Reward hacking

Author: Barry Morisse

Moving beyond goal-driven behavior

At the risk of significantly over-simplifying the field, software development is achieved by articulating a certain goal or objective and mapping out a logical, algorithmically-based method of getting there. It takes in various inputs, performs some sort of processing and then releases some form of output which will hopefully meet the objective of the developer. The objective of the software is what drives every decision – be it the functionality, the design, the efficiency, the flexibility or any other range of factors. As a result, the best developers tend to be the best problem solvers – who can deliver on the stated goal/objective in the most efficient way.

These skills will become even more crucial as we move closer to advanced artificial intelligence (AI). A plausible end-game for this type of technology is AI that is capable of learning how to alter its own code, without developer interference, and move towards the stated objective in ways we can't even begin to comprehend. These machines won't be limited to the finite number of problem-solving apparatus we have as humans – but will be able to attempt an infinite number of methods to move towards the stated goal. Therefore, the goal we program into an advanced AI is incredibly important as it will be the only lever we get to play with. We have to get it right.

This is exactly where most of the doomsday prophets get their material. In the world of AI, it is called the alignment problem. How do we program a machine with goal that won't be misinterpreted and delivered in a way that we don't expect or can't predict? The now infamous thought experiment that illustrates this scenario was presented by philosopher Nick Bostrom and it goes something like this:

The Paperclip Maximiser Experiment

Imagine an advanced AI whose sole purpose is to create paperclips. This seemingly innocuous goal is the only thing this machine is designed to do – and so it is released to explore the possibilities and is unhampered by human programming. It begins by creating paperclips in the same way as humans have always done. It learns from best practices and repeats these at ever-increasing speeds. After a short while, it starts to experiment a bit and tries different ways of creating paperclips, improving the process and its efficiency – which impresses its human creators. Over time, it improves the efficiency to such an extent that it is a perfect process. There is no more improvement to be found, so now it turns to quantity. In service of its ultimate goal of maximising the number of paperclips, it starts to look for additional resources, be it additional machines that can be converted to paperclip maximisers, repurposing the metal from other machinery nearby, even moving drastically towards mining nearby mines for metal or harnessing any matter whatsoever to create the atoms necessary to make more paperclips.

The machine begins to run away from humans and goes way beyond what they actually wanted because its single-minded purpose is the one goal it was programmed with in the first place. It's a startling thought experiment that has rightly received some criticism because of its over-simplification – but the visual it creates still does the work in explaining the point. We will never be sophisticated enough to imagine exactly how our goal will be interpreted by an advanced AI, no matter how specific you get or how many caveats you include. At the end of the day, we will need to find a way to move beyond goal-driven behaviour for these types of intelligences because otherwise the risk that they will run amok will be significant.

It's clear that we haven't figured out a solution to this just yet, because throughout the lifecycle of software development, the goal-driven nature has been the driving force that has made it so powerful. A significant paradigm shift will be needed to get away from that.

One idea that is gathering steam in this regard is something that harks back to our evolutionary predisposition to make decisions based on pain and pleasure, failure and reward. Our personalities and the ways we learn are shaped by the types of reinforcement we receive from those people who raise us as children, and this moulds us into the type of person we become – instilling in us values and morals that should hold fast no matter what the situation or the goal we move towards.

So, in a technological sense, perhaps the way forward is to aim to instil this same type of guiding compass into the AI to constrain its abilities somewhat and make it more controllable. This could possibly be achieved through creating a desire for some sort of reward and then playing on that desire through positive reinforcement, while the machine tackles a certain goal. Isaac Asimov's three laws of robotics come to mind here: "1) A robot may not injure a human being or, through inaction, allow a human being to come to harm. 2) A robot must obey orders given it by human beings except where such orders would conflict with the First Law. 3) A robot must protect its own existence as long as such protection does not conflict with the First or Second Law." While initially written as a science fiction trope, those kinds of looped logical models capture much of our intuition here and may be a good starting point for the kind of reinforcement that will be needed.

Moving beyond purely goal-driven decision-making is going to be exceedingly difficult and somewhat antithetical to the way programming is done today. But it's a crucial piece of the puzzle. As we unleash advanced AI on the biggest problems we face in our world today, we need to be confident that we don't unknowingly invite an externality that puts us in danger. To keep our interests aligned, and ideally humans in control, we need a new paradigm. That's the only way to harness the extraordinary power while still maintaining a sustainable ecosystem for our species to thrive.