

Reinforcement Learning Programming (CSCN8020)

INTRODUCTION TO REINFORCEMENT LEARNING


A solid blue horizontal bar spanning the width of the slide, located at the bottom.

Content

Course Information

- Instructor Introduction
- Course Objectives & Content
- Course Grading

Introduction to Reinforcement Learning

- What is Reinforcement Learning (RL)?
 - Key RL Terminology
 - RL vs other machine learning paradigms
 - The RL Problem
 - RL Applications
 - Introduction to Markov Decision Process
- 

Instructor

Name: Mahmoud Nasr

Email: mmohamed@conestogac.on.ca

Office Hours: Set by email



Course Objectives


Course Description

This course delves into the principles of reinforcement learning (RL) to equip students with the knowledge and skills needed to model and solve problems using RL and Deep RL techniques. Unlike traditional supervised and unsupervised learning methods, RL thrives in environments where learning is driven by interaction and delayed feedback.

Course Objectives

- **Theory and Application:** Students will develop a strong understanding of RL theory while gaining practical experience in coding RL algorithms.
- **Problem Domains:** The course covers the application of RL techniques to various real-world domains, including robotics, autonomous systems, financial technology, and gaming.
- **Ethical Considerations:** Ethical implications of RL in different contexts will be examined, enabling students to understand responsible AI deployment.
- **Capstone Project:** Students will demonstrate their RL proficiency through a final project that showcases their ability to solve practical problems using RL methods.

This academic exploration of RL equips students with the tools to enhance their problem-solving abilities in interactive and dynamic environments.



Course Outline

Week 1: Introduction to RL

Week 2: Markov Decision Process (MDP)

Week 3: Dynamic Programming for RL

Week 4: Monte Carlo Algorithms

Week 5: Temporal Difference Learning

Week 6: Exploration Strategies

Week 7: Deep Learning Fundamentals

Week 8: Student Success Week

Week 9: Deep Q-Networks (DQN)

Week 10: Policy Gradient Method

Week 11: Actor-Critic Method

Week 12: Advanced Topics in RL

Week 13: Advanced Topics in RL-II + RL Ethics

Week 14: Project Delivery and Presentation

Week 15: Final Project Discussion

Course Grading

Item	Percentage
Quizzes (12)	10%
Assignments	45%
➤ Assignment 1	➤ 15%
➤ Assignment 2	➤ 15%
➤ Assignment 3	➤ 15%
Project	45%
➤ Proposal	➤ 5%
➤ Progress Report	➤ 5%
➤ Presentation	➤ 10%
➤ Final Report	➤ 25%
Total	100%

What is Reinforcement Learning?

Definition 1

Reinforcement Learning is learning what to do-how to map situations to actions-so as to maximize a numerical reward signal.

Definition 2

Reinforcement Learning (RL) is a type of machine learning paradigm in which an agent learns to make a sequence of decisions by interacting with an environment. The agent receives feedback in the form of rewards or penalties, enabling it to learn optimal strategies over time.

- Learning by trial-and-error with a delayed reward
- Actions are taken sequentially, current decisions affect future outcomes

Key Components of RL

Environment

The external system with which the agent interacts, providing feedback in the form of rewards

Agent

The learner and decision-maker that interacts with the environment

State (s)

Full representation of the environment

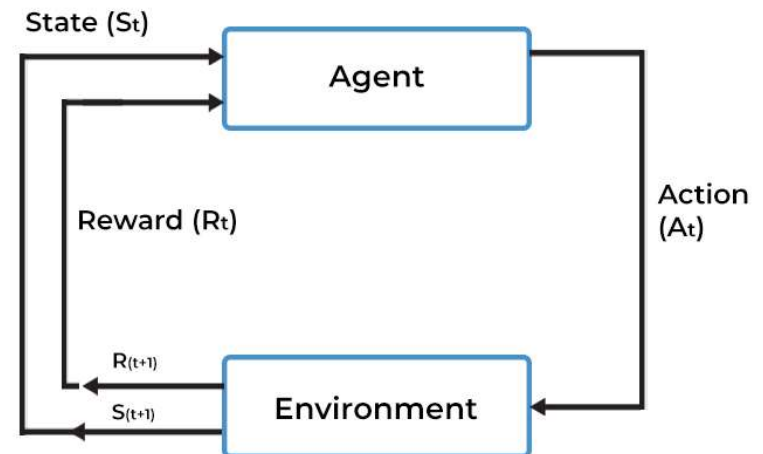
Action (a)

Decision taken by agent at a given state to transition it to another state

Reward (r)

A numerical representation that quantifies the immediate feedback after the agent takes an action at a state


REINFORCEMENT LEARNING MODEL



Comparison of Learning Paradigms

LEARNING PARADIGM	HOW THEY LEARN	INTERACTION WITH ENVIRONMENT	FEEDBACK MECHANISM
Supervised Learning	Learns from labeled data with input-output pairs	Passive: Receives predefined labels	Direct, Explicit
Unsupervised Learning	Learns patterns and structures in unlabeled data	Passive: Learns patterns from data	Indirect, Intrinsic Structures
Reinforcement Learning	Learns from interaction with the environment, receiving delayed feedback in the form of rewards	Active: Takes actions in the environment	Delayed, Scalar Rewards

Advantages of RL

- **Focuses on the long-term goal**
 - **Easy data collection process**
 - **Learn from interaction**
 - **Operates in an evolving & uncertain environment**
 - **Sequential Decision-Making**
- 
- A solid blue horizontal bar spanning the width of the slide, located at the bottom.

The RL Problem

The Reinforcement Learning (RL) problem is a framework in machine learning where an agent interacts with an environment over a series of discrete time steps. The agent's goal is to learn a policy that maximizes the cumulative rewards it receives from the environment.

RL explicitly considers the whole problem of a goal-directed agent interacting with an uncertain environment. This contrasts with many approaches that consider subproblems without addressing how they might fit into a larger picture.

Key Components

Environment, Agent, State, Action, Reward

Policy (π): The strategy or mapping from states to actions that the agent follows. The goal of RL is to learn an optimal policy that maximizes the expected cumulative rewards.

Value Function (V): The expected cumulative rewards that an agent can expect to receive from a given state onwards, following a specific policy. It helps the agent evaluate the desirability of different states.

Model (Optional): A learned or known model of the environment's dynamics, which provides information about how the environment will respond to the agent's actions. Models are not always used, but when available, they can assist in planning and decision-making.

Exploration vs Exploitation

Exploration and exploitation is one of the key characterizing aspects of RL

There is always a trade-off between the two, where the agent must balance between:

- Taking new actions to explore the environment and gather information
- Taking the best known strategy so far in a greedy fashion to maximize the reward

There are multiple ways to balance between exploration and exploitation, and

RL Applications

Gaming

Teaching agents to play games like chess, Go, pong or video games

Robotics

Training robots to perform tasks in real-world environments such as robotic arms for pick-and-place or path planning for ground vehicles

Finance

Optimizing investment strategies in financial markets.

Healthcare

Personalizing treatment plans based on patient data.

A solid blue horizontal bar spanning the width of the slide, located at the bottom.

Questions?

Markov Decision Process

- A mathematical framework which formalizes sequential decision making.
- In MDPs, actions influence not just immediate rewards, but also subsequent states and, consequently, future rewards.
- An MDP consists of states, actions, transition probabilities, and rewards.

States (S): Represent possible situations or configurations in the environment.

Actions (A): Define the set of possible decisions or choices available to the decision-maker.

Transition Probabilities (P): Describe the probabilities of transitioning from one state to another based on the chosen action.

Rewards (R): Provide numerical feedback to the decision-maker after each action in each state.

$$MDP = f(S, A, P, R)$$

The Markov Property

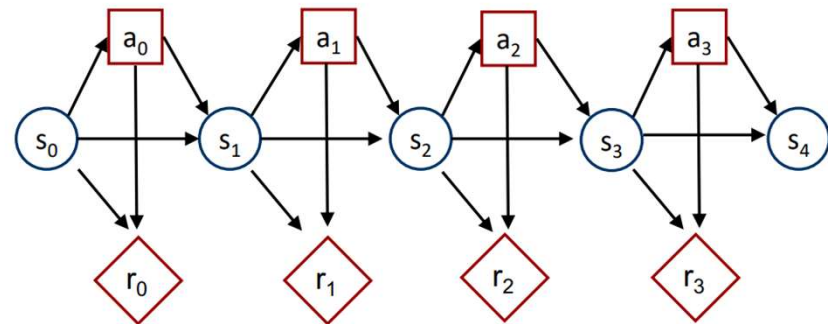
The Markov property simplifies the modeling process, as it allows decision-makers to focus on the present state without the need to consider the entire history of states and actions.

Markov Property

1. The future is independent of the past, given the present
2. MDPs are memoryless; the current state contains all relevant information for decision making

In a MDP, the agent interacts with the environment at discrete time steps $t = 1, 2, 3, \dots$

1. At each time step, the agent receives a representation of the environment, known as the state ($S_t \in S$)
2. Based on the state, the agent selects the action ($A_t \in A(s)$)
3. At the next time step, the agent receives a numerical reward based on its action $R_{t+1} \in R$, and the agent is now in a new state S_{t+1}
4. Therefore, the trajectory is:
 $S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \dots$



Finite MDP

In a finite MDP, the sets of states, actions, and rewards all have finite number of elements.

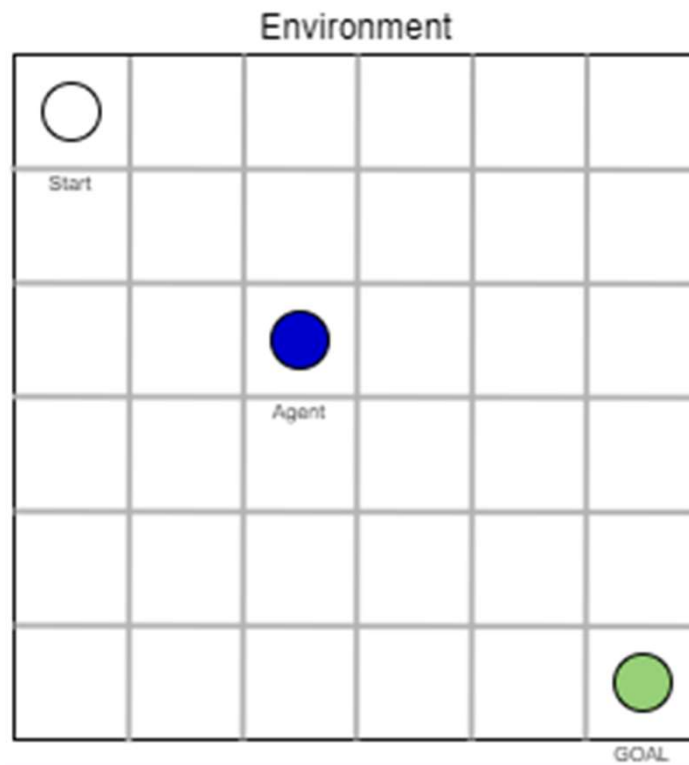
In this case, the random variables R_t and S_t have well-defined discrete probability distributions

For particular values of those random variables $s' \in S$ and $r' \in R$, there is a probability of those values occurring at time t , given the values of the previous state and action.

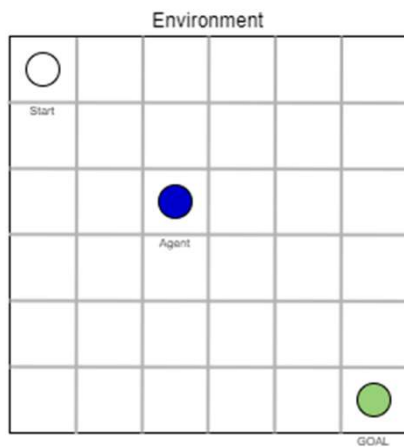
$$p(s', r | s, a) \doteq \Pr\{S_t = s', R_t = r' | S_{t-1} = s, A_{t-1} = a\}$$

$$\sum_{s' \in S} \sum_{r \in R} p(s', r | s, a) = 1 \quad \forall s \in S, a \in A(s)$$

Example: Simple Maze

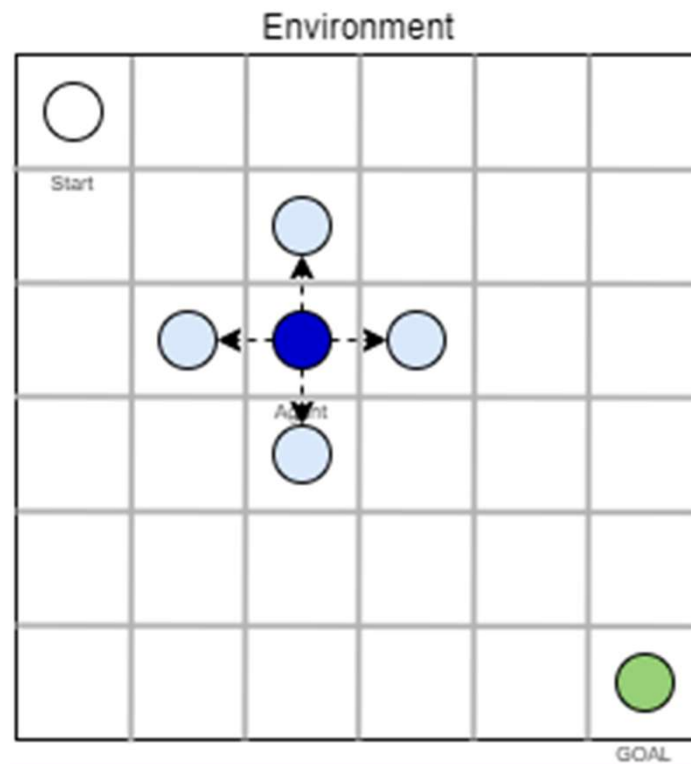


Example: Simple Maze

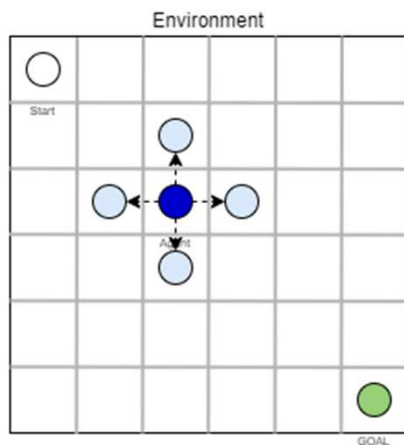


- What is the agent?
- What are the states?
- What kind of Actions does the agent have?
- Can this be modeled as a finite MDP?

Example: Simple Maze

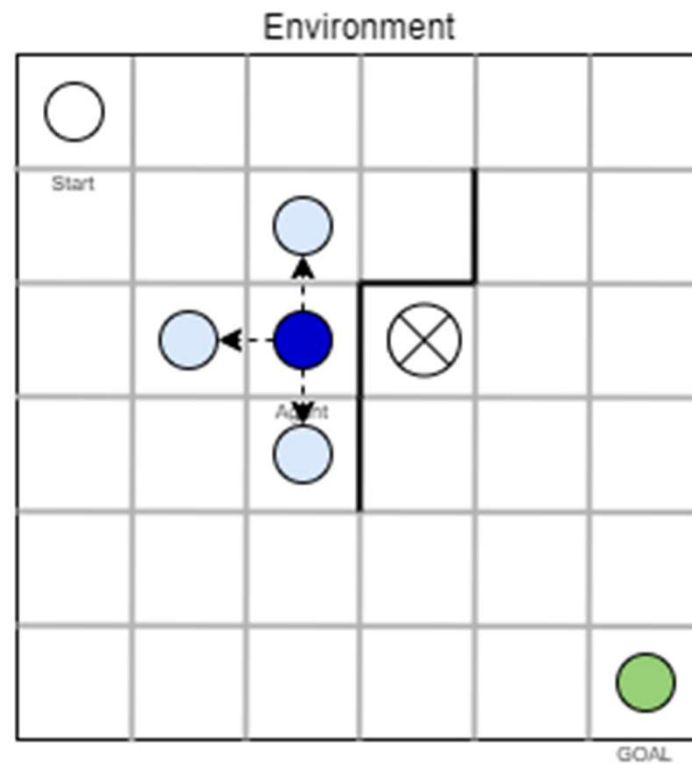


Example: Simple Maze

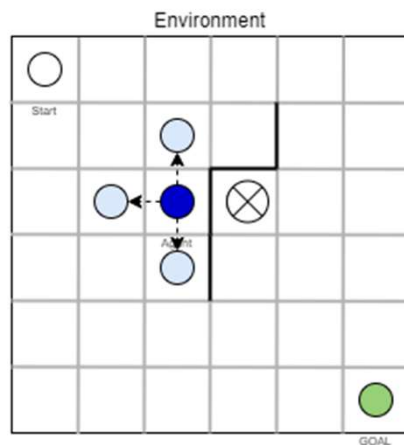


- What is the probability of each state given each action? – lets try a few examples
- Define the actions
- $p(s_{0,0} | s_{2,2}, a_{left})$
- $p(s_{2,3} | s_{2,2}, a_{right})$
- $p(s_{2,1} | s_{2,2}, a_{right})$
- $p(s_{1,2} | s_{2,2}, a_{up})$
- others?

Example: Simple Maze



Example: Simple Maze



- What is the probability of each state given each action? – lets try a few examples
- Define the actions
- $p(s_{0,0} | s_{2,2}, a_{left})$
- $p(s_{2,3} | s_{2,2}, a_{right})$
- $p(s_{2,1} | s_{2,2}, a_{right})$
- $p(s_{1,2} | s_{2,2}, a_{up})$

Thank you!
