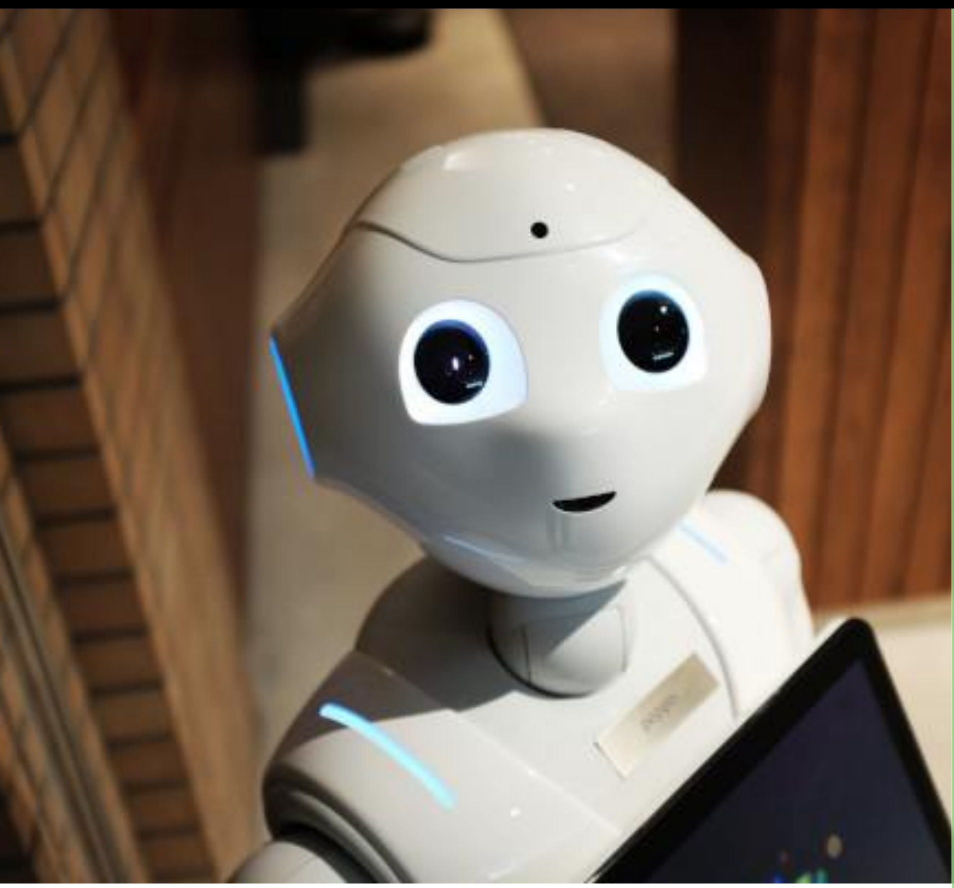
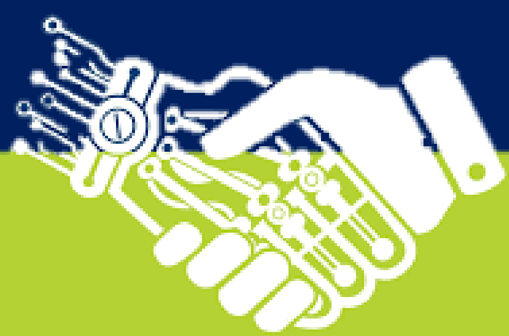
2024

Reinforcement Learning for Bioengineers (BIOE70077) Coursework-2



IMPERIAL



Dr Faraz Janan (Module Leader)
Department Of Bioengineering
Imperial College London, UK

IMPERIAL

Important Information

Method of Submission

Submission via Blackboard; 12th December 2024 @3 PM (check Acadiary for any changes).

Naming convention for the PDF to upload on Blackboard: *FirstName_LastName.pdf*

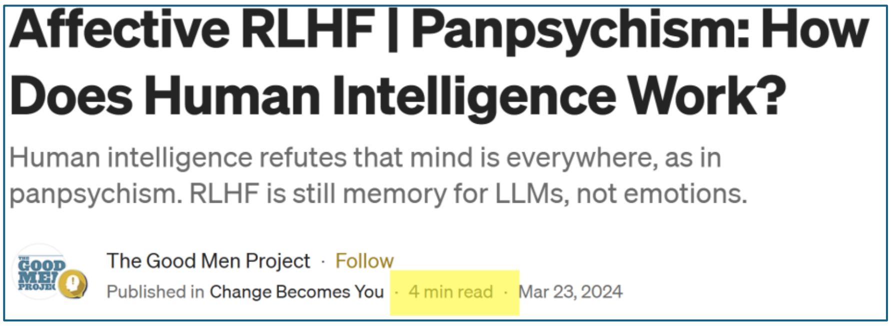
Link to your online article should the first thing at the top of you PDF.

Instructions

The coursework-2 requires that you complete 1 task, that will make up for 50% of the module mark. The other 50% has already been assessed via Coursework-I (programming and report). This coursework offers considerable flexibility, emphasising independent learning, core RL concepts, self-directed research, creativity, and programming skills. It places significant importance on employability and acquiring new competencies, including technical reading and writing about RL principles, familiarising oneself with seminal works in the field, and engaging with insightful blog posts.

Word Limit

There is no word limit, however, your article must not be longer than a ***10 minutes read***. This will show up under the title of your article, for example:



This assignment is an out-of-the-box learning approach that has tangible benefits for your career if done right. It offers you a unique opportunity to not only learn advanced RL methods, coding, and applications but also to build a professional online portfolio. By publishing your tutorial on platforms like Medium.com, you will showcase your expertise in RL to a global audience, demonstrate your ability to understand complex concepts, run and interpret experiments, and establish a credible and visible presence in the academic and professional community.

Additional Information for Completion of Assessment

This is an individual piece of work. Please make sure you have a clear understanding of the grading principles for this component as detailed in the Marking Scheme. If you are unsure about any aspect of this assessment, please seek the advice of a member of the delivery team.

Important Information on Dishonesty & Plagiarism

Plagiarism will be dealt seriously. Students are directed to the college regulations for details of the procedures and penalties involved. For further information, see www.plagiarism.org

Use of Generative AI

For this assignment, you are permitted to use generative AI tools (e.g., Copilot, ChatGPT) to support your work, such as for generating ideas, drafting, or summarising information. However, it is mandatory to acknowledge the use of these tools within your submission.

When acknowledging the use of generative AI tools, please include the following details:

1. The name and version of the AI tool used (e.g., ChatGPT-3.5).
2. The publisher (e.g., OpenAI).
3. The URL of the AI tool.
4. A brief description of how the tool was utilised in your work.
5. A confirmation that the final submission is your own work.

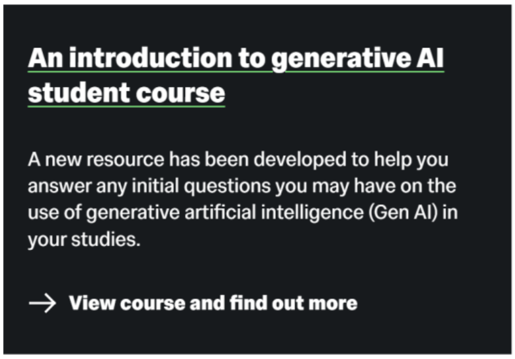
You can find further information on the use of GenAI at this link:
https://www.imperial.ac.uk/staff/educational-development/teaching-toolkit/use-of-generative-ai-for-teaching-learning-and-assessment/

Example Acknowledgment Statement: "*I acknowledge the use of ChatGPT-3.5 (OpenAI, https://chat.openai.com/) to generate an outline for background study. I confirm that no content generated by AI has been presented as my own work*."

Additionally, ensure you adhere to the following guidelines when using AI tools:

- Avoid entering sensitive or confidential data into these systems, as queries may be stored and could contribute to training datasets.
- Do not input personal or private information into AI platforms.

If you are new to Generative AI, the college offers a free course to get you up to speed with it. You can enroll the course by clicking on the following image:



Click here to enroll in the course

Please note that, as of today, none of the most renowned top-tier tutorials, books, papers, lessons, or articles are generated by generative AI (GenAI). They remain the result of human creativity, expertise, and dedicated effort. This assignment, therefore, is not only an opportunity to engage with cutting-edge technology responsibly but also to develop a valuable skill that may not present itself again in the near future. Embrace this as a chance to enhance both your understanding of RL and your ability to critically evaluate the use of these tools in a professional and academic context.

Task Overview

This assignment is designed to help you achieve the core learning objectives of the module, while simultaneously preparing you to apply advanced RL methods in the context of bioengineering. By focusing on advanced RL techniques, this assignment reflects the module's emphasis on theory, implementation, and evaluation, with a strong alignment to real-world bioengineering applications. Your task is to write a high-quality tutorial on one of the selected reinforcement learning methods. The tutorial should explain the chosen method in a clear and engaging way, focusing on its practical implementation and how it could be applied in real-world scientific problems. While you are **NOT** required to write new code, apart from minor changes necessary to produce visuals; you must demonstrate that you understand and can run the method by generating your own plots, graphs, or visualisations. Contextualising the method within a bioeng can earn you up to **5 bonus marks**.

You may choose from any **ONE** of the following topics to write a tutorial, that is easy to follow and help understand the method in simple words. Please note that these topics build upon what we have learned in this module, but vary in complexity, ensuring there is something to learn for students of all abilities and backgrounds. The quality of your tutorial, the challenge it addresses, and how effectively it simplifies the topic will be the key factors in determining a higher mark. There is no limit on how many students can choose to write on the same topic – hence you are in a competition. Here is the list of selected topics:

Human-Level Control with DQN

This method applies DQN to Atari games, demonstrating how deep RL can achieve human-level performance in complex, high-dimensional environments like video games. It leverages DQN and Double DQN concepts for scaling to more complex tasks. Paper and Code

Double Deep Q-Networks (Double DQN)

Double DQN addresses the overestimation bias in DQN by decoupling action selection and action evaluation, improving stability and performance in environments with noisy rewards. It builds on DQN's foundations, refining its accuracy and robustness. Paper and Code

Deep Deterministic Policy Gradient (DDPG)

DDPG is an actor-critic method designed for continuous action spaces, combining deterministic policies with deep function approximation for efficient learning in robotic and control tasks. It transitions from discrete action spaces (DQN) to continuous ones, introducing policy gradient methods. Paper and Code

Asynchronous Advantage Actor-Critic (A3C)

A3C introduces parallel training for actor-critic methods, improving sample efficiency and stability while learning value functions and policies simultaneously. It builds on actor-critic architectures like DDPG and extends them with asynchronous updates for faster convergence. Paper and Code

Addressing Function Approximation Error in Actor-Critic Methods (TD3)

TD3 improves upon DDPG by addressing overestimation bias using twin Q-networks, delayed policy updates, and target smoothing, leading to more stable and robust training. It builds on DDPG, directly addressing its limitations in stability and overestimation. Paper and Code

Proximal Policy Optimisation (PPO)

PPO is an on-policy reinforcement learning algorithm that simplifies trust region policy optimisation, making it more efficient and scalable. It uses clipping and surrogate objectives to stabilise updates, achieving state-of-the-art performance in both discrete and continuous control tasks, such as robotics and high-dimensional simulations. Paper and Code

Soft Actor-Critic (SAC)

SAC is an off-policy actor-critic algorithm that incorporates entropy regularisation to encourage exploration. It refines actor-critic methods like DDPG and A3C by improving exploration and robustness in high-dimensional continuous tasks. Paper and Code

Model-Agnostic Meta-Learning (MAML)

MAML is a meta-learning framework that trains models to adapt quickly to new tasks with minimal data, making it highly versatile for RL and multi-task learning scenarios. It builds upon actor-critic and policy gradient methods, focusing on generalisation across multiple tasks. Paper and Code

The links at the end of each method takes you to a website https://paperswithcode.com/ . It is a powerful resource, hosting over 8,000 machine learning methods with links to trusted implementations and benchmarks. For students, it provides a bridge between theoretical learning and hands-on experimentation, allowing you to test algorithms, debug code, and generate high-quality visualisations efficiently. It also connects you to the global research community and ensures your work aligns with industry standards, making it an invaluable tool for mastering reinforcement learning. You are welcome to watch any youtube videos or tutorials, take inspiration from other people's work that already exists on your selected topic.

Marking scheme

A Simple Introduction (5 Marks)

Begin with a clear and concise overview of the reinforcement learning method. Highlight its significance in solving practical problems, and explain the challenges it addresses.

Theoretical Explanation (10 Marks)

Provide a conceptual explanation of the method, outlining its key mechanisms. Include sufficient mathematical detail to explain its functionality (e.g., loss functions or core algorithm steps), but avoid complex derivations. The focus should remain on accessibility and clarity.

Code Demonstration (5 Marks)

Showcase the method's implementation by focusing on its core logic—what makes it work. Avoid a line-by-line explanation, but ensure key parts of the code are thoroughly explained. Include links to the original code (e.g., GitHub or Papers with Code) to provide transparency and reproducibility.

Results and Visualisation (25 Marks)

Be creative! Use the provided or referenced code to generate original learning curves, heatmaps, or policy visualisations, tables, figures etc. Explain these results clearly, focusing on what they reveal about the method's performance and behaviour. If you are unable to run or understand the entire code, instead of leaving this part blank, do as much as you can and state any limitations. If takes too long – then try to run a more manageable version of the code or look for alternative implementations.

Professional Presentation: 5 Marks

Figures, style, clarity, structure, innovation etc.

Bioengineering Contextualisation (Optional, up to 5 *Bonus* Marks)

Suggest how the RL method could be applied to a bioengineering problem. Examples include optimising robotic prosthetics, designing personalised drug delivery systems, or modelling biomechanics for rehabilitation.

Note: If for ANY reason you ***do not want to publish online***, you can upload just the PDF of your article to the blackboard. The grading criteria will remain the same. However, you may benefit from your online visibility showcasing your work.

Example Articles to get started:

Here are some articles worth checking:

- Deep Q-Networks (DQN): Step-by-Step
- Structured Control Nets for RL
- Reinforcement Learning Tutorial: Cloud Q-Learning
- Our NIPS 2017 Learning to Run Approach
- Convolutional Neural Networks: A Comprehensive Guide

Submission Requirements

Create a PDF version of your tutorial and upload it to Blackboard by 12th December 2024 @3 PM (check Acadiary for any changes). Please do not leave submissions to the last minute to avoid mishaps, as the student office will penalise late submissions.

Publish your tutorial on Medium.com and submit the link in your PDF. All visualisations (e.g., learning curves, Q-value heatmaps) must be original and generated by you. Submit on BB or cite (link) any scripts or notebooks used to generate your results.

Tips for a Good Read

Here are some tips that will be helpful:

Make Your Article Insightful, In-Depth, and Easy to Understand:

Explain the RL method clearly, breaking down complex concepts into manageable parts. Use examples, figures, and simple language to make the content accessible to readers, especially those with a bioengineering background.

Ensure Your Content is Original:

Write in your own words, and add unique insights or examples that connect the RL method to bioengineering applications. Original figures and visualisations will help demonstrate your understanding.

Focus on Clear Messaging:

Ensure your tutorial has a clear narrative. State the significance of the method early on and maintain focus throughout. Avoid unnecessary technical jargon unless it is explained.

Use a Short, Catchy Title and Meaningful Subtitles:

Choose a title that reflects the method and its relevance. Use subtitles to guide the reader logically through the tutorial, such as theoretical explanation, code demonstration, results, and bioengineering context.

Deliver Value to Your Audience:

Think about what other students, professionals, or researchers could gain from your tutorial. Highlight the practical applications of the RL method and provide actionable insights.

Display Code Clearly (if included):

Include only the most important sections of code in your tutorial, focusing on the core functionality of the method. Format the code neatly and provide links to the full implementation for reproducibility.

Fact-Check Thoroughly:

Verify your explanations and results. Ensure your tutorial aligns with the theoretical understanding of the method and that all claims are backed by evidence or references.

Keep Your Conclusion Concise and Focused:

Summarise key findings and discuss the significance of the method. Highlight strengths, limitations, and potential future applications, especially in bioengineering.

Use Precise Tags:

Choose tags that reflect the content of your tutorial, such as "Reinforcement Learning," "Deep Q-Networks," or "Bioengineering Applications," to help your article reach the right audience.

Include Engaging and High-Quality Images:

Create original figures, such as learning curves or heatmaps, using the results of your experiments. Ensure visuals are clear, relevant, and enhance the tutorial’s readability.

Cite Data and Sources Properly:

Reference any external code, papers, or resources used in your work. Crediting your sources adds credibility to your tutorial and demonstrates academic integrity.

Seek Feedback Before Submitting:

Share your draft with peers, friends, or mentors for constructive feedback. This will help refine your narrative, visualisations, and clarity before submission.

Happy writing!