DISCIPLINE SPECIFIC ELECTIVE COURSE: Reinforcement Learning

Credit distribution, Eligibility, and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSE7c: Reinforcement Learning	4	3	1	0	Pass in Class XII	Machine Learning/Artificial Intelligence

Learning Objectives

The objectives of this course are:

1. to prepare students to visualize reinforcement learning problems
2. to introduce students to the concepts based on Markov Decision Process, Dynamic Programming, Monte Carlo methods, and Temporal-Difference learning.
3. recognize current advanced techniques and applications in Reinforcement Learning

Learning Outcomes

On successful completion of the course, students will be able to:

1. learn Reinforcement Learning task formulations and the core principles behind Reinforcement Learning.
2. work on problem-solving techniques based on Dynamic Programming, Monte Carlo, and Temporal-Difference.
3. implement in code common algorithms following code standards and libraries used in Reinforcement Learning.
4. learn the policy gradient methods from vanilla to relatively complex cases.

Syllabus

Unit 1 Introduction: Historical perspective of Reinforcement Learning (RL), Basics of RL: definition, how reinforcement learning happens, examples, terminology, notation, and assumptions, Elements of RL: polices, value function, reward Functions and Bellman Equation, different techniques for solving RL problem, Code Standards and Libraries used in RL using Python/Keras/TensorFlow/MATLAB.

Unit 2 Markov Decision Process (MDP) and Dynamic Programming (DP): Markov property, Introduction to Markov decision process (MDP), creating MDPs, goals and rewards, returns and episodes, optimality of value functions and policies, Bellman optimality equations. Overview of dynamic programming for MDP, principle of optimality, iterative policy evaluation, Policy Improvement, policy iteration, value iteration, generalized policy iteration, Asynchronous DP, Efficiency of DP.

Unit 3 Monte Carlo (MC) Methods: Monte Carlo methods (First visit and every visit Monte Carlo), Monte Carlo control, On policy and off policy learning, Importance sampling.

Unit 4 Temporal Difference (TD) Learning: Temporal-Difference learning methods - TD (0), SARSA, Q-Learning and their variants. Markov reward process (MRP), Overview of TD (1) and TD(λ).

Unit 5 Approximation Methods and Policy Gradient: Function approximation methods (Gradient MC and Semi-gradient TD (0) algorithms), Eligibility traces, After-states, Least squares TD. Policy Approximation and its advantages, Naive REINFORCE algorithm, bias and variance in Reinforcement Learning, Reducing variance in policy gradient estimates, baselines, advantage function, actor-critic methods, an introduction to Deep Reinforcement Learning

References

1. Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction* 2nd Edition, MIT Press, 2018.
2. Enes Bilgin *Mastering Reinforcement Learning with Python: Build next-generation, self-learning models using reinforcement learning techniques and best practices***,** 1st edition, Packt Publishing, 2020.

Additional References

(i) Phil Winder *Reinforcement Learning: Industrial Applications of Intelligent Agents*, O'Reilly Media, 2020.
(ii) Alexander Zai, Brandon Brown *Deep Reinforcement Learning in Action*, 1st edition, Manning Publications, 2020.

Suggested Practical List

Implement the following exercises using Python/Keras/TensorFlow/MATLAB.

1. Dynamic Programming Policy Evaluation algorithm.
2. Dynamic Programming Policy Iteration algorithm.
3. Dynamic Programming Value Iteration algorithm.
4. Monte Carlo Prediction
5. Off-Policy Monte Carlo Control with Importance Sampling
6. SARSA On policy TD learning algorithm
7. Q-learning OFF policy TD learning algorithm.
8. Policy Gradient REINFORCE algorithm
9. Policy Gradient Actor-Critic method algorithm

**For exercises 1 to 7, consider the following environments for testing: GridWorld, Blackjack, WindyGridWorld*

**For exercises 8 onward, consider the following environments for testing: CartPole, CartPoleRaw*