

Reinforcement Learning

Coursework Introduction

Mhairi Dunion, Trevor McInroe

13 February 2024



THE UNIVERSITY *of* EDINBURGH
informatics

Lecture Outline

- Coursework Outline
- Marking Details
- Submission Instructions
- Getting Started
- Contact
- Demonstration Session

Coursework Outline

Material Covered

- Dynamic Programming (15 Marks)
- Tabular Reinforcement Learning (20 Marks)
- Deep Reinforcement Learning for Discrete Actions (32 Marks)
- Deep Reinforcement Learning for Continuous Actions (18 Marks)
- Fine-tuning the Algorithms (15 Marks)

Question 1 - Dynamic Programming

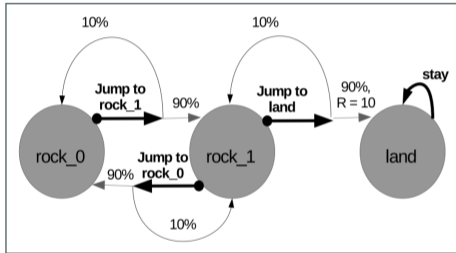


Figure 1: Example MDP for Exercise 1

- Implement functions for **Value Iteration & Policy Iteration**
- Marked based on correctness of your implementation
- Create your own MDPs to debug your implementation

Question 2 - Tabular Reinforcement Learning



Figure 2: FrozenLake Environment

- Implement functions for **Q-Learning** and **Monte Carlo** with ϵ -greedy policy
- Marked based on:
 - Correctness of Implementation
 - Hyperparameter tuning questions

Question 3 - Deep Reinforcement Learning

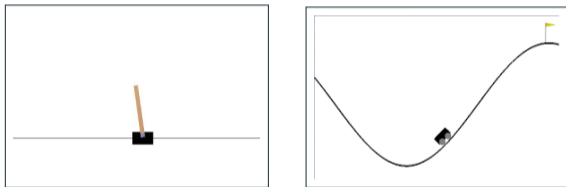


Figure 3: Rendering of the Cartpole and MountainCar environments

- Implement deep reinforcement learning algorithms
 - Value-based: **DQN**
 - Policy gradient: **REINFORCE**
- Marked based on:
 - Correctness of Implementation
 - Hyperparameter tuning & scheduling questions
 - Loss understanding questions

Question 4 - Deep Deterministic Policy Gradient

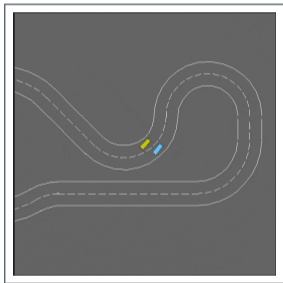


Figure 4: Rendering of the Racetrack environment

- Implement the Deep Deterministic Policy Gradient (DDPG) algorithm
- Stable hyperparameters for Racetrack are provided to check your algorithm's correctness
- Tune the deep networks' sizes
- Provide a trained model for DDPG in Racetrack
- Marked based on:
 - Correctness of Implementation
 - Mean returns achieved by your agent at the end of training

Question 5 - Fine-tuning the Algorithms

- Same setting and environment as in Q4, but harder performance targets
- You are given complete freedom over hyperparameter tuning and scheduling
- Provide a trained model for DDPG in Racetrack
- Marking based on:
 - Mean end of training returns achieved by your agent
 - A description of your hyperparameter tuning strategy

Organisation

Marking Details

- Unit tests to evaluate correctness
 - Read documentations for desired outputs of each function
- Evaluation of performance measured by average returns
 - See coursework document for expected performance of each environment
- Write-up
 - Fill the functions in *answer_sheet.py*

Q4 and Q5 Performance Marks

- You need to provide trained models for
 - Q4 DDPG in Racetrack
 - Q5 DDPG in Racetrack (\neq the one for Q4)
- Make sure you also provide the correct configuration dict for your model in *train_ddpg.py*

Submission Details

- Organize submission files as instructed in coursework document
- Submit as zip file through LEARN
- LEARN assignment submission guidelines:

*[https://blogs.ed.ac.uk/ilts/2019/09/27/
assignment-hand-ins-for-learn-guidance-for-students/](https://blogs.ed.ac.uk/ilts/2019/09/27/assignment-hand-ins-for-learn-guidance-for-students/)*

- **Deadline: March 29th 2024, 12:00 Noon**

Getting Started

Getting Started

- Read the coursework description (download from the [course page](#))
- Read the code base documentation
- Understand the training script for each question
- Read the cited papers for each question

Codebase Setup

- Install Python3
- Set up virtual environment
- Download code base

```
git clone https://github.com/uoel-agents/uoel-rl2024-  
coursework
```

- Install package dependencies

```
pip install -e .
```


Support

- Questions should be posted on Piazza
- Clarification will be posted by TA team (Mhairi, Trevor)
- Use tags in Piazza to organize posts for easy search
- Please do not discuss implementation details!

Demonstration Session

- Labs will be held from W5 (this week!) to W8
- W5: Installation & setup troubleshooting
- W6: PyTorch Demonstration
- W7: Q&A with priority on Q1, Q2 and Q3
- W8: Q&A with priority on Q4 and Q5

Come with questions prepared! These are Q&A's, not lectures!

Common Pitfalls!

- Extension deadlines are strict! Never submit late on an extension!
- Forks are public. Keep your code private and never share it.
- Start early!
- Q5 is meant to be harder than the rest of the assignment. Focus on Q1-Q4 first.
- Following suggestions of our [RL tutorial lecture](#), keep track of performance and hyperparameters using plotting tools.
- Do not implement optimisations or tricks outside of the provided specifications (even if they improve performance)!

Any Questions?

Codebase introduction