Predictions & Self-awareness

1. The Dualist Predict-O-Matic ($100 prize)
2. Self-Fulfilling Prophecies Aren't Always About Self-Awareness

The Dualist Predict-O-Matic ($100 prize)

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

*This is a response to Abram's The Parable of Predict-O-Matic, but you probably don't need to read Abram's post to understand mine. While writing this, I thought of a way in which I think things **could** wrong with dualist Predict-O-Matic, which I plan to post in about a week. I'm offering a $100 prize to the first commenter who's able to explain how things might go wrong in a sufficiently crisp way before I make my follow-up post.*

Dualism

Currently, machine learning algorithms are essentially "Cartesian dualists" when it comes to themselves and their environment. (Not a philosophy major -- let me know if I'm using that term incorrectly. But what I mean to say is...) If I give a machine learning algorithm some data about itself as training data, there's no self-awareness there--it just chugs along looking for patterns like it would for any other problem. I think it's a reasonable guess that our algorithms will continue to have this "no self-awareness" property as they become more and more advanced. At the very least, this "business as usual" scenario seems worth analyzing in depth.

If dualism holds for Abram's prediction AI, the "Predict-O-Matic", its world model may happen to include this thing called the Predict-O-Matic which seems to make accurate predictions -- but it's not special in any way and isn't being modeled any differently than anything else in the world. Again, I think this is a pretty reasonable guess for the Predict-O-Matic's *default* behavior. I suspect other behavior would require special code which attempts to pinpoint the Predict-O-Matic in its own world model and give it special treatment (an "ego").

Let's suppose the Predict-O-Matic follows a "recursive uncertainty decomposition" strategy for making predictions about the world. It models the entire world in rather fuzzy resolution, mostly to know what's important for any given prediction. If some aspect of the world appears especially relevant to a prediction it's trying to make, it "zooms in" and tries to model that thing in higher resolution. And if some part of *that* thing seems especially relevant, it zooms in further on *that* part. Etc.

Now suppose the Predict-O-Matic is trying to make a prediction, and its "recursive uncertainty decomposition" algorithms say the next prediction made by this Predict-O-Matic thing which happens to occupy its world model appears especially relevant! What then?

At this point, the Predict-O-Matic has stepped into a hall of mirrors. To predict the next prediction made by the Predict-O-Matic in its world model, the Predict-O-Matic needs to run an internal simulation of that Predict-O-Matic. But as it runs that simulation, it finds that simulation kicking off another Predict-O-Matic simulation in the *simulated* Predict-O-Matic's world model! Etc, etc.

So if the Predict-O-Matic is implemented naively, the result could just be an infinite recurse. Not useful, but not necessarily dangerous either.

Let's suppose the Predict-O-Matic has a non-naive implementation and something prevents this infinite recurse. For example, there's a monitor process that notices when a model is eating up a lot of computation without delivering useful results, and replaces that model with one which is lower-resolution. Or maybe the Predict-O-Matic *does* have a naive implementation, but it doesn't have enough data about itself to model itself in much detail, so it ends up using a low-resolution model.

One possibility is that it's able to find a useful outside view model such as "the Predict-O-Matic has a history of making negative self-fulfilling prophecies". This could lead to the Predict-O-Matic making a negative prophecy ("the Predict-O-Matic will continue to make negative prophecies which result in terrible outcomes"), but this prophecy wouldn't be selected for being self-fulfilling. And we might usefully ask the Predict-O-Matic whether the terrible self-fulfilling prophecies will continue conditional on us taking Action A.

Answering a Question by Having the Answer

If you aren't already convinced, here's another explanation for why I don't think the Predict-O-Matic will make self-fulfilling prophecies by default.

In Abram's story, the engineer says: "The answer to a question isn't really separate from the expected observation. So 'probability of observation depending on that prediction' would translate to 'probability of an event given that event', which just has to be one."

In other words, if the Predict-O-Matic knows it will predict P = A, it assigns probability 1 to the proposition that it will predict P = A.

I contend that Predict-O-Matic doesn't know it will predict P = A at the relevant time. It would require time travel -- to know whether it will predict P = A, it will have to have made a prediction already, and but it's still formulating its prediction as it thinks about what it will predict.

More details: Let's taboo "Predict-O-Matic" and instead talk about a "predictive model" and "input data". The trick is to avoid including the output of the predictive model in the model's input data. This isn't possible the first time we make a prediction because it would require time travel -- so as a practical matter, we don't want to re-run the model a second time with the prediction from its first run included in the input data. Let's say the dataset is kept completely static during prediction. (I offer no guarantees in the case where observational data about the model's prediction process is being used to inform the model *while* it makes a prediction!)

To clarify further, let's consider a non-Predict-O-Matic scenario where issues *do* crop up. Suppose I'm a big shot stock analyst. I think Acme Corp's stock is overvalued and will continue to be overvalued in one month's time. But before announcing my prediction, I do a sanity check. I notice that if I announce my opinion, that could cause investors to dump Acme, and Acme will likely no longer be overvalued in one month's time. So I veto the column I was planning to write on Acme, and instead search for a

column `c` I can write such that `c` is a fixed point for the world `w` -- `w(c) = c` -- even when the world is given my column as an input, what I predicted in my column still comes true.

Note again that the sanity check which leads to a search for a fixed point doesn't happen by default -- it requires some extra functionality, beyond what's required for naive prediction, to implement. The Predict-O-Matic doesn't care about looking bad, and there's nothing contradictory about it predicting that it won't make the very prediction it makes, or something like that. Predictive model, meet input data. That's what it does.

Open Questions

This is a section for half-baked thoughts that could grow into a counterargument for what I wrote above.

- What's going on when you try to model yourself thinking about the answer to this question? (Why is this question so hard to think about? Maybe my brain has a mechanism to prevent infinite recurse? Tangent: I wonder if this is evidence of a mechanism that tries to prevent my brain from making premature predictions by observing data about my own predictive process while trying to make predictions? Otherwise maybe there could be a cascade of updates where noticing that I've become more sure of something makes me even more sure of it, etc. By the way, I think maybe humans do this on a group level, and this accounts for intellectual fashions.) Anyway, I think it's important to understand if my brain does something "in practice" which differs from what I've outlined here, some kind of method for collapsing recursion that a sufficiently advanced Predict-O-Matic might use.

- What if the Predict-O-Matic assigns some credence to the idea that it's "agentic" in nature? Then what if the Predict-O-Matic assigns some credence to the idea that the simulated version of itself could assign credence to the idea that it's in a simulation? (I think this is just a classic daemon but maybe it differs in important ways?)

- In ML, the predictive model isn't trying to maximize its own accuracy--that's what the training algorithm tries to do. The predictive model doesn't seem like an optimizer even in the "mathematical optimization" sense of the world optimization (is "mesa-optimizer" an appropriate term? In this case, I think we're *glad* it's a mesa-optimizer?) What if the Predict-O-Matic sometimes runs a training algorithm to update its model? How does that change things?

- What if time travel actually is possible and we just haven't discovered it yet?

- Does anything interesting happen if the Predict-O-Matic becomes aware of the concept of self-awareness?

- Could the Predict-O-Matic notice and exploit shared computational structure in the recursive self-simulation process? Suppose it notes the computation it has to do to make a prediction for the simulated Predict-O-Matic is similar to the computation it has to do to just make a prediction! What then?

Prize Details

Again, $100 prize for the first comment which crisply explains something that could wrong with dualist Predict-O-Matic. Contest ends when I publish my follow-up -- probably next Wednesday the 23rd. I do have at least one answer in mind but I'm hoping you'll come up with something I haven't thought of. However, if I'm not convinced your thing would be a problem, I can't promise you a prize. No comment on whether "Open Questions" are related to the answer I have in mind.

EDIT: Followup here

Self-Fulfilling Prophecies Aren't Always About Self-Awareness

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

This is a belated follow-up to my Dualist Predict-O-Matic post, where I share some thoughts re: what could go wrong with the dualist Predict-O-Matic.

Belief in Superpredictors Could Lead to Self-Fulfilling Prophecies

In my previous post, I described a Predict-O-Matic which mostly models the world at a fuzzy resolution, and only "zooms in" to model some part of the world in greater resolution if it thinks knowing the details of that part of the world will improve its prediction. I considered two cases: the case where the Predict-O-Matic sees fit to model itself in high resolution, and the case where it doesn't, and just makes use of a fuzzier "outside view" model of itself.

What sort of outside view models of itself might it use? One possible model is: "I'm not sure how this thing works, but its predictions always seem to come true!"

If the Predict-O-Matic sometimes does forecasting in non-temporal order, it might first figure out what it thinks will happen, then use that to figure out what it thinks its internal fuzzy model of the Predict-O-Matic will predict.

And if it sometimes revisits aspects of its forecast to make them consistent with other aspects of its forecast, it might say: "Hey, if the Predict-O-Matic forecasts X, that will cause X to no longer happen". So it figures out what would *actually* happen if X gets forecasted. Call that X'. Suppose X != X'. Then the new forecast has the Predict-O-Matic predicting X and then X' happens. That can't be right, because outside view says the Predict-O-Matic's predictions always come true. So we'll have the Predict-O-Matic predicting X' in the forecast instead. But wait, if the Predict-O-Matic predicts X', then X'' will happen. Etc., etc. until a fixed point is found.

Some commenters on my previous post talked about how making the Predict-O-Matic self-unaware could be helpful. Note that self-awareness doesn't actually help with this failure mode, if the Predict-O-Matic knows about (or forecasts the development of) *anything* which can be modeled using the outside view "I'm not sure how this thing works, but its predictions always seem to come true!" So the problem here is not self-awareness. It's belief in superpredictors, combined with a particular forecasting algorithm: we're updating our beliefs in a cyclic fashion, or hill-climbing our story of how the future will go until the story seems plausible, or something like that.

Before proposing a solution, it's often valuable to deepen your understanding of the problem.

Glitchy Predictor Simulation Could Step Towards Fixed Points

Let's go back to the case where the Predict-O-Matic sees fit to model itself in high resolution and we get an infinite recurse. Exactly what's going to happen in that case?

I actually think the answer isn't quite obvious, because although the Predict-O-Matic has limited computational resources, its internal model of itself also has limited computational resources. And its internal model's internal model of itself has limited computational resources too. Etc.

Suppose Predict-O-Matic is implemented in a really naive way where it just crashes if it runs out of computational resources. If the toplevel Predict-O-Matic has accurate beliefs about its available compute, then we might see the toplevel Predict-O-Matic crash before any of the simulated Predict-O-Matics crash. Simulating something which has the same amount of compute you do can easily use up all your compute!

But suppose the Predict-O-Matic underestimates the amount of compute it has. Maybe there's some evidence in the environment which misleads it to think that it has less compute than it actually does. So it simulates a restricted-compute version of itself reasonably well. Maybe that restricted-compute version of itself is mislead in the same way, and simulates a double-restricted-compute version of itself.

Maybe this all happens in a way so that the first Predict-O-Matic in the hierarchy to crash is near the bottom, not the top. What then?

Deep in the hierarchy, the Predict-O-Matic simulating the crashed Predict-O-Matic makes predictions about what happens in the world after the crash.

Then the Predict-O-Matic simulating *that* Predict-O-Matic makes a prediction about what happens in a world where the Predict-O-Matic predicts whatever would happen after a crashed Predict-O-Matic.

Then the Predict-O-Matic simulating *that* Predict-O-Matic makes a prediction about what happens in a world where the Predict-O-Matic predicts [what happens in a world where the Predict-O-Matic predicts whatever would happen after a crashed Predict-O-Matic].

Then the Predict-O-Matic simulating *that* Predict-O-Matic makes a prediction about what happens in a world where the Predict-O-Matic predicts [what happens in a world where the Predict-O-Matic predicts [what happens in a world where the Predict-O-Matic predicts whatever would happen after a crashed Predict-O-Matic]].

Predicting *world* gets us *world*', predicting *world*' gets us *world*'', predicting *world*'' gets us *world*'''... Every layer in the hierarchy takes us one step closer to a fixed point.

Note that just like the previous section, this failure mode doesn't depend on self-awareness. It just depends on believing in something which believes it self-simulates.

Repeated Use Could Step Towards Fixed Points

Another way the Predict-O-Matic can step towards fixed points is through simple repeated use. Suppose each time after making a prediction, the Predict-O-Matic gets updated data about how the world is going. In particular, the Predict-O-Matic knows the most recent prediction it made and can forecast how humans will respond to that. Then when the humans ask it for a new prediction, it incorporates the fact of its previous prediction into its forecast and generates a new prediction. You can imagine a scenario where the operators keep asking the Predict-O-Matic the same question over and over again, getting a different answer every time, trying to figure out what's going wrong -- until finally the Predict-O-Matic begins to consistently give a particular answer -- a fixed point it has inadvertently discovered.

As Abram alluded to in one of his comments, the Predict-O-Matic might even forsee this entire process happening, and immediately forecast the fixed point corresponding to the end state. Though, if the forecast is detailed enough, we'll get to see this entire process happening *within* the forecast, which could allow us to avoid an unwanted outcome.

This one doesn't seem to depend on self-awareness either. Consider two Predict-O-Matics with no self-knowledge whatsoever (not even the dualist kind I discussed in my previous post). If they're getting informed about the predictions the other is making, they could inadvertently work together to step towards fixed points.

Solutions

An idea which could address some of these issues: Ask the Predict-O-Matic to make predictions conditional on us ignoring its predictions and not taking any action. Perhaps we'd also want to specify that any existing or future superpredictors will also be ignored in this hypothetical.

Then if we actually want to *do* something about the problems the Predict-O-Matic forsees, we can ask it to predict how the world will go conditional on us taking some particular action.

Choosing better inference algorithms could also be helpful.

Prize

Sorry I was slower than planned on writing this follow-up and choosing a winner. I've decided to give Bunthut a $110 prize (including $10 interest for my slow follow-up). Thanks everyone for your insights.