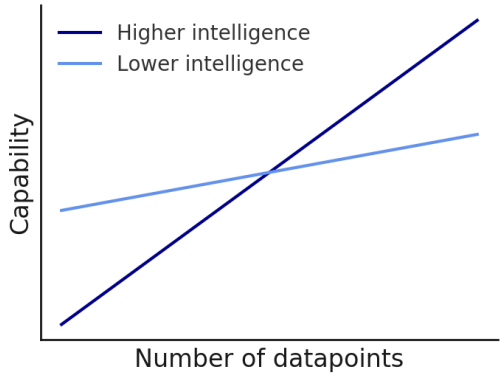
I'll make this argument by considering a number of other AGI hypotheses that prominent ML researchers have had over the years, and accepting them or rejecting them based on the evidence in 2024. This will greatly narrow down the search space for AGI and, I claim, provide support for the "convergent evolution hypothesis".

TODO table

Before we talk about AGI, we first need to define intelligence.

(1) Accept: intelligence is data efficiency, not capabilities

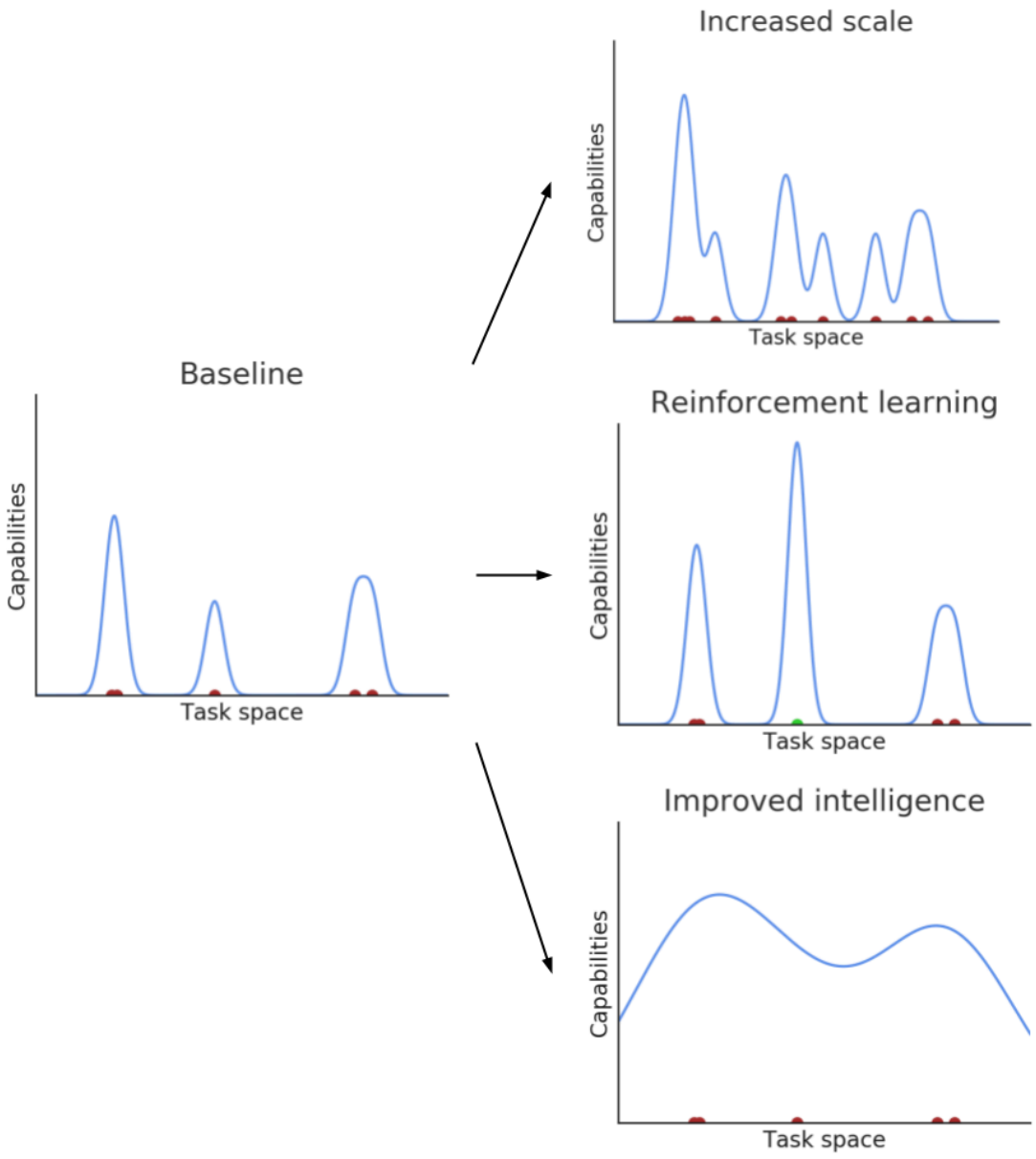
Claim: There is a distinction between *capabilities* and *intelligence*, which are often confused. Capability (or skill) is your performance level on any particular task. Intelligence is the rate at which you improve your capabilities, with respect to how much data you've trained on. On a plot where capability is the y-axis, and amount of data is the x-axis, it is the *slope*, not the y-axis, that determines whether one model is more intelligent than another.



A machine being superhuman on any particular task does not necessarily indicate anything about its intelligence (AlphaGo could be considered a system that is highly capable, but less intelligent than even GPT-2 due to its lack of general learning ability). Intelligence, data efficiency, generalization ability, and learning ability are all the same thing.

In humans, we can find this distinction between intelligence and capabilities in the difference between fluid intelligence (IQ, basically), and crystallized intelligence (skills gained after experience). In deep learning, fluid intelligence corresponds to the training process (DNNs + SGD), and crystalized intelligence corresponds to the trained model (the resulting DNN).

The gaps and brittleness we frequently see with deep learning based methods cannot be fixed with increased scale or reinforcement learning; we need to fundamentally increase the generalization ability of our methods.



Supporters: François Chollet

My view: It's correct.

Rationale: This is really definitional, but it's easy to see why this is the correct definition.

(1.1) The data-limited Chinese room

Consider a thought experiment: let's say the internet had 10^10^10 tokens (as arbitrary a number as the 10^13 on our internet). We could plausibly train an N-gram model on it that could win a gold medal at the IMO and be so useful it would generate $1T a year in revenue. While

such a system is certainly useful and would excel at a wide range of capabilities, it's obviously less intelligent than GPT-4 because it's unable to generalize beyond the data it was trained on.

(Note that this is a *data limited* version of John Searle's "Chinese room" argument. He wasn't totally correct, but had an interesting point, which I'll expand on in section (?))

We can see that generalization ability and data efficiency are the same thing: generalization is the result of squeezing every bit of information out of your datapoints, understanding all correlations and causations, and connecting all the dots. "Squeezing every bit of information" is meant literally: generalization is the result of compression, as I'll describe in section (?).

It's also much easier to understand the "lumpiness" of GPT-4's "intelligence" in this lens. Why can it pass the Bar exam but fail to win at tic-tac-toe? In reality, GPT-4's intelligence isn't lumpy -- it's fixed, and its capabilities are lumpy because its training distribution is lumpy.

(1.2) The spectrum of generalization

Zooming out, I think intelligence / data efficiency / generalization ability fall on a spectrum:

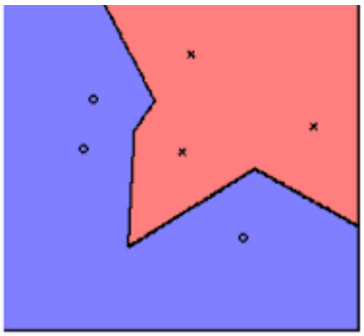
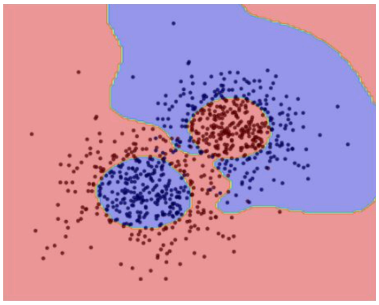
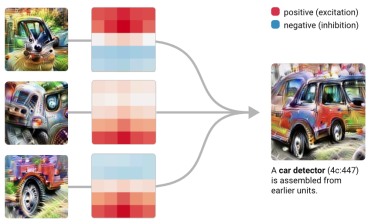
Level 1 = memorization based methods (e.g. N-gram models, nearest-neighbor classifiers)
Level 2 = shallow learners (e.g. word embedding models, SVMs)
Level 3 = deep learners (e.g. LLMs, resnets)

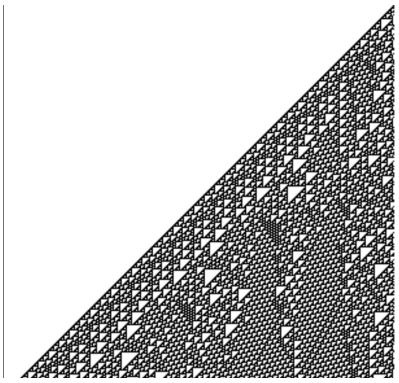
I'll speculate that human-level data efficiency is at "level 4". Solomonoff induction, an uncomputable procedure which has theoretically optimal generalization ability, could be seen as "level infinity". As we go up the levels, a number of things increase together:

- generalization ability
- data efficiency
- computation spent per datapoint
- "unwieldiness" of the method
- emergent complexity from the data
- expressive power

So I'll speculate that the "level 4" paradigm will, compared to deep learning, use significantly more computation per datapoint, be even more "unwieldy", have even more emergent complexity from the data, and even more expressive power -- on the order of how deep learning compares to shallow learning.

Generalization level	Name	Examples	Training FLOPs/token (very approximate)	Visual example of emergent complexity from the data

1	Memorization	N-gram models, nearest-neighbor classifiers	~1	 Decision boundary of a nearest-neighbor classifier.
2	Shallow learning	Word embedding models, SVMs	~10^4	 Decision boundary of a SVM with the RBF kernel.
3	Deep learning	LLMs, ResNets	~10^11	 Emergent feature detectors in Inception-v1.

4	?	The human brain	~10^15	 Speculative: the emergent complexity of “level 4” may look more like the emergent complexity shown in cellular automata than the complexity shown in a fixed feedforward DNN, for reasons I’ll discuss in section (?). The Turing-complete rule 110 shown here.
∞	Optimal generalization	Solomonoff induction	∞	?

I’ve made the claim that humans are more intelligent and have better data efficiency than LLMs in a way that is general and universal. Some would argue against this, so let’s discuss it next.

(2) Reject: human intelligence is specialized and not general

Claim: Humans evolved in a specific evolutionary environment, and we are built to solve problems that were found in this evolutionary environment. We come with many priors “hard-coded” by evolution -- “Core Knowledge” about the physics of our world, basic counting abilities, the ability to reason about agents and goal-directedness, as well as priors about the syntax of language. While we can use and combine these priors in novel ways, we are not general in any universal sense.

It’s also unfair to compare the data efficiency of LLMs to the data efficiency of humans because we come with these priors. These priors give us an initial “leg-up” against LLMs, but do not indicate LLMs are less intelligent.

Supporters: François Chollet, Yann LeCun

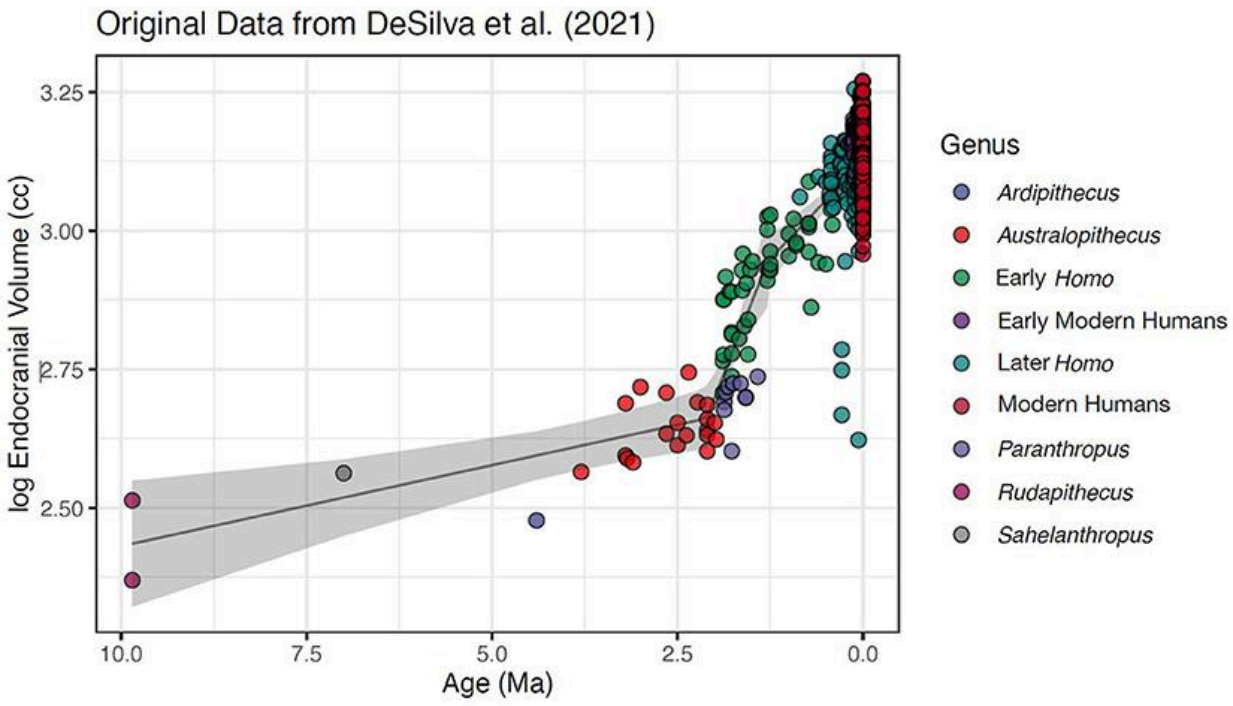
My view: It's incorrect.

Rationale: This is an understandable argument; in many ways it feels like deep learning methods like LLMs, having fairly lax priors, are in an unfair fight against billions of years of evolution. But we must reject it for many reasons:

(2.1) Language is recent

It's often claimed that humans acquire language more quickly than LLMs because we have lots of language-specific priors built into us by evolution. The implicit claim being: while we could in principle hard-code these priors into our LLMs, they are not general and do not indicate an intelligence gap.

But on evolutionary timescales, language is too recent to have substantial amounts of hard-coding dedicated to it (estimates of when language originated vary widely, but are generally between 50,000 to 2 million years ago). Most likely, our language-learning abilities are based on a more general learning algorithm, and the differences between us and our primate ancestors are of low description length (more along the lines of "minor algorithmic tweaks + large amounts of scaling").



(2.2) We've been down this road before

In previous eras of AI, it was debated whether the structure of language could be learned *at all* or whether it was innate and hard-coded (the latter being the camp of Noam Chomsky, which has largely been rejected by the child language acquisition research community). The success of LLMs has now made it seem completely obvious that language *can* be learned, the question is just how efficiently. It seems silly to think that LLMs, with their glaring failures of generalization (like hallucinations and the reversal curse), are as good as it's going to get.

There's a name for this cognitive bias towards thinking the complex behaviors of the mind must be hard-coded instead of learned: the bitter lesson.

(2.3) It's not about language, it's about everything

The claim that "humans are more data-efficient than LLMs" is not just a claim about the syntax of language; it is a claim about *everything*. I don't think anyone would seriously claim LLMs generalize as well as humans, but generalization is exactly the result of data efficiency. It's precisely this generalization ability that makes us special -- why else can we learn programming languages, or the "language" of mathematics, or come up with the theory of relativity, when none of these things were even remotely in our evolutionary environment? General intelligence is real; we are not just a big bag of hacks.

(2.4) Specialization is for insects

It's undeniable that many aspects of our intelligence are specialized and hard-coded by evolution (our ability to recognize faces, our priors about the world being 3D, etc.). But specialization is not particularly interesting (it is, after all, for insects). What is interesting about human intelligence is its generality.

Specialization and generality are fundamentally at odds. A baby horse is typically walking around within an hour, but it takes human babies around a year to do the same. This is the result of the corticalization of motor control: moving motor control from primitive parts of the brain based on hard-coding (like the brainstem) to the newer cerebral cortex (which favors learning from experience). As a result, we have far greater capability to pick up new motor skills, but it comes at the cost of having the longest childhood period amongst all the animal kingdom.

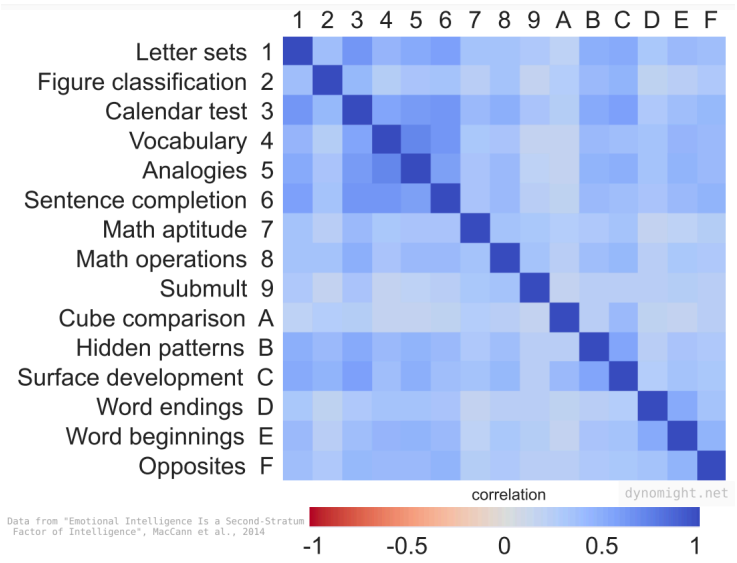
Human intelligence may ultimately come from nature learning the bitter lesson: that while specialized methods can win in the short run, general methods that leverage computation through learning (our big, expensive brains) win in the long run (after our long childhood). It's bitter for a reason: most evolutionary environments won't allow for such a long investment period of childhood or have much incentive for general intelligence, but our evolutionary environment was special.

To the extent we care about having AGI solve problems that are far outside our evolutionary environment (e.g. discovering the cure for cancer or a unified theory of physics), and exceeding

our capabilities in the long run, we should focus on generality and not specialization. In the long run, specialized intelligence is vestigial.

(2.5) Psychometrics indicates that general intelligence is real

TODO



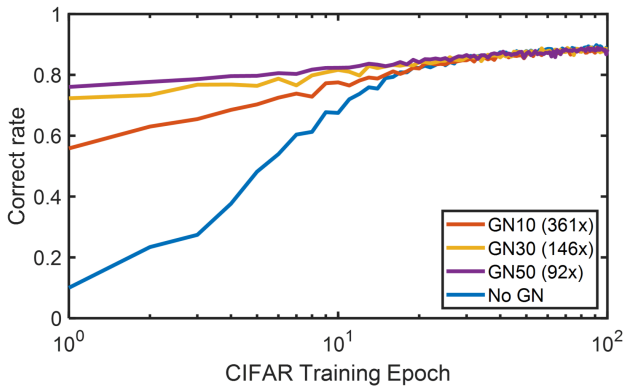
(2.6) General intelligence is possible theoretically

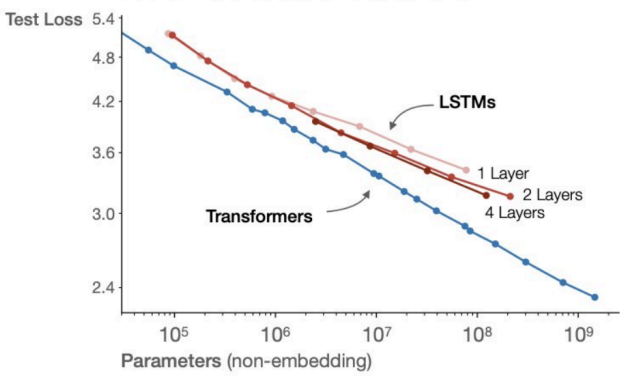
TODO

(2.7) Big blobs of compute win in the end

Implicit in the idea that "humans only learn language quickly because of evolutionary hard-coding" is the assumption that we *could* make LLMs learn language as quickly as humans if we wanted to, but the hard-coding wouldn't be general. Is that really true?

Could the hard-coding be in the weights? Encoding Innate Ability through a Genomic Bottleneck tries this approach: compressing weights by several orders of magnitude, which could conceivably be "hard-coded" in the genome, and then using it as a subsequent initialization. This improves the initial performance with no data substantially, but does nothing for final model performance. Moreover, the maximum 8MB of genomic difference we have to work with would be a compression factor of 22,000 for GPT-3 -- well over the factor of 361 tried in the paper. This isn't plausible.





Could the hard-coding be in the architecture? There was a Cambrian explosion of architectures after AlexNet in 2012 -- including attempting to hard-code linguistic priors into neural networks -- and which eventually culminated in the Neural Architecture Search line of research. What actually ended up working? A giant, relatively homogenous blob of compute, with no linguistic priors, which was not evolved but carefully designed: the Transformer. Even the Transformer is basically a constant factor improvement from LSTMs -- the state of the art from 1997. It's a great constant factor, for sure, but does not really improve data efficiency (the slope). A colleague from OpenAI summed it up best: trained on the same dataset for long enough, pretty much every model with enough weights and training time converges to the same point. Architecture won't help us here.

What does that leave? Learning. Weight priors can give us a constant offset. Architecture can give us a constant multiplier. *Learning* is responsible for the slope. There must be a problem with the combination of DNNs + SGD leading to this lack of data efficiency and poor generalization. How fundamental will this problem be? We should remember this basic formula has gone essentially unchanged since 1986; whatever the problem may be, it's not likely to be an easy fix like "just use more dropout". I'll discuss the fundamental problems more in section (?).

This datapoint should update us twice, that:

(1) There is a fundamental problem DNNs + SGD leading to their poor data efficiency and lack of generalization.

(2) Big blobs of compute win in the end. As long as your architecture is reasonably well suited for your learning algorithm, lots of things work basically the same and task-specific priors add unnecessary complexity. In the case of SGD, the main requirements are not having exploding or vanishing gradients, being numerically stable, and "letting the gradients flow" between different parts of the architecture -- as long as those are true, optimization will take care of the rest.

TODO - the neocortex is also very uniform and "blob like"

An occasional argument for why humans are more intelligent than LLMs is that humans are embodied and take actions in the world, and also have rich multimodal inputs. In 2024, I think we know enough to say this is *not* the reason why humans are more intelligent than LLMs.

(3) Reject: intelligence needs to be embodied or multimodal

Claim: Humans can move and take physical actions in an environment, and this drives our intelligent behavior. So AGI needs to be embodied and take actions in a physical environment.

Also, humans receive far more bits of information through visual input than the total bits of information an LLM is exposed to during training. LLMs can't have human-level intelligence without being exposed to a similar visual stream of information.

Supporters: Yann LeCun

My view: It's incorrect.

Rationale: I think we should be pretty skeptical of this argument *a priori*. While it's often true that humans use our intelligence to take physical actions in our environment, we also use our intelligence for things that are fairly disembodied -- like writing code -- and it seems unclear why motor movement would be the bottleneck for intelligence there.

It seems similarly unclear why visual input or the number of bits of information would matter for a task like writing code. We could certainly turn our text input into visual input by, e.g. using a screenshot of an IDE instead of text tokens, but this doesn't fundamentally change the information content. We could also add noise to our screenshots to increase the information content, but that doesn't make the task any easier.

Empirical evidence seems pretty strong against this hypothesis. Helen Keller was deafblind since the age of 19 months, losing substantial amounts of embodiment and multimodal input during the vast majority of her childhood development, but still turned out perfectly intelligent. Also, most foundation models these days are multimodal, and could easily be hooked up to a robot to make them embodied. It's pretty obvious that this has not made them AGI, and making

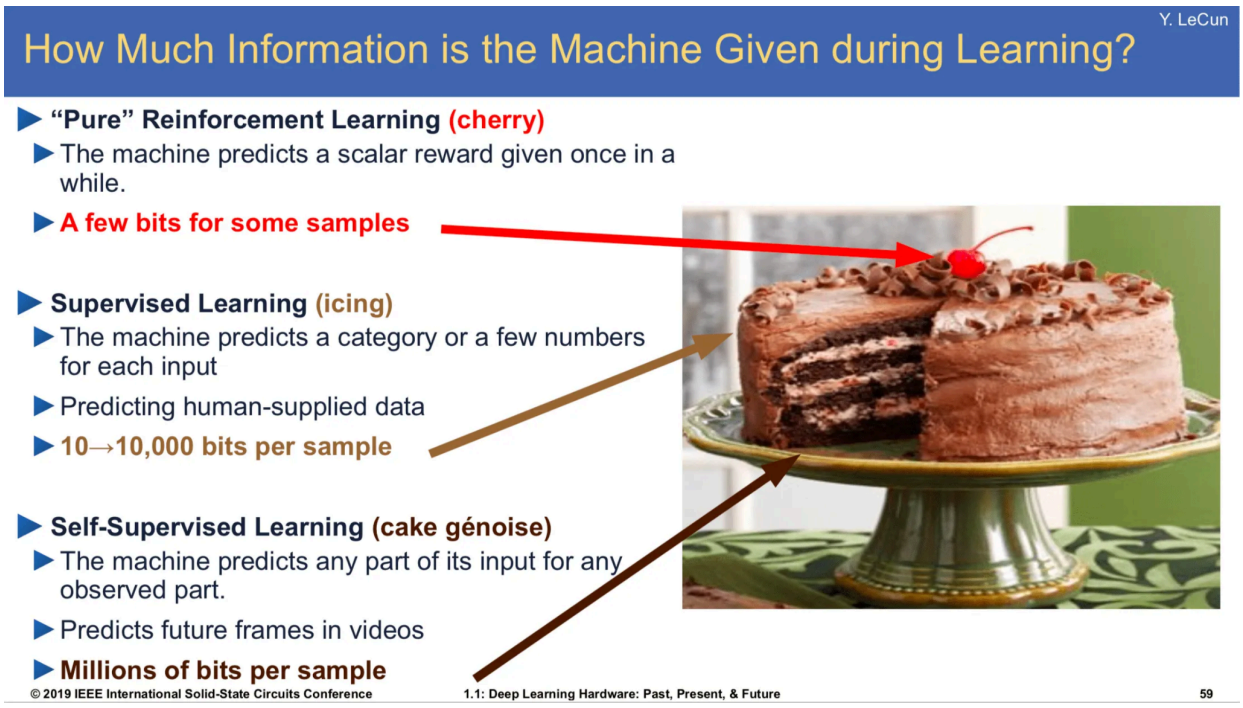
them embodied will just expose their generalization failures instead of turning them into AGI. It also seems clear that text-only foundation models are intelligent to some degree, without having any embodiment or multimodality, which is unexplained by this hypothesis.

I think embodiment and multimodality are best thought as additional surface area for intelligence, not the bottleneck to intelligence itself.

The closest thing we have to AGI today are agents built on top of LLMs, like ChatGPT. I'd argue such systems are a proto-AGI -- they fit the right shape, though they're clearly not yet human-level intelligence. But it's remarkable how similar the process of creating ChatGPT looks to a process of "designing" a brain, and I think this should provide some intuition for what the path to AGI looks like going forward.

(4) Accept: building AGI will be like building a brain

Claim: It's silly to ignore the only example of general intelligence we have -- the human brain -- in creating AGI. While AGI won't be an exact replica of the human brain, we should expect it to follow the same basic principles. In particular, we should expect it to involve a lot of unsupervised learning, some supervised learning, and a little reinforcement learning (Yann LeCun's cake).



We should also expect it to be made entirely of neural networks, and in the case of *general* intelligence, acquire behaviors mostly from learning and experience using a large, relatively homogenous “blob of compute” (analogous to the neocortex).

It should be highly data efficient from birth and efficiently form internal models of the world. It may also involve large amounts of discrete, recurrent computations like those performed by neurons in the brain.

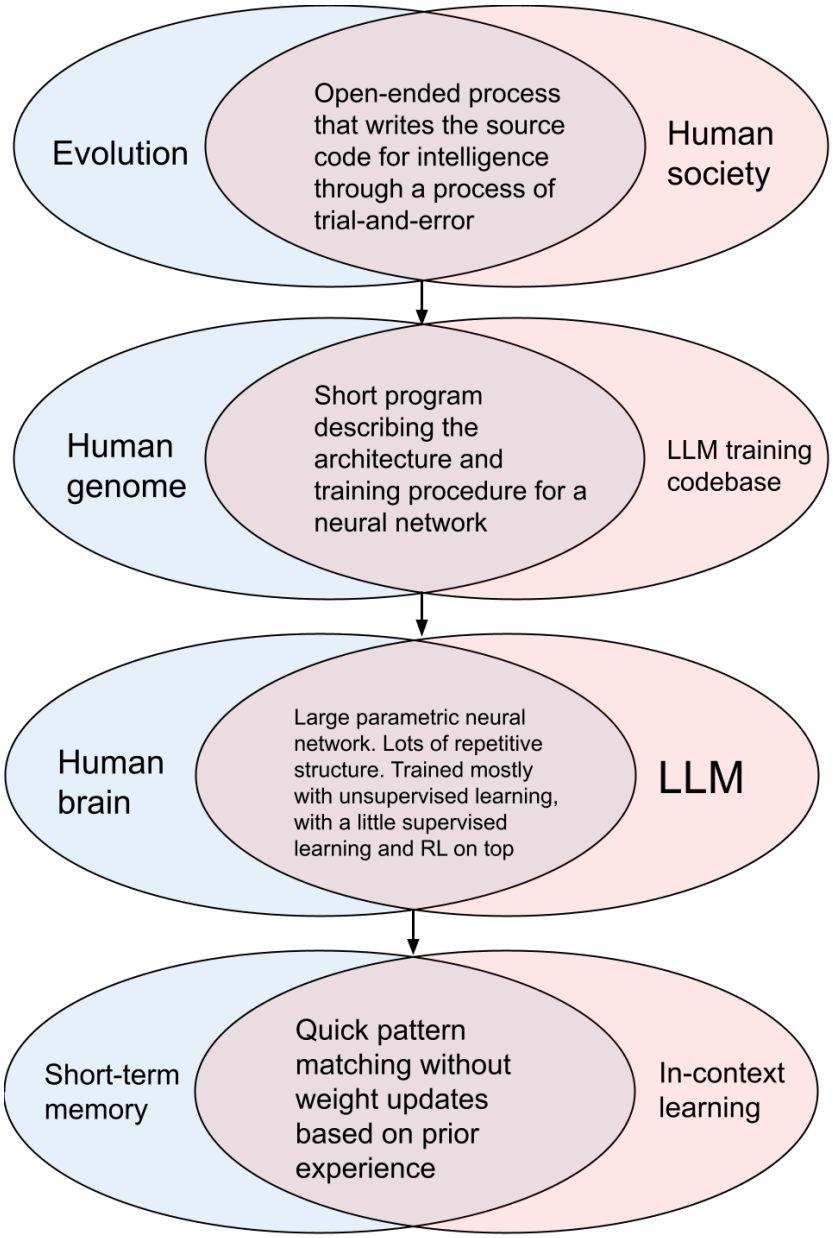
Supporters: DeepMind, Yann LeCun

My view: It’s correct.

Rationale: At first glance, “building AGI will be like building a brain” seems deeply pessimistic, because the brain is extremely complex and it could take centuries of study to fully understand it. But the most intelligent systems we have today (agents built on top of LLMs) are already quite a bit like the brain, and while they are sophisticated, their basic principles are actually pretty simple.

(4.1) LLMs and the brain

We can view the similarities between agents built on top of LLMs and the brain with the following set of Venn diagrams:



Alan Turing first noted the similarities between how evolution constructed the biological brain and how we could construct machine intelligence in his famous paper *Computing Machinery and Intelligence*; since then the similarities have only deepened.

Now we can also begin to reject a common argument justifying the data inefficiency of LLMs: that pretraining is like evolution, and in-context learning is the “real” sample efficient learning. We can reject it twice:

One, pretraining is nothing like evolution. Evolution learns a short program -- the genome -- that describes the learning procedure and architecture for a neural network. The genome is far more analogous to the *training codebase* of an LLM and not the trained LLM itself. The shortness of

the genome is a necessary ingredient for generalization: it's impossible to store the specific words or situations our ancestors were exposed to directly through the genome; evolution is forced to create general learning algorithms because of the information bottleneck.

Two, in-context learning is not general, and depends highly on the training dataset, unlike the general formula of transformers + SGD. Obviously, if our training dataset only contains documents of 10 tokens, we aren't going to see magical generalization to documents of 10,000 tokens. In-context learning is just the model using its context with its normal generalization abilities, and its normal generalization abilities are sub-human precisely because of its sub-human pretraining data efficiency. While it's sometimes thought that in-context learning is a "meta-learning" algorithm with extra generalization power, this is not true, and the fact that the per-token loss decreases in-context does not imply any form of "learning" is happening (a sufficiently large N-gram model will show the same behavior). I'll expand on this in section (6.2), but the correct way to think of in-context learning is that it is quick pattern-matching based on prior experience -- a form of crystallized intelligence, not the fluid intelligence that we're missing.

(4.2) Building the brain

"Like building the brain" is a phrase with a lot of ambiguity. The reality is that the brain has a lot of complexity we don't need to worry about for AGI:

- complexity from vestigial specialized intelligence (e.g. face recognition)
- complexity from evolution optimizing for resource and energy consumption
- emergent complexity from simple learning algorithms applied to complex data

There's a lot of truth in the idea that we don't need to make planes fly like birds -- evolution creates organisms with a lot of complexity, and we can skip most of the complexity in creating intelligent machines. But birds and planes operate on the same basic principles of flight -- lift, weight, drag, and thrust -- and engineers and scientists need to deeply understand these principles before they're able to build an F35. F35s do not "emerge" from a magical optimization process; they're very carefully designed according to these principles, and similarly we should expect to build AGI by deeply understanding the principles of intelligence and carefully designing an intelligent machine.

The principles of intelligence are already well-understood at a high level; they're the three paradigms of machine learning: unsupervised learning, supervised learning, and reinforcement learning. The fundamental problem we have is that our methods are far too data inefficient. We can begin to diagnose the problem by understanding the origins of data efficiency, which I'll discuss later in section (?).

There is definitely an antipattern to avoid here though, which is blindly copying the brain or trying to reverse engineer it. The brain is better thought of as a proof of existence than a template: like birds are a proof of existence that flight is possible, brains are a proof of existence that general intelligence is possible.

We can build further intuition that “building a brain” is the right approach by considering all the other approaches to AGI that have worked considerably less well. One broad class of these approaches is the idea that we should “evolve” a brain instead of building one directly.

(5) Reject: intelligence should emerge from an outer optimization loop

Claim: While the human brain is the only example of general intelligence we have today, it is also hopelessly complex to understand. We’re better off simulating a simpler optimization process, like evolution, and having intelligent agents emerge from that optimization process -- like nature did.

The extreme version of this hypothesis is having AGI emerge from artificial life, while softer versions of this hypothesis include the “scaling hypothesis” and the “reward is enough” hypothesis, which I’ll discuss in sections (6) and (7).

Supporters: OpenAI, Deepmind (partly)

My view: It’s incorrect.

Rationale: This is wrong for many reasons.

One, as established in section (4), the brain being complex does not imply AGI will be complex, and we are already on our way to designing a brain.

Two, evolution operates on the space of short, highly expressive programs -- the genome. Both the shortness and the expressivity of the program are necessary to represent learning algorithms that generalize. We have no idea how to optimize over something as flexible as the space of Python programs. The closest attempts like Neural Architecture Search pale in expressivity and are horrendously expensive. Ironically, our best bet to optimize over the space of Python programs is to have humans write them, and in a process of intelligence-guided trial-and-error, write the source code to AGI. That’s exactly consistent with section (4.2), placing humans in the role of an evolutionary process that designs the brain.

Three, evolution is not really a “simple optimization procedure”, but a deeply open-ended process. It’s often abstracted away that evolution “optimizes fitness”, but if fitness were all evolution was about, bacteria would be the pinnacle of evolution. Evolution produces infinite novelty and complexity, in turn producing self-play environments, curriculum learning, the evolution of evolvability, etc. -- all of which are necessary ingredients for intelligence to emerge. We have no idea how to simulate such an open-ended process, and due to computational irreducibility, there may not be a shortcut other than simulating our entire universe.

The empirical results of the “purist” form of this approach, artificial life, speak for themself (there are no results). Softer versions of this hypothesis that are willing to go further in the brain-like direction (using neural networks and some form of unsupervised or reinforcement learning) show some signs of life, precisely because they’re willing to be more like the brain.

The “scaling hypothesis” is one variant of an “outer optimization loop” procedure to create AGI. It’s often claimed the GPT series is a vindication of the scaling hypothesis, but ironically it’s actually a rejection of it.

(6) Reject: the “scaling hypothesis” / scale is the bottleneck

Claim: Although initially poor at generalization, neural networks can *increase* their generalization ability by meta-learning new algorithms with superior generalization ability, when trained on a large set of sufficiently diverse tasks. So training large neural networks on a broad variety of tasks is sufficient for AGI to emerge from the optimization process.

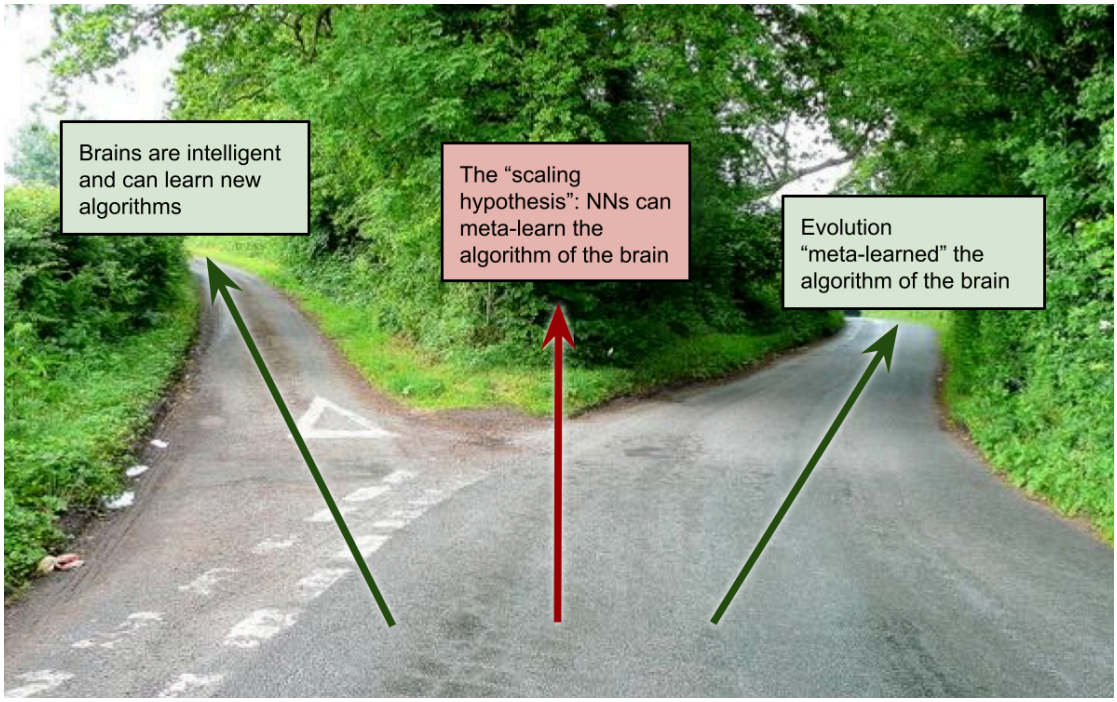
Supporters: OpenAI

My view: It’s incorrect.

Rationale: This is a pretty specific hypothesis, so it’s worth understanding where it came from. It is essentially the mode-averaging of two correct statements:

(1) Brains are sophisticated neural networks and can learn new algorithms when applied to data
(2) Evolution is a simple learning algorithm and “meta-learned” the algorithm of the brain when “optimizing” across a broad spectrum of “tasks”

So maybe if we take our current neural networks, and train them on a broad set of tasks, we’ll also meta-learn the algorithm for AGI?



It almost sounds plausible, but it's wrong for two reasons:

(6.1) Meta-learning a level-up in generalization is impossible

One question should give us immediate pause: obviously not any learning algorithm is sufficient for the "scaling hypothesis" to be true -- we couldn't scale up SVMs and get AGI. Why is our current formula of DNNs + SGD sufficient?

The reality is that it's not. Algorithms of lower level of generalization cannot simulate or meta-learn algorithms of higher levels of generalization. SVMs will never be able to simulate DNNs, no matter how much data you throw at them, for basic fundamental reasons. And for similar fundamental reasons I'll discuss in section (?), DNNs ("level 3") will not be able to simulate human-level generalization ("level 4").

Also, note that the brain does not actually do any "meta-learning" of generalization abilities, it gains *crystallized intelligence* (aka capabilities) by the application of its *fluid intelligence* (aka generalization abilities). While crystallized intelligence grows with experience, fluid intelligence does not. In general, the idea that *learning from experience* can drastically multiply *learning ability* is an infinite recursion that does not hold.

Evolution in some sense did "meta-learn" human-level generalization, but nothing we do is like evolution, as described in section (5). All we're doing is training neural networks.

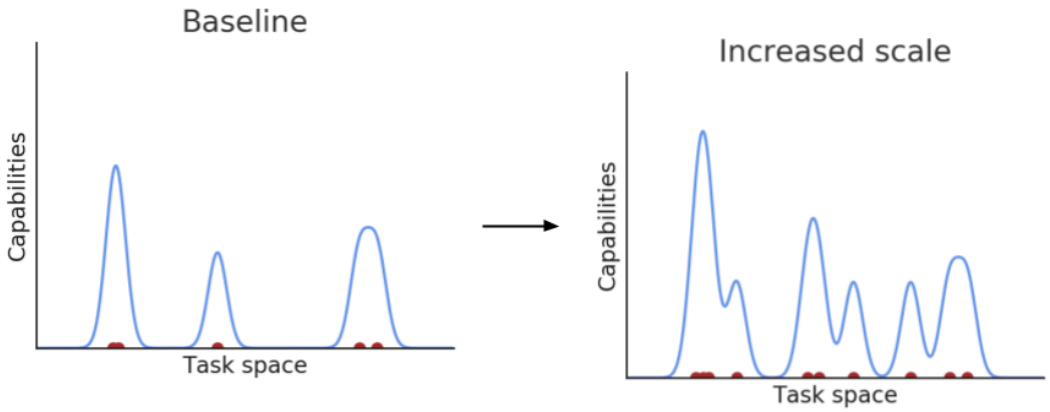
(6.2) Neural networks don't meta-learn

There's another question that should give us pause. The diversity and scale of tasks apparently matters. How much diversity do we need? How much scale? Do we need diversity and scale on the level of evolution, or does a collection of tasks on the internet happen to be perfectly enough to AGI?

The reality is the diversity of tasks only matters insofar as we want to fool ourselves into thinking the breadth of our training distribution is the breadth of our model's intelligence. There is no "meta-learning", there is only *learning*. Evolution is disanalogous to multitask training because there is no separate short-program genome when we train neural networks; there's only the model weights and SGD, which makes no distinction between "meta-learning" and "learning". And even in cases where DNN architectures can technically simulate sophisticated algorithms, there is no reason SGD will find them (technically, an infinite-precision RNN is Turing-complete, but this doesn't matter at all).

(6.3) What actually happens with scale

So what is really happening with the "scaling hypothesis" is much more boring than having "meta-learned algorithms emerge from the data" -- we're just training a neural network with sub-human ("level 3") generalization on a lumpy set of data, and seeing a lumpy set of capabilities result. There's no improvement in generalization.



This is fully born out empirically: AGI did not "emerge" from the training of GPT-4 -- a large DNN trained the largest, most diverse data source we have access to. Instead, what we got was an unreliable lump of capabilities that's human-level where its training data is dense (e.g. academic practice exams) but still fails to win at tic-tac-toe. This is a crappy brain trained on a big pile of data, and not particularly interesting.

(6.4) A caveat

To be fair, there are some *plausible* signs of life for increased generalization at scale:

- Chinchilla scaling laws indicate that an infinite-parameter LLM would be ~9x data efficient compared to an LLM that was trained compute-efficiently
- Transfer learning between datasets sometimes gets better with scale
- Generalization benchmarks like the Abstraction and Reasoning Corpus (ARC) improve with scale

It's debatable whether any of these "truly" signifies increased generalization, but it's not worth debating anyway -- even if it were true, the rate of improvement is glacial, and would never reach human-level generalization for fundamental reasons.

---

(7) Reject: "reward is enough" / RL is the bottleneck

Claim:

(1) All intelligent behavior in an agent is for the purpose of maximizing its reward signal.
(2) Therefore, if we train an agent to maximize its reward signal via reinforcement learning, intelligent behavior will emerge.
(3) Similarly, generalization will come from training in a wide diversity of environments, which will require general intelligence to emerge.

Supporters: OpenAI, DeepMind

My view: It's incorrect.

Rationale: Like the "scaling hypothesis", this hypothesis also comes from the mode-averaging of two correct statements:

(1) Evolution created intelligent agents.
(2) Intelligent agents maximize reward with reinforcement learning.

So if we use reinforcement learning to maximize reward with an agent, maybe it will become intelligent? No, this is just backwards causality, and as discussed in section (6), intelligence won't "emerge" from the optimization of a neural network.

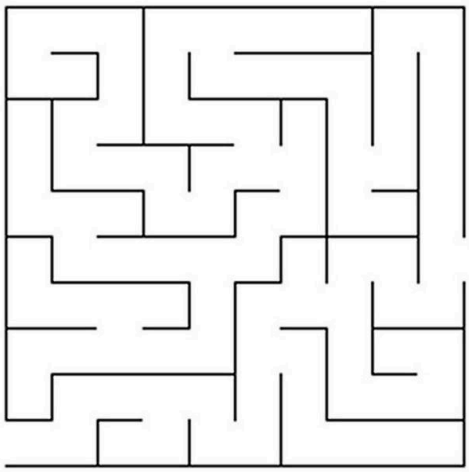
But we can actually make an even stronger statement here: in many cases, reinforcement learning on an objective is a poor way of optimizing the objective itself.

(7.1) The myth of the objective

One intuition commonly held in deep learning is that SGD on a DNN can produce incredible results on a wide variety of objective functions, because local minima in high dimensional

spaces are rare -- it's much more likely you'll end up on a saddle point, and and can go down one end to continually make progress.

This intuition is completely wrong in RL. Here's a simple example: consider an agent in a maze, where we would like to minimize the distance of our agent from the endpoint of the maze. A naive loss function could just be ||x_agent - x_endpoint||.



We could plausibly do gradient descent to minimize this objective, but it will obviously just get stuck in a corner. It doesn't matter if our agent has a 100b parameter neural network to control its position. What's going on here? The fundamental problem is that although we're optimizing in a high-dimensional space, the *intrinsic* dimensionality of that space is low, and still has the original local minima of the maze. Neural networks and SGD won't save us here.

An RL maximalist might describe this as an "exploration problem", but in a maze, exploration is the *entire problem.* It *requires* intelligence to be built in the first place, and won't emerge from getting stuck in a corner.

Note that the claim here is not that we can't get a computer agent to solve a maze. Of course we can, if we as a programmer are willing to program in various heuristics about exploration and backtracking. But this is missing the point: we want agents that can discover like we can, not which contain what we have discovered.

What problems in life are like a maze? Again, the ones that actually require intelligence.

Fermat's last theorem was not solved step-by-step or with simple backtracking search. First posed in 1637, Fermat's last theorem withstood thousands of failed attempts to prove it over literal centuries. In the meantime, entire branches of mathematics sprouted and grew -- not motivated by Fermat's last theorem, but by the intrinsic pursuit of interestingness and novelty. In 1955, mathematicians Goro Shimura and Yutaka Taniyama conjectured a connection between

two previously disconnected branches of mathematics -- elliptic curves and modular forms -- and in 1986 it was shown that this conjecture implied that FLT was true. Andrew Wiles, seeing this, realized his childhood dream and finally proved the theorem true in 1995. It is true that the proof of FLT required some goal-driven behavior at the end, but the vast majority of the process was driven by the pursuit of whatever mathematicians found interesting. Reaching the goal was just the cherry on the cake.

It is this pursuit of novelty that separates us from the rest of the animal kingdom, equally as much as it is our intelligence. Intelligence and the pursuit of novelty go hand in hand: it's impossible to pursue novelty without intelligence, and having intelligence without pursuing novelty is pointless -- it's just walking into the corner of a maze.

(7.2) Formalizing novelty

It's obvious that humans follow their own interests and creativity in a way that's unexplained by the direct maximization of the utilitarian reward that evolution hard-coded into us. How can we formalize this?

What is the principled way of solving a maze with no *a priori* knowledge of mazes? An agent, dropped into a maze, should first wander around a little and bump into the walls. Then it should realize it's in a hallway and proceed all the way down. Then it may take a fork in the road, and arrive at a dead end. Then it may go back and take the other fork in the road, and also arrive at a dead end. Eventually it should realize the general winding structure of the maze that it's in. Only *then* can it begin to more rigorously devise methods of backtracking search, and eventually find its way out of the maze. Planning, search, and "System 2" are also the cherry on the cake -- the cake itself comes from repeatedly forming and breaking an internal model of the world.

Forming an internal model of the world will be done with probabilistic modeling and compression, as I'll discuss in section (?). "Breaking" this model is the part that drives this curiosity-based exploration. As you might expect, Schmidhuber already solved this problem in 1991.

Let p_i(d_{1:j}) be the probability of all the data our agent has seen from t=1 to t=j, using its model of the world formed at t=i. We can define the novelty reward:

$$n_i = \log p_i(d_{1:i-1}) - \log p_{i-1}(d_{1:i-1})$$

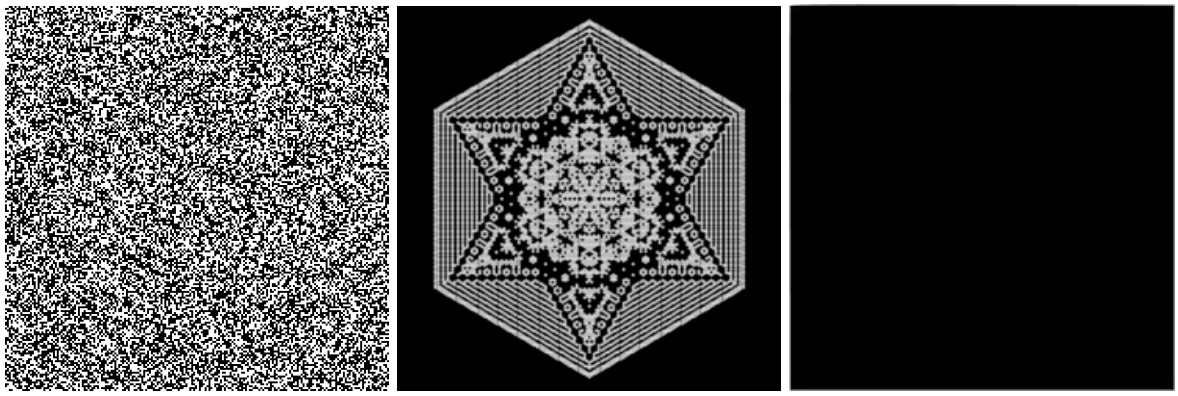
Informally speaking, the novelty reward is the rate of change of our ability to compress our past observations at a particular time. Understanding is compression, so it is the rate of change of our understanding of the world, with respect to time. The total amount of novelty reward will then correspond to the total increase of our understanding of the world over our initial predictions.

Suppose our understanding of the data d_{1:i-1} had converged (i.e. additional time without data would not change p). We observe d_i, and think about it until t=inf. One way of writing the novelty due to observing d_i is:

$$\sum_{j=i+1}^{\infty} n_j = (\log p_\infty(d_i|d_{1:i-1}) - \log p_i(d_i|d_{1:i-1})) + (\log p_\infty(d_{1:i-1}) - \log p_i(d_{1:i-1}))$$

Which is a formalization of the notion that something interesting is some combination of something that:

(1) Initially seems hard to predict, but is something we can later recognize patterns and structure in.
(2) Helps us better connect the dots in our previous understanding of the world.



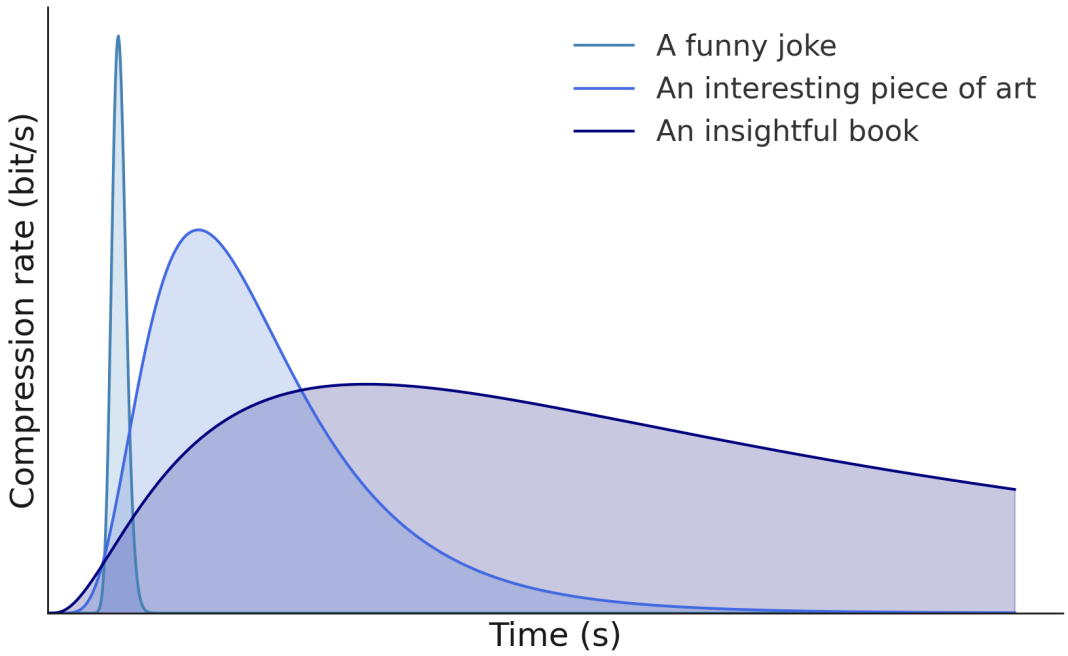
Why is the snowflake-automata in the middle more interesting than static noise or a plain square? Because we can continually find patterns and better compress it with time. Its low Kolmogorov complexity makes it like a bone for us to chew on.

What makes for an interesting book? Often not because of the content itself, but because it helps you reinterpret and better understand experiences in your own life. It's no wonder reading classic novels is so boring in high school; they can only be interesting in the light of experience (but they can at least make later experiences more interesting).

Why are jokes funny? Because when we get to the punchline, we actually *didn't* predict it (if we knew what the punchline was going to be, it wouldn't be funny!). But after hearing the punchline, we quickly connect the dots and have a burst of interest. Note that the novelty reward here actually does require an initial autoregressive prediction. But the deeper *understanding* of the joke requires non-autoregressive compression, which I'll formalize in section (?).

We enjoy art, books, and jokes fundamentally because this novelty reward is hardwired into our dopaminergic pathways through temporal difference learning (a technique first invented by Arthur Samuel in 1959 to make a checkers-playing program; it was later found to be consistent with the reward pathways in the brain -- convergent evolution!).

We can visualize the different types of novelty as follows:



With the novelty reward, we can solve the maze in a general way. We can define the final reward

$$r_i = (1 - \alpha)\, u_i + \alpha\, n_i$$

to interpolate between the agent's utility reward u_{i} (for example, how much closer or further it got from the end of the maze) and its novelty reward.

Simple, straightforward problems can be solved with smaller alpha and less intelligence. But harder problems will require larger alpha and greater intelligence.

(7.3) The origins of reasoning

One of the most confused concepts in machine learning is *reasoning*. There are an infinite number of ways humans reason:

- deductive reasoning
- inductive reasoning
- abductive reasoning
- causal reasoning
- reasoning by thought experiment
- reasoning by analogy
- reasoning by search

Or even stranger forms of reasoning, like Terrence Tao reasoning by writhing on the floor:

“There was one time when I was trying to understand a very complicated geometric transformation in my head involving-- I was rotating a lot of spheres at the same time. And the way I actually ended up visualizing this was actually lying down on the floor, closing my eyes, and rolling around. And I was staying at my aunt's place at the time. And she found me rolling on the floor with my eyes closed. And she asked me what I was doing. And I said, I was thinking about a math problem, and she didn't believe me.”

These are all *crystallized* forms of reasoning; they should not be hardcoded into our agents. Reasoning is the result of an agent pursuing its own novelty rewards -- it is a *learned* behavior. The pursuit of novelty is how we will get agents that can develop new reasoning methods like we can - not merely applying the reasoning methods we have already developed.

(7.4) Psychoanalyzing novelty and understanding

In humans, I think we could say that an individual’s value of α determines whether they will become an engineer or a mathematician; a designer or an artist. That is: to what extent they prefer to maximize their evolutionary utility reward (preferring goal-driven behavior like resource acquisition), or to what extent they maximize their novelty reward (pursuing their own notion of interestingness, usually at the cost of resource acquisition). This is not to say that lower alpha indicates lower intelligence, but that higher alpha offers a higher capacity to express intelligence.

I think we can go further and say there are two modes of understanding:

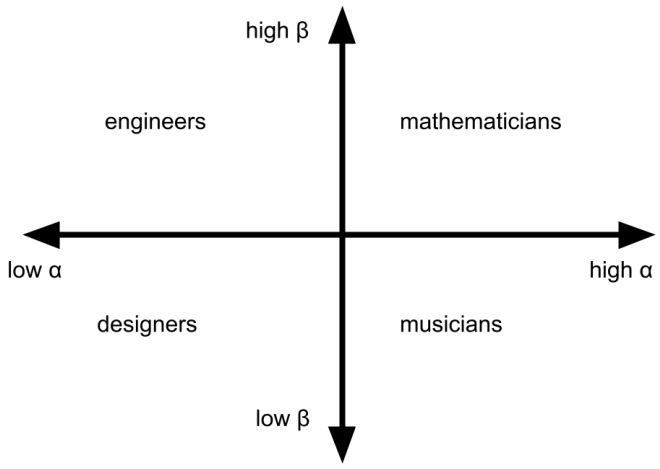
(1) Unconscious understanding (from the “id”) - a nonverbal, intuitive, and instinctive form of understanding, most active when e.g. listening to an interesting song and understanding its patterns and structure. It’s often been said that deep learning is well-suited for this type of intuitive reasoning, and I think that’s true. Deep learning could be seen as the learning of circuits, so this type of understanding seems suited to an analog computer.

(2) Conscious understanding (from the “ego”) - a highly verbal, symbolic, and logical form of understanding, most active e.g. when understanding a mathematical proof and understanding it logically. This type of understanding seems suited to a digital computer.

We could roughly formalize this by saying we have two probabilistic models: p_analogue(x) and p_digital(x) corresponding to these conscious and unconscious modes of understanding. Our final model p(x) will be a product of these two experts:

$$p(x) = p_{analogue}(x)\, p_{digital}(x)/Z$$

and we could imagine a hyperparameter β, where $0 \leq \beta \leq 1$, determining what proportion of computation to spend on digital computing instead of analogue computing. I think it’s interesting to characterize human behavior by an individual’s value of α and β. For example, classifying job occupation:



We could also determine an AI researcher's preference for connectionist or symbolic methods by their personal value of beta (and similarly, to what degree they believe techniques like "chain-of-thought" are related to "thinking").

I don't mean to suggest there are literally two probabilistic models p_analogue(x) and p_digital(x). But beta is a helpful hyperparameter to think about, and I'll formalize it in section (?). The reality is that logical reasoning and intuitive reasoning are inseparably intertwined. My favorite example of this is Ramanujan's "reasoning by having a goddess reveal elliptic integrals to you in a dream":

"While asleep I had an unusual experience. There was a red screen formed by flowing blood as it were. I was observing it. Suddenly a hand began to write on the screen. I became all attention. That hand wrote a number of results in elliptic integrals. They stuck to my mind. As soon as I woke up, I committed them to writing."

It may be tempting to dismiss Ramanujan as a deeply alien form of intelligence that we can ignore. But Ramanujan and the average Joe share 99.9% of their DNA -- it's the same algorithm with a few hyperparameter tweaks. These extreme cases aren't a *different* form of intelligence - they elucidate the way our intelligence really works under the hood. Clearly, some hyperparameters must be highly sensitive; I think beta is among these. I think there is another hyperparameter for that fine line between ignorance, genius, and madness, which I'll formalize further in section (?).