

RL-Competition, RL-Viz, and RL-Library

It's Dec 2007. I've been working roughly since I got back from India in January on various software packages, all of which have had various code-names over the last year: bt-glue, bt-viz, bt-library, rl-viz, etc. These are all incarnations of the same ideas, that have come up over and over again. We need tools for reinforcement learning that allow us to quickly plug and play reinforcement learning agents and environments together, and to visualize, inspect and debug their behavior.

This project started as a Mac-only project based in the C4 Game Engine architecture, it then moved to be a stand-alone Cocoa Mac application. Then I ported it to Java as a stand-alone application. Somewhere in there Adam White convinced me to make it work with RL-Glue, which was a bunch of work, because instead of using direct method calls, everything had to be rewritten to work as remote procedure calls through the ew agent_message and env_message parts of the RL-Glue interface.

The software has many attractive features, like allowing you to choose the agent and environment AND their parameters at runtime, control the experiment in an interactive way, and visualize whatever aspects of the agent and environment you want (value function visualization, trajectory tracking, graphical environment visualizer, etc).

The software has become the base for the upcoming Reinforcement Learning Competition: <http://rl-competition.org>. I get to exert some influence in these decisions as I am the Technical Committee Chair (sounds fancy) for the competition, which basically means I am writing or supervising much of the software and infrastructure development for the competition.

Ok, so now I've said a bit, I'm not going to paste in an announcement that I made on the competition website.

----- It's nothing fancy, but we've decided to announce the location of the code-base that we use for the competition distribution.

<http://rl-competition.googlecode.com/>

We do NOT host any of the private source code for the domains, the proving application, or the proving or test MDPs on this website. Instead, the site is where those products go once they're built. This also includes the scripts that we use to run things, the source code for all of the sample agents and trainers, as well as the source code for the real time strategy domain and eventually keepaway.

The idea here is that the community can find ways to make all of these things work better, and should feel empowered to push those contributions back into the competition community. Some of the issues we have fixed with new software releases are simple things that anyone could have done.

This will also provide us with a more direct way to releasing tiny bug fixes without having to test and roll out a whole new release of the software (that process takes a whole day).

I can also take this chance to advertise for another google code project:

<http://rl-viz.googlecode.com/>

This is the software platform that all of the java stuff is built on, including the graphical and console trainers, all of the dynamic loading, the environment parameterizations, etc, etc. We haven't officially made a release of the RL-Viz software yet, but it's coming along. May of the fixes that we've made to the competition software are actually improvements and fixes to RL-Viz.

Finally, we will soon start advertising the RL-Library, which is a repository for environments, agents, and experiment programs based on RL-Glue and optionally on RL-Viz. We hope to release vanilla, non-generalized, standard versions of all of the competition domains to the RL-Library in the near future.

Of course, your involvement and contributions are welcomed to all of these projects.

Oh, and of course, don't forget the nuts and bolts underlying everything, RL-Glue:

<http://rl-glue.googlecode.com/>

