Learning with catastrophes

Paul Christiano Follow
May 28, 2016 · 5 min read

A *catastrophe* is an event so bad that we are not willing to let it happen even a single time. For example, we would be unhappy if our self-driving car *ever* accelerates to 65 mph in a residential area and hits a pedestrian.

Catastrophes present a theoretical challenge for traditional machine learning—typically there is no way to reliably avoid catastrophic behavior without strong statistical assumptions.

In this post, I'll lay out a very general model for catastrophes in which they are avoidable under much weaker statistical assumptions. I think this framework applies to the most important kinds of catastrophe, and will be especially relevant to AI alignment.

Designing practical algorithms that work in this model is an open problem. In a subsequent post I describe what I currently see as the most promising angles of attack.

Modeling catastrophes

We consider an agent A interacting with the environment over a sequence of episodes. Each episode produces a transcript τ, consisting of the agent's observations and actions, along with a reward $r \in [0, 1]$. Our primary goal is to quickly learn an agent which receives high reward. (Supervised learning is the special case where each transcripts consist of a single input and a label for that input.)

While training, we assume that we have an oracle which can determine whether a transcript τ is "catastrophic." For example, we might show a transcript to a QA analyst and ask them if it looks catastrophic. This oracle can be applied to arbitrary sequences of observations and actions, including those that don't arise from an actual episode. So training can begin before the very first interaction with nature, using only calls to the oracle.

Intuitively, a transcript should only be marked catastrophic if it satisfies two conditions:

1. The agent made a catastrophically bad decision.
2. The agent's observations are plausible: we have a right to expect the agent to be able to handle those observations.

While actually interacting with the environment, the agent cannot query the oracle—there is no time to wait for a QA engineer to review a proposed action to check if it would be catastrophic.

Moreover, if interaction with nature ever produces a catastrophic transcript, we immediately fail. The performance of an algorithm is characterized by two parameters: the probability of catastrophic failure, and the total reward assuming no catastrophic failure.

We assume that there are some policies such that no matter what nature does, the resulting transcript is *never* catastrophic.

Traditionally in RL the goal is to get as much reward as the best policy from some class C. We' slightly weaken that goal, and instead aim to do as well as the best policy from C that never makes a catastrophic decision.

Batch learning

I've described an online version of learning with catastrophes. We can also consider the batch version, where the learner is first given a large number of "training" episodes.

In the batch version, there is no penalty for catastrophes at training time, and we don't care about training error. The two performance parameters are test-time performance and test-time catastrophe probability.

The oracle

This definition depends on an oracle who determines which transcripts are catastrophic.

For weak AI systems, the oracle may be a human. But a powerful AI system might take actions which are catastrophic but which look inoffensive to a human judge, so this approach doesn't cut it.

In general, the judge should be a human+AI team which is *more* competent than the system being trained, armed with an adequate solution to the informed oversight problem.

Approach

Learning with catastrophes is straightforward *given an unlimited number of queries to the catastrophe oracle*. Given any online learning algorithm A, we can "harden" it by running the following process before the beginning of each episode (a similar process can be applied to a batch learner):

1. Search over all possible environments, running A on each one to obtain a transcript.
2. If we find a catastrophic transcript τ, then add τ to A's training data with a reward of -1, and go back to step 1.

It's easy to prove that this process converges, if A is competent: the number of times we invoke step 2 is at most the time required to learn an optimal catastrophe-free policy (plus the number of episodes).

The big problem with this algorithm is the exponential search in step 1.

Adversarial training

The most obvious approach to *tractable* learning with catastrophes is to replace step 1 with a tractable algorithm that approximates the search. This process is an example of adversarial training.

We can formalize this as a simple game between the agent A and an adversary B. The two agents alternate taking moves, with A playing the role of the agent and B playing the role of the environment. B wins if the resulting transcript is catastrophic, A wins if it is not catastrophic.

Before each episode, we play this game N times, where $N > 0$ is a hyperparameter. If B is able to win *any* of the games, then we add the resulting transcript to A's training data and start again. Once B is not able to win any games, then we move on to a "real" episode. We hope that B is good enough at the game that if it can't produce a catastrophic transcript, then the real environment can't produce a catastrophic transcript either.

More precisely, before each episode we perform the following process:

1. Set $i = 0$.

2. A and B alternate taking moves, producing a transcript τ.
3. If τ is catastrophic, we add τ to A's training data with a reward of -1, and add τ to B's training data with a reward of $+1$. Then we go back to step 1.
4. If τ is not catastrophic, we add τ to B's training data with a reward of -1.
5. If $i < N$, we increment i and go back to step 2.

I discuss this idea in more detail in my post on red teams. There are serious problems with this approach and I don't think it can work on its own, but fortunately it seems combinable with other techniques.

Conclusion

Learning with catastrophes is a very general model of catastrophic failures which avoids being obviously impossible. I think that designing competent algorithms for learning with catastrophes may be an important ingredient in a successful approach to AI alignment.