

Degree programme	Type	Course
Bachelor in Artificial Intelligence	OB	3

Contact lecturer

Name : Jordi Casas Roma

Email : jordi.casas.roma@uab.cat

Prerequisites

Students are expected to have successfully completed the courses "Fundamentals of Machine Learning" and "Neural Networks and Deep Learning", as these provide the essential background knowledge required for this subject.

Group languages

You can consult this information at the [end](#) of the document.

Objectives

The aim of this course is to provide comprehensive training in the fundamental methodologies, techniques, and algorithms of reinforcement learning.

Throughout the course, students will study, implement, and apply a broad range of reinforcement learning methods, covering both classical and modern approaches. They will develop a solid understanding of the underlying theoretical concepts while gaining practical experience through hands-on implementation.

Students will build key algorithms from scratch to deepen their understanding, explore and use state-of-the-art reinforcement learning libraries, and apply these techniques to solve real-world decision-making and sequential learning problems.

Learning outcomes

- CM07 (Integrate machine learning-based solutions into the design, deployment, and deployment of AI applications.) Integrate machine learning-based solutions into the design, deployment, and deployment of AI applications.
- KM23 (Explain the principles and mathematical formulation on which machine learning algorithms are based.) Explain the principles and mathematical formulation on which machine learning algorithms are based.
- KM24 (Identify the advantages and disadvantages of different learning algorithms and paradigms in their application to learning problems.) Identify the advantages and disadvantages of different learning algorithms and paradigms in their application to learning problems.
- SM25 (Critically and reasoned analysis of the results of the application of a machine learning model according to the requirements of the problem.) Critically and reasoned analysis of the results of the application of a machine learning model according to the requirements of the problem.
- SM26 (Select the appropriate paradigms, algorithms, and architectures for a learning problem, based on the domain and type of problem, the available data, and any other constraints or requirements.) Select the appropriate paradigms, algorithms, and architectures for a learning problem, based on the domain and type of problem, the available data, and any other constraints or requirements.

- SM27 (Efficiently and optimally use the computing resources needed for training and validating learning models.) Efficiently and optimally use the computing resources needed for training and validating learning models.

Contents

1. Introduction and Gymnasium

- Introduction to RL and Gymnasium library.

2. Tabular solutions

- Markov Decision Process
- Dynamic Programming
- Monte Carlo
- TD learning

3. Approximate solutions

- Deep Q-Networks
- Policy Gradients
- Actor-Critic

4. Multi-Agent Reinforcement Learning

- Game models
- Solution concepts
- MARL in games

5. Real-world problems

- Environment design and implementation

Learning activities and methodology

Title	Hours	ECTS	Learning outcomes
Project	50	2	CM07, KM24, SM25, SM26, SM27
Evaluable practical activities	42	1.68	CM07, SM26, SM27
Non-evaluative practical activities	21	0.84	CM07, SM26, SM27
Theory	21	0.84	KM23, KM24, SM25

The course consists of four main types of learning activities:

- Theory sessions
- Evaluable practical exercises
- Theory tests
- Project

Theory sessions

These sessions introduce the theoretical foundations of reinforcement learning. For each topic, the main concepts, mathematical formulations, and corresponding algorithms are presented and discussed. To reinforce the material, non-evaluative practical exercises may be proposed during the lectures. These sessions will also include the guided resolution of practical problems and programming exercises using Python.

Evaluable practical exercises

Throughout the semester, students will complete several individual practical assignments designed to assess their understanding of the course contents. These exercises will involve implementing and applying reinforcement learning algorithms to solve a variety of problems.

Theory tests

Students will complete individual theory tests through Moodle during the semester. These tests are intended to assess their understanding of the theoretical concepts covered in the lectures.

Project

During the semester, students will complete a group project involving the solution of a reinforcement learning problem of moderate complexity. Projects will be carried out in teams of two or three students.

Each team will remain unchanged throughout the project and will be responsible for self-managing all aspects of the work, including task allocation, planning, resource management, and conflict resolution. Teams are expected to work autonomously while applying the concepts and techniques learned in the course.

The above activities will be complemented by office hours and tutorial sessions, where students may seek guidance and ask questions outside the scheduled class hours.

All course materials, announcements, assignment descriptions, and other relevant documentation will be made available through the Virtual Campus (cv.uab.cat).

Annotation: within the schedule set by the centre or degree programme, 15 minutes of one class will be reserved for students to evaluate their lecturers and their courses or modules through questionnaires.

Assessment

Continuous assessment activities

Title	Weight	Hours	ECTS	Learning outcomes
Theory test	10%	2	0.08	KM23, KM24
Practical activities	20%	4	0.16	CM07, SM25, SM26, SM27
Individual exams	40%	4	0.16	KM23, KM24
Project	30%	6	0.24	CM07, KM24, SM25, SM26, SM27

The assessment system is designed to evaluate both the students' understanding of the theoretical foundations of reinforcement learning and their ability to apply this knowledge to practical problems.

Final Grade

The final grade is computed as follows:

$$\text{Final Grade} = (0.40 \times \text{Theory Exams}) + (0.10 \times \text{Theory Tests}) + (0.20 \times \text{Practical Activities}) + (0.30 \times \text{Project})$$

This formula is applied only if both of the following conditions are satisfied:

- Theory Exams ≥ 5.0
- Project ≥ 5.0

If the weighted average is 5.0 or higher, but the minimum required grade is not achieved in either the Theory Exams or the Project, the final course grade will be 4.5 (Fail).

1. Theory Exams (40%)

The theory exams assess each student's individual understanding of the theoretical concepts covered in the course. Assessment is carried out through two written examinations:

$$\text{Theory Exams} = (0.50 \times \text{Exam 1}) + (0.50 \times \text{Exam 2})$$

- Exam 1 (Midterm): Covers the first part of the course and is held approximately halfway through the semester.
- Exam 2 (Final): Covers the remaining course contents and is held at the end of the semester.

To pass the Theory Exams component, both Exam 1 and Exam 2 must have a minimum grade of 5.0.

Students who do not satisfy these requirements may take a resit examination, which will assess only the part(s) of the course that were not passed (Exam 1, Exam 2, or both).

2. Theory Tests (10%)

Theory tests are individual Moodle-based assessments conducted throughout the semester. Their purpose is to reinforce and evaluate students' understanding of the theoretical concepts presented in class.

The Theory Tests grade is computed as the arithmetic mean of all theory tests completed during the course.

3. Practical Activities (20%)

Practical activities assess students' ability to implement and apply the reinforcement learning techniques studied in the course.

Students will individually complete several programming assignments throughout the semester.

The Practical Activities grade is computed as the arithmetic mean of all practical assignments.

4. Project (30%)

The project is designed to assess students' ability to integrate the knowledge acquired throughout the course to solve a reinforcement learning problem of moderate complexity.

Projects are completed in teams of two or three students and are assessed through:

- a written project deliverable; and
- an oral presentation.

The project grade is computed as follows:

$$\text{Project} = (0.70 \times \text{Deliverables}) + (0.30 \times \text{Oral presentation})$$

Active participation in both the written deliverable and the oral presentation is mandatory. If a student does not participate in any of these compulsory activities, the maximum project grade will be 4.5, regardless of the weighted average.

Projects are not recoverable. Students who fail to submit the project deliverable or whose final project grade is below 5.0 will not be able to pass the course.

Bibliography

- Reinforcement Learning: An Introduction (Second edition). R. S. Sutton, A. G. Barto, MIT Press, Cambridge, MA, 2018.
- Deep Reinforcement Learning Hands-On (Third edition). M. Lapan, Packt Publishing, 2024

Software

The practical activities in this course will be developed using Python and its scientific computing and machine learning ecosystem. Students will work with widely used libraries, including NumPy, Matplotlib, pandas, and scikit-learn, as well as deep learning frameworks such as PyTorch and TensorFlow. Reinforcement learning exercises and projects will be implemented using the Gymnasium framework and the Stable-Baselines3 library.

Course groups and languages

The information provided is provisional until November 30. After this date, you will be able to consult the language of each group through this [link](#). To access the information, you will need to enter the course CODE

Type of teaching	Group	Language	Semester	Shift
(TE) Theory	1	English	second semester	afternoon
(PAUL) Classroom practices	1	English	second semester	afternoon
(PLAB) Practical laboratories	1	English	second semester	afternoon
(PLAB) Practical laboratories	2	English	first semester	afternoon