Not just learning

Paul Christiano Follow
Oct 16, 2016 · 6 min read

(*Note: this post uses the word "learning" in a non-standard way. I mean to include all algorithms that find a model or policy by evaluating similar models/policies a large number of times. This is a defining characteristic of deep learning, and also applies to most existing approaches to learning, but we could imagine alternative algorithms which learn very quickly and to which the discussion in this post does not apply.*)

Over the last year I have been working on value alignment for machine learning systems: how can we teach machines to do what we want?

Learning poses some serious challenges for control and security, and recently it has been the bread and butter of AI. So I think that aligned learning deserves to be a priority, and I'll probably continue to focus on it.

But learning isn't everything. In this post I'll argue that the most powerful future AI systems probably *won't* be learned end-to-end, though they will almost certainly use learned components.

Just as learning systems may fail to be aligned with human interests, other components of an AI system may introduce misalignment even if the building blocks work correctly.

I think that the goal of AI control should be to ensure that for every algorithm or technique that might be useful for building a powerful AI, we have a "safe" version which is:

1. **equally useful**: using the safe version doesn't make a system much more expensive or significantly reduce its ability to influence the world.
2. **alignment-preserving**: as long as all of the building blocks perform their intended functions in the desired way, the overall system performs its intended function in the desired way. (ETA: a clearer discussion).

I’ve been trying to achieve these properties for learning algorithms. But we can and should try to do the same for other AI components; I understand MIRI’s agent foundations agenda as (mostly) addressing the alignment of these other elements.

Why learning isn’t everything and ALBA can’t be sufficient for aligned AI

Learning algorithms require training, which involves evaluating the model-to-be-learned a whole bunch of times. For example, the Q networks in DeepMind’s DQN paper were evaluated 320 million times during training, while AlphaGo’s value network was evaluated 1.5 billion times during training.

As a result, the learned model can use only a small fraction of available computing resources. If we want to make a really good decision, we are going to want to somehow use more computing power.

For example, AlphaGo does not use the learned model directly to make decisions—it could use the policy network directly, but this results in weak moves. Instead, AlphaGo uses MCTS, calling the value network hundreds of thousands of times in order to make a single move.

It seems like this is a fundamental limitation of learning. If we want to make the best decision possible, we are going to want to use as much computing power as we can. And almost by definition, that means that we won’t have enough computing power to train the whole system end-to-end.

ALBA critically exploits this feature of learning. Each learning agent A receives feedback from an overseer H which is much smarter than A. The agent H can be much smarter than A only because it uses much more computational power. Using much more computing than A is feasible only because A needs to be run many times during training (and only if almost all of the runs don’t require labels), and so training already needed to use much more computing power than is required to run A a single time.

This approach simply can’t apply to the most powerful AI that we are capable of building—we can’t define this system’s reward by appealing to a more powerful overseer, because we aren’t capable of building a more powerful overseer.

Examples of {AI} \ {learning}

There are many AI algorithms other than learning, and most of them present possible safety problems.

Search / planning / etc.

One of the simplest and oldest AI algorithms is to to search for an action or plan that is predicted to lead to a desired outcome. We can throw as much computing power as we'd like at the search in order to find better and better elaborate plans, and we don't have to run our search algorithm 300 million times in order to learn how to do it.

If we use a learned model to predict which plans will "lead to a desired outcome," then an expensive search seems most likely to lead to adversarial examples rather than actions that would actually influence the world. But if this difficulty were overcome, then systems based on expensive searches might have a profound influence on the world. We would have no hope of implementing a more powerful overseer, and so we would need to use some different technique to implement a very robust value function.

The most natural approach is to build an explicit model of the user's preferences. This is the goal of IRL and other approaches to value learning, but I think it's safe to say that we don't yet have a clear understanding of how to do this.

Bayesian inference

If we have a distribution over possible worlds and are able to compute conditional probabilities of the form P(observation|world), then we can apply Bayes' theorem to compute P(world|observation). We can extend this basic idea to compute conditional marginal probabilities in complex causal systems or other models.

We can throw a lot of computing power at inference, and the posterior could be more sophisticated than any aligned overseer we can construct.

In order to use inference we need to specify a probabilistic model. Once we are in the business of explicitly specifying probabilistic models it is easy for things to go awry, e.g. to inadvertently specify a prior/model which leads to confident yet unacceptable conclusions; and once the posterior is more sophisticated than any aligned agent that we could construct, its not clear that we can reliably correct an error. This would probably usually lead to inept behavior, but it could also lead to behavior which was sophisticated but unintended.

Logical deduction

Logical deduction is a powerful-if-conservative framework for deriving new beliefs from old beliefs. Logic allows me to derive beliefs like “A⇒Z” from premises like “A⇒B”, “B⇒C,” etc…

(Logical deduction is closely related to Bayesian inference and plausible mechanisms for general reasoning would presumably capture aspects of both of them.)

Logical reasoning can be much more powerful than the reasoning underlying each individual deductive step, and the conclusions can be much more sophisticated than any aligned overseer we can construct. Once again we have the situation where we may need to specify some logical premises, and be unable to effectively correct any unacceptable conclusions that follow from those premises.

For example, if we need to write down moral premises about what outcomes or actions are desirable, but we are skeptical about our own ability to correctly formalize these claims, then we may end up with a system that is able to act effectively in the world yet has a mistaken view about what what is good.

Relation to capability amplification

Capability amplification is the problem of using learned components as a building block to implement a more powerful agent, while maintaining alignment.

In general, capability amplification seems to be substantially easier than developing safe versions of the the non-learning techniques in an AI. For example, if we have alignment-preserving versions of planning / inference / deduction, then we could directly use those techniques as our capability amplification scheme.

More generally, we could break capability amplification up into two parts:

1. Developing safe versions of non-learning techniques.

2. Doing capability amplification while ignoring the “maintaining alignment” requirement.

Part #2 looks much easier than part #1. So it’s likely that capability amplification will be radically easier if we can solve the rest of the alignment problem. That suggests we should put capability amplification on hold while we work on other aspects of the alignment problem, or at least we should avoid any angles of attack on capability amplification which aren’t also promising angles on the rest of the alignment problem.

Conclusion

I think that figuring out how to do learning in an aligned way is one of the most important problems in AI control. But the most powerful AI systems are not likely to be learned end-to-end, and so aligned learning is not the end of the story.