An unaligned benchmark

Paul Christiano Follow
Mar 20, 2018 · 11 min read

My goal is to design AI systems that are aligned with human interests and competitive with unaligned AI.

I find it useful to have a particular AI algorithm in mind. Then I can think about how that algorithm could cause trouble, and try to find a safer variant.

I think of the possibly-unaligned AIs as a benchmark: it's what AI alignment researchers need to compete with. The further we fall short of the benchmark, the stronger the competitive pressures will be for everyone to give up on aligned AI and take their chances.

I have a few standard benchmarks I keep in mind. This post describes one of those benchmarks. It also tries to lay out clearly why I think that benchmark is unsafe, and explains how I think my current research could make a safe version.

I. Model-based RL with MCTS

We train three systems in parallel:

- A generative model to sample sequences of observations, conditioned on sequences of actions.
- A reward function that takes as input a sequence of actions and predicted observations and produces a reward.
- A policy and value function which take as input a sequence of observations and produce the next action and an estimate of the future return.

We train the policy and value function using (roughly) the AlphaZero algorithm: Use MCTS to improve the current policy. Update the policy at the root to predict the best move found by MCTS, update the value to predict its predicted value. Use the generative model to sample environment transitions and the reward function (with a small discount rate) to score them.

We train an autoregressive generative model, to maximize the log probability assigned to the actual sequence of actions and observations produced by the AI (with each observation conditioned on the past actions). This isn't actually a good way to train the generative model, but it's not really central to the discussion.

We train the reward function by showing humans sequences of actions and predicted observations, asking them to assign scores, then predicting those scores with supervised learning. We show humans the sequences of actions that look most promising to the system.

There are plenty of details you'd need in order to make this work well, but that's the basic idea. When applied with very powerful networks, it's plausible that this system would be able to decisively outcompete humans. It would be capable performing a large intelligent search over long sequences of actions to find those that would be rated highly.

II. What goes wrong?

There are two classes of problems:

Problem 1: Bad objective

The goal of the system is to produce (action, observation) sequences that look good to humans. I claim that optimizing this objective faithfully will lead to bad outcomes.

As the system improves, the rationale of many individual actions will become incomprehensible to a human overseer. At this point the only option for a human is to evaluate sequence of observations based on whether the consequences look good.

The observations present a narrow view of the world, and I strongly suspect that the AI will find sequences of actions that make that narrow view look good without actually *being* good.

Control vs. intrinsic goodness. I think there are two strategies for defining a reward function:

1. Reward worlds in which *humans remain in control of the situation*, in which they are able to get accurate information and correct course as needed.

2. Reward worlds in which *intrinsically good things are happening*

Both of these strategies seem unworkable.

Strategy #1: maintaining control. This appears to be unworkable because determining if humans are actually in control is incredibly difficult—at best you can tell whether they appear to be and feel in control. It's very hard to understand if the humans are getting accurate information, if their understanding of the situation is roughly accurate, if their instructions are being faithfully executed, and so on. This is already an incredibly hard problem amongst humans and gets way harder as the AI gets smarter.

This problem is serious because, if there is even a small gap between the "real" notion of being in control (which an AI cares about in order to generate maximally appealing observation sequences further into the future) and a superficial appearance of control (which the reward function incentivizes), then the AI is going to exploit that gap by taking the real notion of control for itself and giving humans the appearance of control. Of course, if the long-term goals of the AI and human are well-aligned then there would be no conflict of interest, but trying to ensure that alignment brings us back to strategy #2 which I will argue is unworkable.

You could potentially notice this problem by looking ahead to the point where humans attempt to exercise control—if the humans don't get what they want, then they must not have really been in control. But evaluating whether they get what they want brings us back to strategy #2.

Strategy #2: evaluating goodness. This appears to be unworkable because it requires evaluating human preferences over very large objects (the whole world) under very complex conditions (in the far future, where we've had to make many decisions about what kind of world we want to have).

I am quite confident that "world which looks appealing on screen" is not the same as "world which is good;" the gap is already large today, and it will grow much larger over time.

Maintaining a fiction. Ultimately the goal of this AI is to make the sensors look like good things are happening in the world. This can be done either by making the world good or by carefully controlling what gets fed to the sensors. In general, there are many fewer constraints on “what gets fed to the sensors” than “what the world is actually like,” and so it’s going to be possible to produce much more desirable-looking outcomes, much more easily, by controlling the sensors.

If the humans understand what the AI is doing in detail then they can observe the sequence of actions to notice any actions that might corrupt the sensors. But as soon as the AI is engaging in complicated plans this appears to be totally intractable. Even if the AI was human-level, there are many extremely hard-to-detect plans: for example, the AI could collaborate with an external attacker, who will surreptitiously corrupt the sensor. Or the AI could create a copy of itself on an external server who then helped orchestrate a hard-to-detect corruption.

These actions would be very hard to detect even if you were monitoring a human-level intelligence, and the situation seems hopeless for much more powerful searches. Moreover, because you are searching over sequences of actions for those that receive high reward, you are directly applying optimization power to discovering covert attacks.

Other problems. I have a (fairly common) intuition that applying extremely powerful optimization at a target that isn’t *quite* what you want will often lead to bad outcomes. The discussion above is not exhaustive, but I think it is illustrative.

Problem 2: distributional shift (and optimization daemons)

Our training procedure produces a policy and value function, most likely represented as (really big) neural networks. At test time, we combine these the policy and value with MCTS to decide on actions.

The value function and policy have been optimized to yield good performance on the data points we’ve seen so far, as judged by human evaluations. Unfortunately, there are likely to be a very large number of networks that encode the “wrong” goals but which also yield good performance. These networks will generalize poorly, and moreover when they fail to generalize they can result in an extremely powerful optimization process being pointed at the wrong objective.

A story about training. Originally the policy and value function don’t encode anything at all. Over time, they begin to encode a complicated soup of heuristics which is correlated with good performance. If we are training sufficiently powerful models we hope they will eventually perform reasoning about outcomes. For example, the policy could learn to backwards chain from heuristics about what is valuable in order to decide which moves are good. This is what we are trying to do—the policy is *supposed* to backwards chain, it's the only part of the system that can use heuristics in order to prioritize the search.

What humans actually want is somewhat complicated, so it seems quite likely that it's easier for models to pursue a complicated soup of heuristic goals than to understand exactly what we want. This is similar to the way in which humans acquired an extremely rich set of goals even though we were optimized according to evolutionary fitness. This is a complicated question, but I think it's the theoretical picture and I think historical experience with deep learning points tends to support it.

As the system improves, the reward function encourages it to exhibit an increasingly precise understanding of what we want. Unfortunately there are two ways to do this:

- The intended way: adjust the implicit goals baked into the model such that they converge towards "be helpful to humans." In the analogy to humans, this is like humans caring more and more about reproductive fitness (and less and less about things like beauty or fun except insofar as they are useful for reproductive fitness).
- The unintended way: correctly understand that earning human approval is necessary to survival and hence to achieving other goals, and act accordingly. In the analogy to humans, this is like humans continuing to care about beauty and fun, but believing that they need to have kids in order to realize those goals in the long run.

In practice, I expect both of these changes to occur to some extent, ending up with a model that has somewhat wrong goals together with an instrumental desire to appear helpful.

Catastrophic failure. This could lead to a catastrophic failure in a few different ways:

- An attacker deliberately produces inputs that drive our AI off of the training distribution, and it starts pursuing the wrong goals. That AI may then launch a similar attack against other AI systems it has access to, leading to cascading failures (as with a computer virus). Or an attacker may be able to simultaneously compromise a large number of systems.

- As AI systems acquire increasing influence in the world, they necessarily move off the training distribution. Eventually this sparks a failure in some systems. These failures could cause chaos in the world, pushing us further from the training distribution and leading to cascading failures; or they may all be triggered by the same events and so be correlated.

In either case, we could end up with a massive correlated failure of AI systems, where they start effectively maximizing the wrong goals. That looks effectively like a conflict between us and the AI systems we've built (just as a virus might effectively lead to a conflict between you and the computer you bought). If the AI systems either have significant responsibilities, or are much more intelligent than unaided humans, then there may not be any way to recover from this failure.

Problem 1.5: non-robust reward functions

There is another risk at the intersection between robustness and value specification.

We may learn a model of human approval which is accurate on the training distribution, but incorrectly assigns a very high value to some bad outcomes that didn't appear in training. Indeed, recent experience with adversarial examples suggests that our models often have very strange behavior on parts of the input space not visited in training and that this problem can be hard to correct. Presumably some of these inputs would be assigned unusually high values (just as some would be assigned unusually low values).

In order to reach the most pathological cases, the agent needs significant control over its own observations, which in in turn requires control over its environment. So even if the pathological inputs aren't any worse than random on average, the ones that the system is able to reach will systematically be those where it has acquired enough influence to control its own observations.

We can potentially correct this problem by eliciting human feedback for the problematic observations and then adding them to the training set. We're only in serious trouble if the reward function is really hard to fix. In that case, we may instead make it harder and harder for the system to modify its own observations. After a while, we might train a system which is only able to control its observations by physically overpowering us and causing a catastrophe.

III. How the problem might be fixed

I think that my research has a chance of producing an AI that (a) is nearly-as-good as the benchmark, but (b) doesn't do anything terrible.

The main changes are:

- Rather than learning the environment model directly from observation, produce it by amplification. The amplification process can use exactly the same data about the world (treating the data as an external resource), and it can ultimately be distilled into a neural network just like the one you would have trained. The important difference is that we now have a way to link the generative model back to (an astronomically large number of) human-comprehensible operations on human-comprehensible data and a sequence of decoders which can map from the latent space of the generative model to an implicit human-comprehensible representation.
- Rather than learning a reward function from human data, we also train it by amplification (acting on the same representations used by the generative model). Again, we can distill the reward function into a neural network that acts on sequences of observations, but now instead of learning to predict human judgments it's predicting a very large implicit deliberation.
- In addition to training the policy and value function using RL, we use some combination of these techniques to ensure there is *no* series of observations on which the policy behaves extremely badly. For example, we may use amplification in order to understand what the policy is doing, then use that understanding in order to identify possible (pseudo-)inputs on which the policy behaves badly, then add those (pseudo-)inputs to the training set.

Competitiveness depends on three properties:

- Training a predictor using amplification makes predictions at least as good as training a generative model directly.
- The worst-case training for the policy and value function don't interfere significantly with average case performance.
- Amplification doesn't add much overhead to the training process.

Safety is more subtle. It depends on three properties each defined in terms of some as-yet-undefined notion of "bad" behavior (incorrigible is the current leading candidate):

1. On average, the reward function incentivizes behavior which competes effectively and gives the user effective control.
2. There do not exist inputs on which the policy choose an action because it is bad, or the value function outputs a high reward because the prior behavior was bad. (Or else the policy on its own will generate bad consequences.)
3. The reward function never gives a behavior a higher reward because it is bad. (Or else the test-time optimization by MCTS can generate bad behavior.) For example, if the AI deludes the human operator so that the operator can't interfere with the AI's behavior, that behavior can't receive a higher reward even if it ultimately allows the AI to make more money.

My goals

In order to make all of that work, we'd need to solve a few research problems.

Reliability. Some combination of these techniques needs to successfully eliminate all bad behavior (and in particular to control optimization daemons).

Amplification. Amplification needs to be good enough for these three tasks:

- Learn models that make competitive predictions about the world, and which are "understandable" enough that we can…
- Implement a reward function, defined on those models, that correctly rewards competitive behavior and penalizes bad behavior.

- Understand the internals of competitive machine learning systems, well enough to do informed oversight and to help with reliability (if interpretability is an important ingredient for reliability). Probably the key requirement is reliably identifying bad behavior.

(Without introducing significant overhead.)

Understanding bad behavior. In order to do either of the above we need some suitable notion of "bad" behavior, such that:

- Avoiding bad behavior is sufficient to avoid catastrophic outcomes.
- Bad behavior is an act of commission that is "easy" to avoid, such that it can be achieved in the worst case.
- We can learn a reward function over that avoids creating instrumental incentives for bad behavior, e.g. by punishing any bad behavior which played an important role in receiving a high reward. (This is only plausible because our reward function operates on sequences of predicted states, and so if bad behavior is instrumentally useful it must be because the model "knows about" it.)