AVD894 – Course project and term paper

1. Introduction

The objective of this course project is for a student to get started with applying RL algorithms and progressing to make some substantial improvements to the state of the art. There are two tracks that a student can take for this course project and term paper.

Track 1: Contributing to reproducible reinforcement learning research and improving the state of the art. Reproducibility of machine learning and reinforcement learning is a big challenge that the research community is facing now. There are actual reproducibility challenges that run such as https://reproml.org/. We will run our own challenge as part of this course. If we do well, then we can submit to such venues as well.

Track 2: Reimplementation of state of the art solutions for public challenges/leaderboards and improving the state of the art. There are multiple challenges/problems that are standard baselines for the reinforcement learning community to study and improve algorithms. We can try contributing to open source code in this track.

Note that improving the state of the art is an important component of the project and some credit (marks) are allocated for just that.

Teams/Formation of teams
The project work has to be done in teams, with four members per team.
It is recommended that each team contains undergraduate and postgraduate students.

Code maintenance
In order to understand best practices for maintaining and working with code across teams, teams have to use Git to maintain version controlled software (with proper git commit history which also needs to be submitted). All teams should maintain a private git repository on Github for this project. The instructor should be given push access to this repository.

Step 0:
Enter details of the team, with a good team name in the circulated Google form (along with additional details).
https://forms.gle/9E5Y32ozqUpEcNvq7

2. Details for Track 1:

The list of papers that have been selected for this reproducibility challenge.

Paper categories:
Communication and communication networks related
1. J. Wigmore, B. Shrader and E. Modiano, "A Novel Switch-Type Policy Network for Resource Allocation Problems," *2025 23rd International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, Linkoping, Sweden, 2025, pp. 1-8, doi: 10.23919/WiOpt66569.2025.11123328.
2. Jali, N., Qu, G., Wang, W. Joshi, G.. (2024). Efficient Reinforcement Learning for Routing Jobs in Heterogeneous Queueing Systems. Proceedings of The 27th International Conference on Artificial Intelligence and Statistics, in Proceedings of Machine Learning Research 238:4177-4185 Available from https://proceedings.mlr.press/v238/jali24a.html.

3. Gu, Zhouyou and She, Changyang and Hardjawana, Wibowo and Lumb, Simon and McKechnie, David and Essery, Todd and Vucetic, Branka, IEEE Journal on Selected Areas in Communications, Knowledge-Assisted Deep Reinforcement Learning in {5G} Scheduler Design: From Theoretical Framework to Implementation, 2021, 39, 7.
4. Z. Ding, R. Schober, and H. V. Poor, No-Pain No-Gain: DRL Assisted Optimization in Energy-Constrained CR-NOMA Networks, IEEE Trans. Communications
5. RFRL Gym: A Reinforcement Learning Testbed for Cognitive Radio Applications, D. Rosen, I. Rochez, C. McIrvin, J. Lee, K. D'Alessandro, M. Wiecek, N. Hoang, R. Saffarini, S. Philips, V. Jones, W. Ivey, Z. Harris-Smart, Z. Harris-Smart, Z. Chin, A. Johnson, A. Jones, W. C. Headley, IEEE International Conference on Machine Learning and Applications (ICMLA), 2023 - *discuss with instructor.
6. L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, *"OpenRAN Gym: AI/ML Development, Data Collection, and Testing for O-RAN on PAWR Platforms,"* Computer Networks, vol. 220, pp. 1-11, January 2023 - *discuss with instructor.

From latest conferences - here is a sample
Any paper from conferences https://github.com/yingchengyang/Reinforcement-Learning-Papers

From indexability/RMAB
1. Killian, J.A., Xu, L., Biswas, A. & Tambe, M.. (2022). Restless and uncertain: Robust policies for restless bandits via deep multi-agent reinforcement learning. Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence in Proceedings of Machine Learning Research 180:990-1000 Available from https://proceedings.mlr.press/v180/killian22a.html.
2. Robledo Relaño, Francisco, et al. "Tabular and deep learning for the whittle index." *ACM Transactions on Modeling and Performance Evaluation of Computing Systems* 9.3 (2024): 1-21.
3. Shisher, Md Kamran Chowdhury, et al. "Online Learning of Whittle Indices for Restless Bandits with Non-Stationary Transition Kernels." *arXiv preprint arXiv:2506.18186* (2025).

3. Details for Track 2:

List of challenge problems with public leaderboards: This year, we will look at https://github.com/openai/gym/wiki/Leaderboard

Problem name	**Details**
LunarLander-v2	The agent must learn a sequence of precise main engine and orientation engine firings to land safely. It requires finding a balance between using fuel (negative reward) and crashing (large negative reward)
Acrobot-v1	A two-link pendulum where only the second joint is actuated. It must swing up to a target height, requiring an unintuitive, periodic swinging motion to build momentum.
DifficultHumanoid-v4	The agent must control a 3D bipedal robot with many degrees of freedom to make it run forward without falling. The state and action spaces are both very large and continuous
Montezuma's Revenge (Atari)	Rewards are extremely sparse; the agent must perform a long, specific sequence of actions (like jumping over an object to get a key) before

	receiving the first positive reward, which is a significant exploration challenge.
DifficultCarRacing-v2	A blend of challenges—the agent must learn to drive a car on a procedurally generated track from raw pixel observations (image input) while issuing continuous steering and acceleration actions. This combines the complexity of high-dimensional state space with continuous actions.

4. Project execution:

Step 1:

Identifying the paper that you are going to reproduce from Track 1 or the problem for Track 2. The details of this choice should be given in the circulated Google form. The details of the paper (i.e., citation, journal/conference information) in the following spreadsheet along with your SC Code and name. Bonus: Understand how to manage references using tools such as Zotero and its use with tools such as LaTeX using bibtex.
https://forms.gle/9E5Y32ozqUpEcNvq7

Step 2:

Read the paper or understand the problem. When reading the paper, right now you have to give attention to the following:
1. What exactly is the problem? Can you state the problem (or a simplified version of the problem) mathematically (as an optimization problem)?
2. How is the problem posed as an RL problem, i.e., a problem in which we try to control a system evolving in time?
3. We studied state space, action space, evolution equation, reward in class – what are the equivalents in the paper/problem.

Task 1:

1. Preparation of a short report on the problem statement. Explain what is the problem solved in the paper. And explain how a simpified version of the problem can be posed as a discrete-state discrete-action-space Markov Decision Process. This simplification can be obtained, for example by assuming that some elements of the problem setting are known, by discretizing the state space, reducing dimensions etc.
2. Solving the MDP in (1) above using value iteration or policy iteration (implemented in Python).
3. At the end of Task 1, the students have to submit the report and associated code for evaluation.

Task 2:

In Task 2, the students have to implement or set up a complete simulator (in Python) of the system that is being controlled or optimized via Reinforcement Learning in the chosen track. This simulator should be able to produce an output run or sample path of the system under simple policies – such as a policy that just randomly choses actions. At the end of Task 2, students have to submit a report which summarizes how the simulator was implemented as well as the associated code.

For both tasks, discussing with the instructor, your peers, and your seniors is important. Please do that. However, please make sure that the report and code are written on your own (of course, team members are allowed to do this collaboratively).

5. Final submission and evaluation

Deliverable 1:

Final report/term paper on the research paper/problem

- The term paper should be prepared in a professional and academic manner using Latex. For this paper, we will use a standard style file which has been used for NeurIPS conference. Please see this link for details regarding the style file.
  https://neurips.cc/Conferences/2024/CallForPapers
- Each team needs to submit a paper containing details of their work to Moodle.
- The term paper should have a maximum of 6 pages in the above standard format.
- The term paper should be written in your own words
  - The paper would be screened for plagiarism as well as for AI generated content
- The term paper should have the following sections and outline
  - Title with authors
  - Abstract
    - Should summarize the problem considered in the paper
    - Why is the problem significant or interesting to the world and you
    - What has been done in the paper – what are the significant contributions of the paper
    - What have you done in the project?
  - Introduction
    - Motivation for the problem considered in the paper
    - Brief summary of the problem considered in the paper
    - What are comparable works in literature? Has this problem been considered before? What has been done before? Please point out/cite these papers (use bibtex for the citations)
  - Model and problem statement
    - What is the environment?
      - What is a simplified model of the environment – the MDP model from the last set of tasks can be written down here?
      - Specify states, actions, rewards
    - How is performance evaluated?
      - Discuss the optimization criterion
      - Why do you think this optimization criterion is appropriate for the system considered or the goal that is being considered?
  - Reinforcement learning agent
    - Discussion about the reinforcement learning agent/method followed in the paper
    - Discussion about your implementation of the agent
      - If you have used an already existing implementation of the agent – please specify that here – proper credit/references should be given.
  - Extensions (this is important, please see evaluation scheme)
    - Can think about different ways to improve performance or even extend to other related problems.
    - Are there other reinforcement learning agents that can be tried out on this problem?
      - Discuss any other reinforcement learning agents that you have come across
      - You can look at standard reinforcement learning implementations such as StableBaselines, CleanRL (https://github.com/vwxyzjn/cleanrl), Tianshou (https://tianshou.org/en/stable/) etc for ideas
    - See if any other algorithms can be reused (or reimplemented) and interfaced with your environment for control. You can also think of non-RL algorithms! (maybe even evolutionary algorithms for optimization).
    - Please note that assumptions of extra information (which can unfairly help RL agents) need to be justified very strongly.
  - Performance analysis and comparison
    - Re-generation of results from the paper and comparison with the results in the paper (here a screenshot of the results in the paper can be compared with whatever results

        you have obtained – note that due to computational constraints it may not be possible to obtain the exact results).
    - Comparison of the results obtained for different RL agents
    - These results should ideally be reported using figures – figures need to be prepared in a professional manner – for example, use vector graphics format, use legible and large enough font sizes, axes and plotlines should be properly labelled, legends should be available etc.
  - Challenges
    - Enumerate any challenges that you had faced during the project as well as ideas for future work and extensions – you can think along the dimensions of improvements in the system model, the problem (either more realistic or more fundamental), improvements in the approach and algorithms.
  - References (use bibtex)

Deliverable 2:
Project Presentations

- Each team has to make a presentation for 10 minutes – team members should take turns to present – team members can be asked to present specific parts of the presentation – so everyone should be comfortable with the whole presentation.
- Presentation time will be strictly enforced.
- The presentation slides need to be submitted to Moodle (can be done after the presentation).
- Slides can be prepared using any software – Latex (Beamer) is preferred. If using Latex the following stylefile is preferable
  https://github.com/4sarathbabu/IISTSlides
- The slides can have the following outline
  - Title with authors (1 slide)
  - Introduction (max 2 slides)
    - Motivation for the problem considered in the paper
    - Brief summary of the problem considered in the paper
    - Table summarizing relevant papers
  - Model and problem statement (max 2 slides)
    - Environment description – describe using a picture
    - Performance criterion
  - Reinforcement learning agent (max 2 slides)
    - Describe the agent – using a picture/animations etc is desirable
  - Extensions (max 2 slides)
    - Summarize any extensions
  - Performance analysis and comparison
    - 1 slide per figure – should explain the figure on the slides itself
  - Challenges/Ideas/Future work (max 1 slide)
    - Enumerate ideas
  - References – usually in presentations it is better to include references as footnotes (say using the footcite package in latex)

Deliverable 3:
Code - hosted on a Github repository.

6. Mark distribution for the course project and term paper:
The total marks allocated for this course project and term paper is 60 marks.

5 marks upto and including Task 1
5 marks for Task 2

10 marks for code development, with proper git commit history:
You should have a proper git commit history showing how the development of code has been done. There will be credit for properly maintained code.

10 marks for associated presentation:
The presentation should not be a dump of information – but should clearly discuss the problem and fundamental issues/challenges associated with the solution.

20 marks for Term paper: The term paper would be evaluated on the basis of its technical content as well as its presentation (because one of the objectives is to make students understand how to write and polish a technical publication for submission to top venues in any research field). A good example to follow would be to look at papers appearing in top venues such as IEEE Transactions. The quality of figures, quality of writing, quality of organization all matters here.

10 marks for Improvement over the state of the art: If a claim of improvement over SOTA is made by a team, then it should be carefully justified by one or more means
1. Mathematical proof – using clever modelling or reasonable approximations
2. Extensive simulations – note that any claim which is based on comparing "means" or averages is not really acceptable. It should be supported by confidence intervals, uncertainity intervals.
Note that hyperparameter optimization as a way to improve performance over state of the art wouldn't normally be accepted as an "improvement" – unless there is some special way in which the hyperparameter optimization was done.

Also, such claims should be clearly justified during a one-on-one meeting with the instructor, which will drill down to the methods used in the paper, fundamental assumptions, demonstration of the method and so on.

Please note that it is your responsibility to maintain proper backups of code and other material for your project.

7. Deadlines:

Upto and including Task 1	October 18
Upto and including Task 2	November 1
Deliverable 1 and 3	November 21
Deliverable 2	November 24, 25, 26