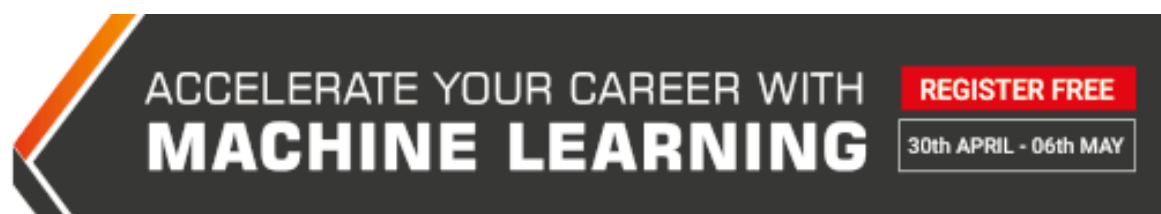


Getting ready for AI based gaming agents – Overview of Open Source Reinforcement Learning Platforms



Introduction

We are living in exciting times. We are all set to create an army of smart machines and robots. Creating these machines has been a dream and one of the biggest challenges humans have faced. What adds to this excitement is that no one knows how these smart machines and robots will impact us in return. Will they end up taking people out of their jobs? Or Will they create new avenues and opportunities, which we humans can't think of as of now! One thing is for sure though – there is a lot of automation about to happen here!

Researchers have created a [roadmap](#) for machine intelligence research. Following suite, some major platforms have been built to give way to this research. In this article, I explain Reinforcement learning in simple terms and compare major platforms for testing reinforcement learning algorithms. Kindly note, I have included only those platforms which have projects dedicated for the environment and not those which have integrated support for reinforcement learning algorithms.

These platforms should enable a generation of research and new findings / developments in Artificial Intelligence and Machine Learning.

Table of Contents

1. What is Reinforcement Learning?
2. Examples of Reinforcement Learning
3. What is a Reinforcement Learning Platform?
4. Examples of Reinforcement Learning Platform
5. Cheatsheet of Major Platforms
6. A few other notable platforms

1. What is Reinforcement learning?

Let us start with a simple analogy. If you have a pet at home, you may have used this technique with your pet.



A clicker (or whistle) is a technique to let your pet know some treat is just about to get served! This is essentially “reinforcing” your pet to practice good behavior. You click the “clicker” and follow up with a treat. And with time, your pet gets accustomed to this sound and responds every time he/she hears the click sound. With this technique, you can train your pet to do “good” deeds when required.

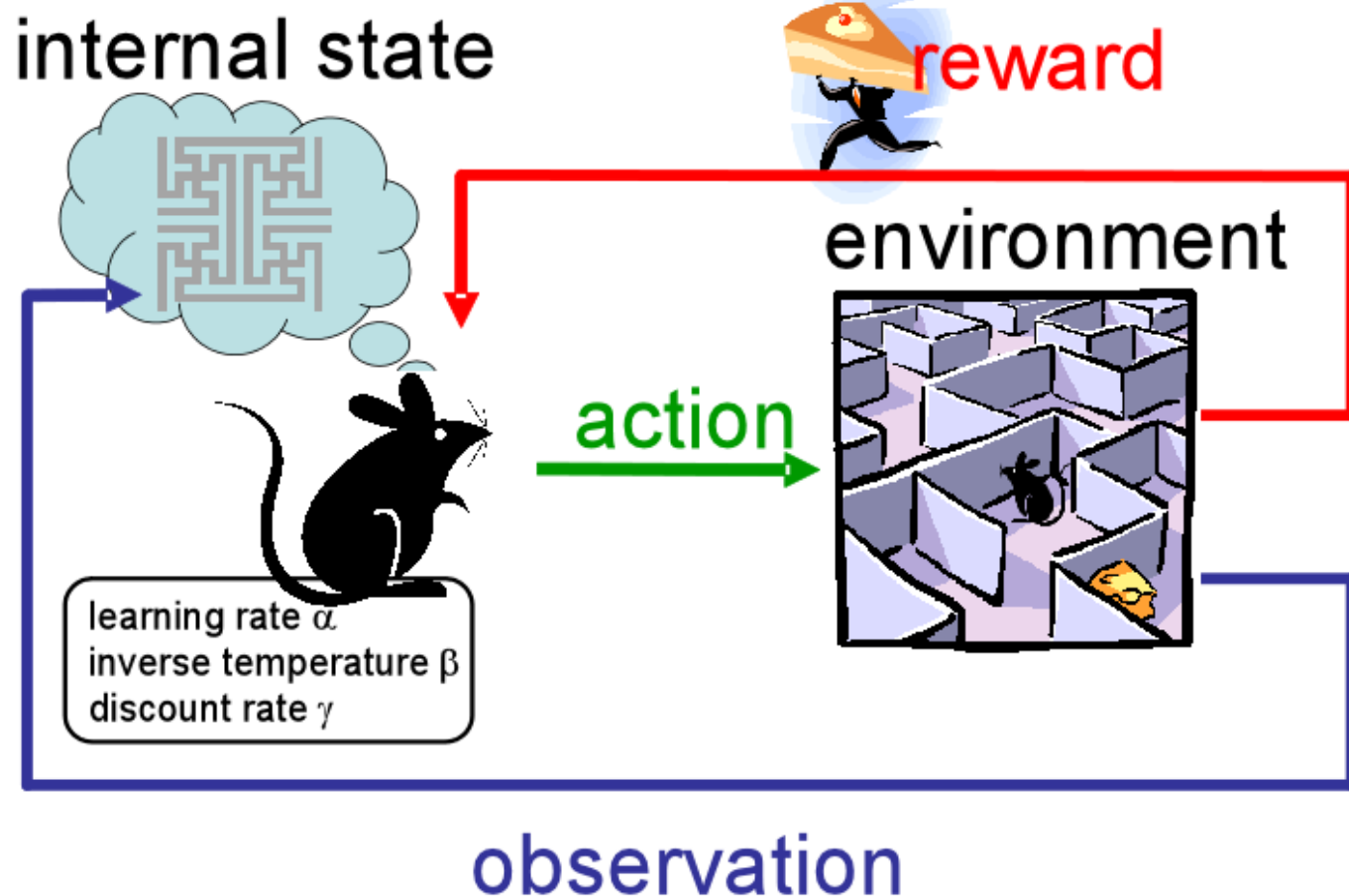
Now let’s make these replacements in the example:

- The pet becomes the artificial agent
- The treat becomes the reward function
- The good behavior is the resultant action

The above example explains what reinforcement learning looks like. This is actually a classic example of reinforcement learning.

To apply this on an artificial agent, you have a kind of a feedback loop to reinforce your agent. It rewards when the actions performed is right and punishes in-case it was wrong. Basically what you have in your kitty is:

- an **internal state**, which is maintained by the agent to learn about the environment
- a **reward function**, which is used to train your agent how to behave
- an **environment**, which is a scenario the agent has to face
- an **action**, which is done by the agent in the environment
- and last but not the least, an **agent** which does all the deeds!



Source: [UTCS RL Reading Group](#)

2. Examples of Reinforcement Learning

Now, I am sure you must be thinking how the experiment conducted on animals can be relevant to people practicing machine learning. This is what I thought when I came across reinforcement learning first.

A lot of beginners tend to think that there are only 2 types of problems in machine learning – Supervised machine learning and Unsupervised machine learning. I don't know where this notion comes from, but the world of machine learning is much more than the 2 types of problems mentioned above. Reinforcement learning is one such class of problems.

Let's look at some real-life applications of reinforcement learning. Generally, we know the start state and the end state of an agent, but there could be multiple paths to reach the end state – reinforcement learning finds an application in these scenarios.

This essentially means that driverless cars, self navigating vaccum cleaners, scheduling of elevators are all applications of Reinforcement learning.

Here is a video of a game bot trained to play flappy bird.

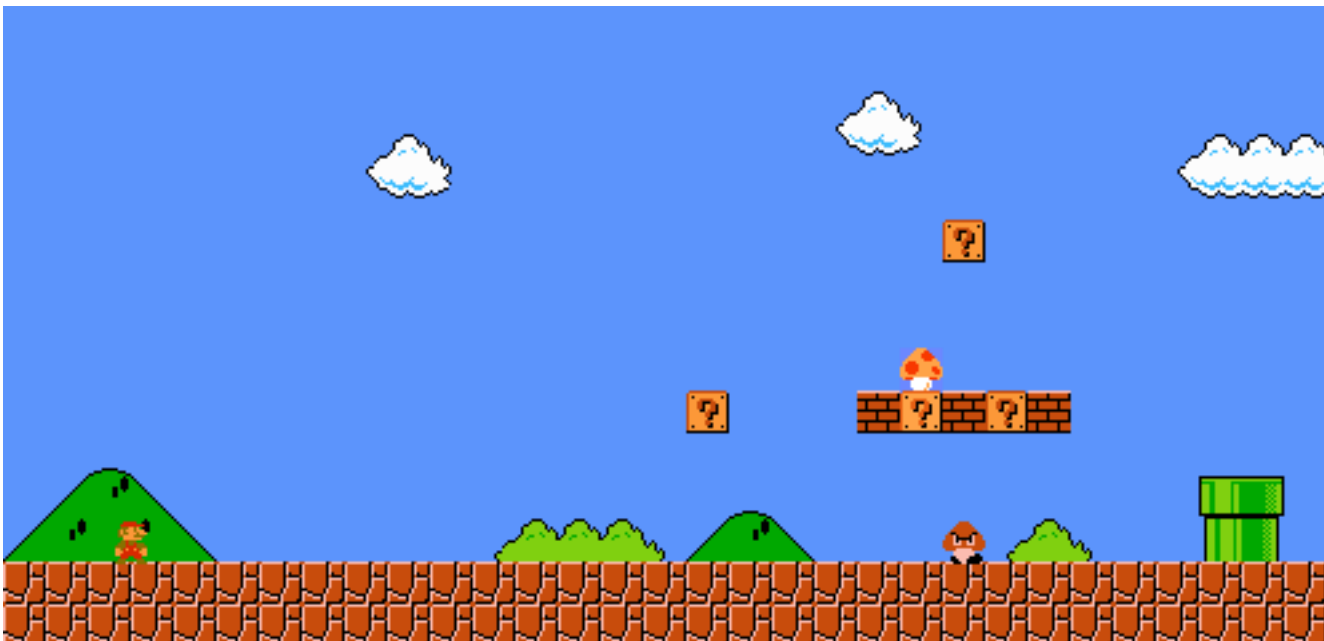
3. What is a Reinforcement Learning Platform?

Before we look into what a platform is, let's try to understand a reinforcement learning environment.

A reinforcement learning environment is what an agent can observe and act upon. The horizon of an agent is much bigger, but it is the task of the agent to perform actions on the environment which can help it maximize its reward. As per ["A brief introduction to reinforcement learning"](#) by Murphy (1998),

The environment is modeled as a stochastic finite state machine with inputs (actions sent from the agent) and outputs (observations and rewards sent to the agent).

Let's take an example,



This is a typical game of mario. Remember how you played this game. Now consider that you are the “agent” who is playing the game.

Now you have “access” to a land of opportunities, but you don’t know what will happen when you do something, say smash a brick. You can see a limited amount of “environment”, and until you traverse around the world you can’t see everything. So you move around the world, trying to perceive what entails ahead of you, and at the same time try to increase your chances to attain your goal.

This whole “story” is not created by itself. You have to “render” it first. And that is the main task of the platform, viz to create everything required for a complete experience – the environment, the agent and the rewards.

4. Major Reinforcement Learning Platforms

i) Deepmind Lab

DeepMind Lab is a fully 3D game-like platform tailored for agent-based AI research

A recent release by Google Deepmind, **Deepmind lab** is an integrated agent-environment platform for general artificial intelligence research with a focus on first person perspective games. It was built to accomodate the research done at DeepMind. Deepmind lab is based on an open-source engine ioquake3 , which was modified to be a flexible interface for integration with artificial systems.

Things I liked

- It has richer and realistic visuals.
- Closer integration with the gaming environment

Things I did not like

- It still lacks variety in terms of a gaming environment, which would get built over time by open source contributions.
- Also at the moment, it supports only Linux, but has been tested on different OS. Bazel (which is a dependency for deepmind lab) is experimental for windows. So windows support for Deepmind lab is still not guaranteed.

Resources to explore further:

- [Release post](#)
- [Open-source Repository](#)
- [DeepMindLab](#)
- [Simple tutorial](#)

ii) OpenAI Gym

(OpenAI Gym is) A toolkit for developing and comparing reinforcement learning algorithms

OpenAI Gym is a platform for creating, evaluating and benchmarking artificial agents in a game environment. The best thing I like about gym is that along with the toolkit, there is a community support built around it, viz an evaluation platform, code sharing platform and a discussion platform. Gym platform consists multiple categories of environment along with sample solutions provided by the community

Things I liked

- Variety of game environments with considerable open-source support.

Things I did not like

- Like Deepmind lab, Gym also has a limit on the number of environments it supports (This is essentially taken care by OpenAI Universe)

Resources to explore further:

- [Release post](#)
- [Open-source Repository](#)
- [Whitepaper](#)
- [Simple tutorial](#)

iii) OpenAI Universe

[Universe](#) is a software platform for measuring and training an AI's general intelligence across the world's supply of games, websites and other applications

This is essentially an extension to OpenAI gym, with support for literally “anything” you can do on a computer. Universe is built to emulate how a human interacts with a computer. It uses [Virtual Network Computing](#) to access a computer remotely, packages any program and converts it into a gym environment.

Things I liked

- Unlimited access to any game environment.
- Universe is not only constrained to gaming environment, it can be used to replace things like manual testing and working on Amazon Mechanical Turk

Things I did not like

- Initial release lacks many things which were promised, the significant one is integration with Windows.

Resources to explore further:

- [Release post](#)
- [Open-source Repository](#)
- [Simple tutorial](#)

iv) Project Malmo

The Malmo platform is a sophisticated AI experimentation platform built on top of Minecraft, and designed to support fundamental research in artificial intelligence.

Project Malmo is a research initiative by Microsoft research, with an aim to build AI agents to do complex tasks. Minecraft is a perfect scenario for building AI agents, and that is why they chose it.

Things I liked

- Integration with a sophisticated environment
- Flexibility for customizing game environment

Things I did not like

- The support is only for minecraft, and no other game environment unlike OpenAI universe.

Resources to explore further:

- [Release Post](#)
- [Open-source Repository](#)
- [Whitepaper](#)
- [Simple tutorial](#)

v) VizDoom

Doom-based AI Research Platform for Reinforcement Learning from Raw Visual Information

I personally found this the most interesting platform to build AI agents, as you can have a multi-agent support with a competitive environment to test the agent. The platform runs on Doom, a first person shooting game, with a variety of levels and modes.

Things I liked

- Variety of game modes with competitive environment

Things I did not like

- Similar to above platform, VizDoom has only support for one environment.

Resources to explore further:

- [Release Post](#)
- [Open-source Repository](#)
- [Whitepaper](#)
- [Simple tutorial](#)

5. Cheatsheet of Major Platforms

	Deepmind lab	Gym	Universe	Vizia	Project Malmo
Topic					
Open Source	Yes	Yes	Yes	Yes	Yes
Game integration	Tightly	Tightly	Superficial	Tightly	Tightly
Game customization	Yes	Yes	No	Yes	Yes
Release organization	DeepMind	OpenAI	OpenAI	Poznan University of Technology, Poland	Microsoft
Language	python, lua	python	python	C++, python, Lua and Java	python, lua, C++, C#, Java
Game diversity	Limited	Limited	Unlimited*	Only Doom	Only Minecraft

A few other notable platforms

- RL-Glue
 - About: A standard interface for variety of languages to connect agents, environments and experiment programs together.
 - Resource: [Main wiki Page](#)
- CommAI
 - About: A platform for training and testing AI systems based on communication tasks
 - Resource: [Github repo](#)
- Burlap
 - About: BURLAP is a java code library for the use and development of single or multi-agent planning and learning algorithms and domains to accompany them.
 - Resource: [Github repo](#)
- rlenvs
 - About: A platform similar to OpenAI Gym but for lua
 - Resource: [Github repo](#)

Acknowledgements

Thanks to AV community and reddit community for the helpful discussions. Special thanks to [johny_cauchy](#), [kendingpku](#) and [Kaixhin](#) for their feedback.

End Notes

In this article, we briefly looked at what reinforcement learning is. I have listed all the major platforms for RL research. Most of these rely on gaming environments to simulate real life conditions. If you know other platforms for reinforcement learning, do let me know in the comments below!

Have you worked on any of these platforms? Share your experience by dropping in your comments. If you have any doubts / suggestions / feedback I would love you hear it. Feel free to post your comments.

You can test your skills and knowledge. Check out [Live Competitions](#) and compete with best Data Scientists from all over the world.