1. Suppose that you have a machine M that lets you sample from the distribution you get when you take any current human, give them any prompt (like, imagine this consisting of a research question with some relevant textbooks etc), and give them time T to produce some output (this could be a paper answering a scientific question given in the prompt, or a report discussing a failed attempt to answer some question, or a better/extended version of a textbook given in the prompt, or whatever else). How large does T need to be for it to be possible to get research progress (e.g. on alignment or on the riemann hypothesis) with a modest/realistic number of calls to M? See this for more on this question: https://docs.google.com/document/d/1IUcbAmYYCA99nZrghWGF5pE5kr5aX_WOz2ORFmf3uHM/edit?usp=sharing
    a. comment by Y: not really sure how to think about "scientific progress" as a privileged concept. a more intuitive story to me is something like "we are doing a search through representation space of models that explain reality, and the tech tree of models we learn is quite sensitive to brain architectures / societal configurations / initialization and there's little a priori reason to expect AIs to have the same difficulties, other than the fact that existing model paradigms seem to be primarily pretrained on human-generated data (which is a reasonably strong prior!)." so I would expect that thinking about the human case tells you very little about the Solomonoff Inductor case, pretty little about GPT-x in the limit, and somewhat more about GPT-6
    b. response by kaarel: one central reason i'm interested in that question is: it's important for understanding when human prediction/emulation (in particular, maybe using christiano-style research decomposition) would start to work. eg if you think you can chain 1-minute terry taos into a riemann hypothesis prover, then one should be more optimistic about getting human-like research from predictive models in time (to take over the world before anyone else does (or for whatever else one wants research for))
2. How do you actually do human prediction/imitation/simulation/emulation/uploading (maybe with ML), well enough to get research progress out? You have a structure out there leaking data about itself various ways — what's so hard about copying it onto silicon? (Why doesn't a pretrained LLM already work?) Make progress on building a framework for understanding when predictive modeling works.
    a. comment by Y: i think there are two questions here:
        i. what are the fundamental limits to brain emulation with current technology
        ii. what predictive model architectures are capable of making research progress
    b. response by kaarel: i agree these are fine questions. i think there are many more questions. the theory im asking for might eg be able to say something interesting about the following:
        i. suppose i make a computer inside game of life that does sth. eg factors an integer
        ii. suppose i have a data set of frames from this game of life world, but they are somewhat blurred/coarse-grained (eg averaged over some spacetime boxes)
        iii. now i use ML techniques to make a predictor of next frames from previous frames (eg like https://gamengen.github.io/)
        iv. when (like: at what resolution, for how much ML-oomph) does this give me something that still computes well?

        this is supposed to be somewhat analogous to trying to make a "human mind upload/emulator/predictor" by predicting blurry brain scans (or even just predicting outputs like pretrained LLMs do). like, in a human, you have a fine structure with simple transitions (at the level of fundamental physics, or maybe at the level of

neurons/synapses, or whatever) on which some high-level "algorithm"/behavior of interest (proving the riemann hypothesis) is implemented. you can take blurry readings of the state and predict next blurry states from previous ones (using ML methods). when does this work well enough (ie get you a RH prover)?

a somewhat different toy case:

* you have a deep boolean circuit (here, the circuit is analogous to a human researcher)

* you can try to "model steal" it with ML by training on its input output pairs

* this will suck (related: the XOR learning literature)

* but! if you give intermediate states like every 3 layers of the circuit and use ML to predict each next intermediate state from the previous one, then you can win

we could also talk about a probabilistic turing machine (here, the turing machine is analogous to a human researcher) having its state predicted one computational step at a time vs having its state predicted over 10^6 time steps. i think ML can handle the former, but not the latter

the theme: making stuff shallow lets ML win

3. Could one get scientific progress from a Solomonoff inductor? For example:
    a. Suppose we formalize all mathematical sentences we have proven/disproven so far, and we also formalize their proofs/disproofs. Then, feed these to a solomonoff inductor, like statement 1 pf/dispf 1 statement 2 pf/dispf 2 ... statement n pf/dispf n, and then append the riemann hypothesis (as statement n+1). And now we have the inductor generate the next m tokens, either by sampling or by taking the outputs of the currently most probable algorithm (if there is a tie for the most probable one, just pick the lexicographically first one or whatever; also, ignore any which wouldn't generate at least m tokens).
    b. question: Do we get a proof (or disproof) of the riemann hypothesis, or do we get sth else (e.g. some nonsense)? (If we look at the output, do we get hacked when looking at it?) or maybe: At roughly how much input data (or for which values of other hyperparams) will we first get a (dis)proof of the riemann hypothesis?
    c. if the answer is that one doesn't get a (dis)proof of the riemann hypothesis here: Can one write down a different (uncomputable) "inductor" that does better than solomonoff induction? (Of course, it shouldn't be one that is fine-tuned to this case in particular — the point is to try to find one that would also work when we try an analogous thing with scientific questions and survey papers answering them in place of mathematical statements and (dis)proofs.)
4. Could you use verification to get pivotal tech from a misaligned AGI/ASI? For example, could you get it to make you a digital copy of you that can run at 100x speed? I think an interesting case to discuss here is the option where you promise to pay the AI for its work (like, you promise to give it a galaxy at some point if you think it did well). relatedly: Can one characterize the problems such that we already understand how to verify solutions to them? Is it basically just math (and stuff that is obviously equivalent to math)? Can we make progress on figuring out protocols for verifying other types of problems? For example, can we build a formal system in which one can write physics olympiad problem solutions such that answers become much easier to verify than to generate? See this for more thoughts.
5. Would a math AGI want to eat the Sun? (see this for elaboration and discussion)
6. Specify a program/system that gives a 1000x clock speed mind upload device (that doesn't also do something bad) (or a particular mind upload, or a paper explaining how to make mind uploads, or analogous things for curing cancer or for other hard scientifico-technologico-philosophical

problems) given unbounded compute, access to arbitrarily much data of the kinds that "already exist" or are reasonably easy per datum to gather, and ability to use a vast quantity of current humans to do fairly short monitoring tasks (one could see this as falling under "ability to gather data" also). Like, existing papers are completely fine to use as data, doing some easy reformatting/labeling of existing data is fine, having humans spend a few hours generating or evaluating stuff is fine, etc.. But having humanity think 1000 years and make mind uploads and then training a model which memorizes that and "can now make mind uploads" is not fine. (Questions 1–5 are all specific things one could encounter when thinking about this question.) Let's make the further simplification that you're granted a completely secure/crisp server, ie one that only interacts with the outside world by receiving whatever inputs/data/code you give it, and acting on the external world only via the output channels you intend it to have (tho note that if you're monitoring it, then you might need to worry about whatever system you have monitoring it potentially getting used by it as an output channel). The point here is to specify a plan that clearly works (as opposed to a thing you might merely assign >50% chance of working after thinking about it for a few hours but still feel really confused about) — you should make a compelling case that your thing works.

7. Come up with some toy case(s) of language modeling that one can fruitfully analyze theoretically (this is sort of a conceptual LLM interp project). Maybe something like: you have a vector for each composite phrase which is computed by some parametrized operation from vectors for its two subphrases (https://en.wikipedia.org/wiki/Merge_(linguistics)), and you also choose parameters giving vectors for "atomic phrases" (ie words or morphemes); you choose all the parameters to optimize for the truth value of the sentence being linearly readable from its vector representation. You could do this eg for natural language or for formal math. Maybe if one makes the right choices of hyperparams, one can figure out some stuff about what doing this well looks like. See this for more thoughts.