Injecting Machine Learning into the Apprentice Learner Architecture, Project Milestone Report 6
15-400, Spring 2020

Cayden Codel (ccodel)
Project website: www.contrib.andrew.cmu.edu/∼ccodel

April 22, 2020

1 Major Changes

No major changes to report.

2 What I've accomplished since last meeting

Since last meeting, I have successfully transferred to the new agent type and have gotten an interactive training interface working.

The new agent type, called the SoarTech agent (after the private company one of the AL members works at), adjusts how the AL agent creates rules and forms prior knowledge. Previously, AL agents had access to a few primitive operations (such as addition, multiplication, reducing common denominators, etc.), but would have to form rules for how to apply these primitive operations on the state given to it. The SoarTech agent instead allows for rules (and other primitive operations) to be hard-coded into the agent as prior knowledge. This means that, if an expressive enough prior-skills ruleset is given to an agent, it can successfully complete its task, even if it has done no learning yet.

How this applies to RumbleBlocks is: I have created a suite of rules the SoarTech agent has at its disposal to "play" the game without learning anything. The suite of rules includes how to place blocks on other blocks, how to rotate blocks, and how to observe checkpoints and goals necessary to complete any particular level.

Initial tests show that the SoarTech agent is manipulating the game state in its working memory—by rotating blocks, etc.—but a little more is necessary for it to play correctly. Perhaps some reward shaping is required (as currently, the agent only knows if it is correct when it finishes a level or times out).

Dr. Erik Harpstead found the old code for an interactive training interface for RumbleBlocks in a previous commit. I've taken the GUI code and put it back into the project, so the infrastructure for interactive training is in place. However, some "bugs" were discovered in the AL code. As it turns out, AL typically trains with an HTML CTAT tutor, which means that values are pulled from an HTML field with a "value" tag. That means that AL has been hard-coded to pull data from HTML fields, moving away from general state input. This was a TODO marked in the code base, and so Dr. Harpstead has put me in charge for the week to see if I can't get AL back to general input. Thus, that is my task for the next week or two (along with getting the new agent to successfully train in RumbleBlocks).

There were also a few Unity issues that needed to be fixed, such as object colliders and particle effects, due to porting to a newer version of Unity.

3 Milestone progress

The SoarTech agent and interactive training interfaces have been completed, excepting some input problems that shouldn't be too hard to fix (for example, floats are being serialized to an inconsistent number of decimal places, causing the agent to not interpret the game state correctly). Once input problems have been fixed, agent rewards can be tweaked to get an agent that not only understands how to manipulate blocks, but how to complete levels.

4 Surprises

No surprises.

5 Looking ahead

Training an agent to play RumbleBlocks!

And designing a poster, of course. Dr. Harpstead has a few ideas for how it should look, and there are a couple of papers that have been done on RumbleBlocks so far that I could look at for inspiration.