Western Engineering

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

AISE 4030 Reinforcement Learning

COURSE DESCRIPTION: This course provides a comprehensive study of reinforcement learning (RL), emphasizing foundational principles, algorithmic strategies, and applications relevant to software engineering. Topics include Monte Carlo and Temporal-Difference methods, Deep Q-Networks (DQN), policy-gradient approaches, actor-critic frameworks, experience replay mechanisms, multi-agent reinforcement learning, and apprenticeship learning through inverse reinforcement learning. Students will develop theoretical understanding and practical knowledge, enabling them to effectively design, evaluate, and apply reinforcement learning techniques in diverse software engineering contexts.

ACADEMIC CALENDAR:

https://www.westerncalendar.uwo.ca/Courses.cfm?CourseAcadCalendarID=MAIN_031457_1&SelectedCalendar=Live&ArchiveID=

Reinforcement Learning (RL) is a field of AI where agents learn to make decisions through trial-and-error to maximize rewards. This course covers core RL concepts, algorithms, and applications, including Q-Learning, Deep RL, and policy-based methods, with hands-on experience to solve real-world problems in robotics, games, and recommendation systems.

PRE OR COREQUISITES:

Prerequisite(s):

- Data Science 3000A/B and (NMM 2276A/B or NMM 2277A/B).

Anti-Requisite(s):

- CS 9670/9671/9170.

Unless you have either the requisites for this course or written special permission from your Dean to enroll in it, you will be removed from this course and it will be deleted from your record.

CEAB ACADEMIC UNITS: Engineering Science 75%, Engineering Design 25%

CONTACT HOURS:

Timetable information is available at https://draftmyschedule.uwo.ca/.

LECTURE:	- 3 hours weekly

TEXTBOOK:

- Reinforcement Learning: An Introduction, 2nd Edition, Richard Sutton & Andrew Barto COST $145.00

RECOMMENDED REFERENCES:

- Lecture Slides
- PyTorch Documentation (https://docs.pytorch.org/docs/stable/index.html)
- CS 285 Deep Reinforcement Learning – UC Berkeley (https://rail.eecs.berkeley.edu/deeprlcourse/)
- Spinning Up in Deep RL – OpenAI (https://spinningup.openai.com/en/latest/)
- Gymnasium – formerly OpenAI Gym (https://gymnasium.farama.org/index.html)

RECOMMENDED SOFTWARE:

- Programming Language: Python
- Libraries: PyTorch, TensorFlow, Matplotlib, Seaborn, Scikit-Learn, OpenCV
- IDE: Microsoft VS Code, Jupyter Notebook

GENERAL LEARNING OBJECTIVES (CEAB GRADUATE ATTRIBUTES):

Knowledge Base	A	**Engineering Tools**	A	**Impact on Society**	
Problem Analysis		**Individual & Team Work**		**Ethics and Equity**	
Investigation	A	**Communication**		**Economics and Project Management**	
Design		**Professionalism**		**Life-Long Learning**	

Notation¿*I: Introductory, D: Developed, A: Applied, or blank*. I – The instructor will introduce the topic at the level required. It is not necessary for the student to have seen the material before. D – There may be a reminder or review, but the student is expected to have seen and been tested on the material before taking the course. A – It is expected that the student can apply the knowledge without prompting (e.g. no review).

COURSE MATERIALS:

Weekly content and guides for the laboratories will be available on the course OWL site. The material for this course will be taught in both lectures and labs; therefore, it is imperative that you attend each lecture and lab.

COURSE TOPICS AND SPECIFIC LEARNING OUTCOMES:

The following table summarizes the course learning outcomes along with CEAB GAIs where the GAIs in bold indicate ones to be measured and reported annually.

Course Objectives and Specific Learning Outcomes	CEAB Graduate Attributes Indicators	Tentative Timeline
Chapter 1: Foundations of Reinforcement Learning **At the end of this unit, the students will be able to:**		
a. Define reinforcement learning and explain how it differs from supervised and unsupervised learning paradigms.	**KB3**	Week 1
b. Identify and describe the core RL elements: agent, environment, state, action, and reward within a real-world software-engineering scenario.		
c. Differentiate episodic from continuing tasks and compute a discounted return G_t given a sequence of rewards and a discount factor γ.		
d. Explain the exploration/exploitation dilemma and compare basic exploration schedules such as fixed and decaying ε-greedy strategies.		
Chapter 2: Learning Strategies: Monte Carlo vs. Temporal-Difference **At the end of this unit, the students will be able to:**		
a. Contrast first-visit and every-visit Monte Carlo methods with TD(0) bootstrapping, identifying when each is appropriate.	**KB3, ET3**	Week 2
b. Derive and apply the tabular TD(0) update rule to estimate $V(s)$		
c. Compute and interpret bias-variance trade-offs between MC and TD estimates.		
d. Implement a small Grid-World agent that learns with both MC and TD and visualize convergence behaviour.		
Chapter 3: Tabular Control: On-Policy vs. Off-Policy TD (SARSA & Q-learning) **At the end of this unit, the students will be able to:**		
a. Define behavior policy and target policy, explaining their roles in on-policy and off-policy learning.	**KB3, ET3**	Week 3
b. Implement SARSA for on-policy control and tabular Q-learning for off-policy control in the same environment.		
c. Identify scenarios where on-policy methods are preferred over off-policy (and vice versa) in software-engineering applications.		
d. Evaluate the learning curves of SARSA and Q-learning, discussing the impact of policy mismatch.		
e. Evaluate various exploration techniques, including decaying -greedy, Softmax, and Upper Confidence Bound (UCB).		

Chapter 4: Introduction to Deep Q-Networks (DQN) **At the end of this section, the students will be able to:**		
a. Explain why function approximation is necessary for high-dimensional state spaces and describe how neural networks serve as Q-function approximators.	**KB3, ET2**	Week 4
b. Outline the architecture of a basic DQN, including input pre-processing, convolutional layers (for image states), and fully connected output heads.		
c. Compare tabular Q-learning and DQN in terms of scalability, generalization, and stability challenges.		
d. Train a simple DQN on Atari Game environment.		
Chapter 5: Experience Replay & Replay Buffers **At the end of this unit, the students will be able to:**		
a. Describe the motivation for experience replay and its role in breaking temporal correlations.	**KB3, ET2, I1**	Week 5
b. Implement a uniform replay buffer and integrate it into an existing DQN agent.		
c. Explain prioritized experience replay (PER) and construct a segment-tree-based sampler.		
d. Analyze how buffer size, sampling strategy, and update frequency affect sample efficiency and convergence.		
Chapter 6: Advanced DQN Variants **At the end of this unit, the students will be able to:**		
a. Identify the over-estimation problem in standard DQN and explain how Double DQN mitigates it.	**KB3, ET2**	Week 6
b. Describe the duelling-network architecture and justify separating state-value and advantage streams.		
c. Identify various DQN variants (e.g., C51-DQN, Noisy DQN, and Rainbow DQN).		
d. Modify a baseline DQN codebase to incorporate Double and Duelling enhancements and report performance gains.		
Reading Week – Week 7		
Chapter 7: Policy-Gradient Fundamentals **At the end of this unit, the students will be able to:**		
a. Explain why value-based methods struggle with continuous or large discrete action spaces.	**KB3, ET2**	Week 8
b. Derive the REINFORCE policy-gradient formula and implement a vanilla policy-gradient agent.		
c. Define and compute baseline and advantage functions to reduce gradient-estimate variance.		
d. Train an agent to solve a continuous action environment (e.g. Pendulum).		
Chapter 8: Modern Policy-Based Algorithms		
a. Analyze the instability issues of vanilla policy gradients, specifically step-size sensitivity and the risk of policy collapse.	**KB3, ET2**	Week 9

b. Explain the concept of "Trust Regions" and the role of KL Divergence in constraining policy updates (TRPO)		
c. Construct the Proximal Policy Optimization (PPO) clipped surrogate objective to enforce stable updates without complex second-order constraints.		
d. Implement a PPO agent and evaluate its stability and sample efficiency compared to REINFORCE.		
Chapter 9: Actor-Critic Algorithms **At the end of this unit, the students will be able to:**		
a. Describe the actor–critic framework and delineate the responsibilities of actor and critic networks.	**KB3, ET2**	Week 10
b. Implement deterministic policy gradients and distinguish between DDPG and its improved variant TD3.		
c. Understand Soft Actor-Critic and the role of Maximum Entropy in encouraging robust exploration.		
d. Assess actor-critic vs. value-only methods in terms of sample efficiency and stability.		
Chapter 10: Multi-Agent Reinforcement Learning (MARL) **At the end of this unit, the students will be able to:**		
a. Articulate why learning becomes non-stationary when multiple agents adapt simultaneously.	**KB3**	Week 11
b. Contrast independent Q-learning with centralized-training, decentralized-execution frameworks.		
c. Discuss a simple cooperative MADDPG.		
Chapter 11: Discussions on Advanced RL Topics **At the end of this unit, the students will be able to:**		
a. Differentiate model-based and model-free paradigms in terms of application scenarios.	**KB3**	Week 12
b. Differentiate between MDP and Partially Observable MDP (POMDP)		
c. Inverse Reinforcement Learning (IRL): Explain the Prisoner's Dilemma, and why observed expert behaviour may not reveal the underlying reward.		
d. Define Hierarchical RL and its applications		
Week 13 – Project Presentation & Discussion		

EVALUATION:

Name	% Worth	Assigned	Due Date	CEAB GAs ASSESSED
Assignment 1	5%	-	Week 4	**ET2, ET3, I1**
Assignment 2	5 %	-	Week 7	**ET2, ET3, I1**
Final Project	20%	-	Week 13	**I3, ET2, ET3**
Midterm Exam	30%	Yes	After Reading Week	**KB3, I1**
Final Exam	40%	Yes	TBA	**KB3, ET2, ET3, I1**

Note that the dates listed above are **tentative** and may be adjusted if needed. Marks will be assigned on the basis of method of analysis and presentation, correctness of solution, clarity and neatness.

For this course, the following assessment has been designated as requiring supporting documentation:

- Midterm Exam
- Final Exam

COURSE POLICIES:

Late Submission Policy:
Please note that the assignment submission deadline includes flexibility in the form of a 48-hour submission window (grace period). As a result, the instructor reserves the right to deny any requests for academic consideration for assignments submitted after this grace period.
If students submit their assignments after the deadline, a penalty of 10% per day will be applied for late submissions, up to a maximum of 3 days. After three days, late submissions will no longer be accepted.

Self-Reported Absence:
Self-reported absence will not extend the assignment deadline as the deadline includes 48-hour grace period.

Midterm Test:
There will be one midterm test, which will be a closed-book exam (no reference materials allowed) and will last for two hours. Calculators are not permitted. If a student misses the midterm, the exam will not be rescheduled. Instead, the weight of the midterm will be added to the final exam, making the final exam worth 70% of the overall grade. If no valid justification is provided for missing the midterm, the student will receive a mark of zero for the test.

Final Examination:
Please note that the final exam is considered to be central to the learning objectives for this course. Accordingly, students seeking academic consideration for this assessment must provide formal supporting documentation. Students who are granted academic consideration for this assessment will be provided with the following opportunity to make up this work: The final examination will take place during the regular examination period. It will be three hours long, closed book, and no calculators are allowed.

Passing Grade
A mark of 50% or more must be achieved on the midterm and final examination to obtain a passing grade in the course. An examination mark < 50% will result in a final course grade of 48% or less.
If the above conditions are not met, your final grade cannot exceed 48%. Students who have failed this course (i.e., final average < 50%) must repeat all course components.

Use of English:
In accordance with Senate and Faculty Policy, students may be penalized up to 10% of the marks on all assignments, tests, and examinations for improper use of English. Additionally, poorly written work with the exception of the final examination may be returned without grading. If resubmission of the work is permitted, it may be graded with marks deducted for poor English and/or late submission.

Attendance:
Attendance will be recorded during every lecture using **iClicker**. Students are expected to maintain a cumulative attendance rate of at least **70%** throughout the semester. Students who fall below this threshold will be reported to the Dean's Office. Consistent with university regulations, any student who fails to meet this attendance requirement (after due warning has been given) may, upon the recommendation of the department and with the permission of the Dean, be disqualified from writing the final examination.