

Round 1: Completed

#reinforcement\_learning

#classroom

# IITM RL Final Project

By [Alcrowd](#) & [IIT Madras](#)

1664

64

25

251

Follow

[Getting Started Code with Random Predictions](#)

[Have a question? Visit the discussion forum](#)

## BSuite Benchmark for Reinforcement Learning

This notebook uses an open-source reinforcement learning benchmark known as bsuite.

<https://github.com/deepmind/bsuite>

BSuite is a collection of carefully-designed experiments that investigate core capabilities of a reinforcement learning agent.

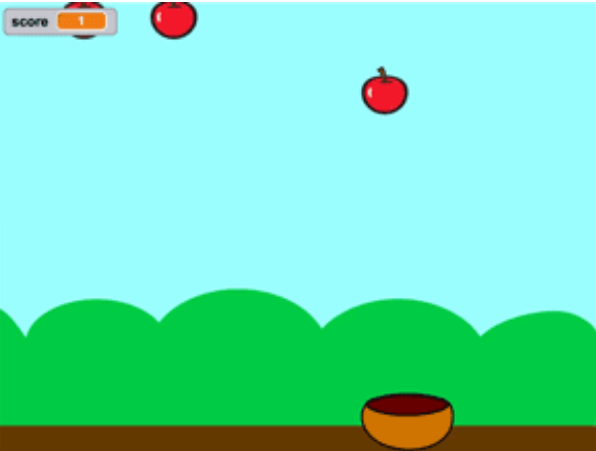
Your task is to use any reinforcement learning techniques at your disposal to get high scores on the environments specified.

**Note:** Since the course is on Reinforcement Learning, please limit yourself to using traditional Reinforcement Learning algorithms.

**Do not use deep reinforcement learning.**

You will be implementing a traditional RL algorithm to solve 3 environments.

## Environment 1: CATCH



In this environment , the agent must move a paddle to intercept falling balls. Falling balls only move downwards on the column they are in.

The observation is an array shape (rows, columns), with binary values: 0 if a space is empty; 1 if it contains the paddle or a ball.

### PARTICIPANTS

C

D

S

A

S

P

S

E

S

G

P

42.

H

J

P

N

R

I

V

N

N

R

T

A

N

M

A

S

J

...

### NOTEBOOKS

See all

1

**BSuite Challenge Starter Kit**

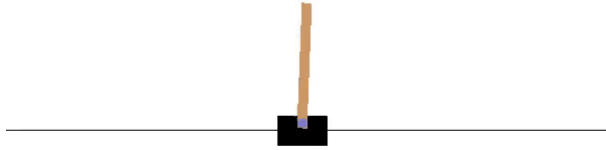
By dipam 4 months ago

0

The actions 3 discrete actions possible: ['stay', 'left', 'right'].

The episode terminates when the ball reaches the bottom of the screen.

## Environment 2: CARTPOLE



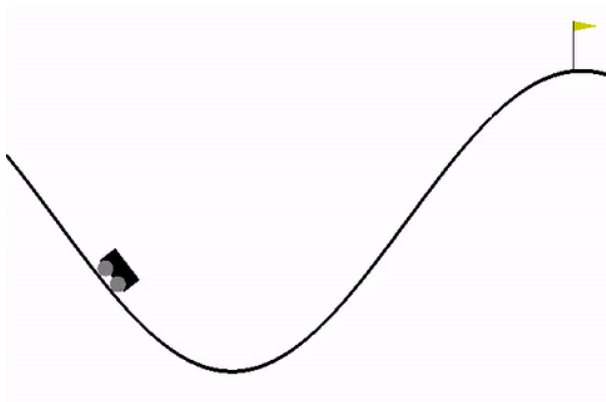
This environment implements a version of the classic Cartpole task, where the cart has to counter the movements of the pole to prevent it from falling over.

The observation is a vector representing: (x, x\_dot, sin(theta), cos(theta), theta\_dot, time\_elapsed)

The actions are discrete and there are 3 of them available: ['left', 'stay', 'right'].

Episodes start with the pole close to upright. Episodes end when the pole falls, the cart falls off the table, or the max\_time is reached.

## Environment 3: MOUNTAIN CAR



This environment implements a version of the classic Mountain Car problem where an underpowered car must power up a hill

[Overview](#) [Leaderboard](#) [Notebooks](#) [Discussion](#) [Insights](#) [Resources](#) [Submissions](#) [Rules](#)

My Team

There are 3 discrete actions available: ['push left', 'no push', 'push right']

Episodes start with the car at the bottom of the hill with no velocity. An episode ends when you reach position  $x=0.5$ , or if 1000 steps have been completed.

Each environment has a NOISE variant which adds a scaled random noise to the received rewards. More details in the [BSuite Paper](#).


## Submission

Before submitting, make sure to accept the [rules](#).

Go to the [starter kit](#) notebook and follow the instructions to implement your agent in the notebook.

# Scoring

We use BSuite's scoring system to determine score for each environment. The final score is the sum of all the test environments' scores.



PLATFORM

[Challenges](#)

[Organize a Challenge](#)

[FAQ](#)


[Our Team](#)


[We are Hiring!](#)


CONNECT


[Blog](#)

[Contact](#)

[GitHub](#) 

[GitLab](#) 

[Twitter](#) 

[Discourse](#) 

LEGAL

[Cookie consent](#)

[Terms of Use](#)

[Participation Terms](#)

[Privacy](#)

© 2021 Alcrowd. All rights reserved.