# **Introduction**

Welcome to the first module of this course. In this module, you will learn **Classical Reinforcement Learning**. Reinforcement Learning (RL) is the field of machine learning in which an 'agent'(i.e. the software being trained) learns to take actions to maximise some cumulative 'reward'*.*

Some classical examples of RL are driverless cars, game playing agents (Chess, Go, etc.), mechanical robots in factories/warehouses etc. The field of Reinforcement Learning has seen some major breakthroughs in recent years:

* [DeepMind and the Deep Q learning architecture](https://deepmind.com/research/dqn/" \t "https://learn.upgrad.com/course/1688/segment/14428/89171/267127/_blank) in 2014,
* [Beating the champion of the game of Go with AlphaGo](https://deepmind.com/research/alphago/" \t "https://learn.upgrad.com/course/1688/segment/14428/89171/267127/_blank) in 2016 (you can watch the short video related to it [here](https://www.youtube.com/watch?v=8dMFJpEGNLQ" \t "https://learn.upgrad.com/course/1688/segment/14428/89171/267127/_blank)),
* [OpenAI and the PPO](https://blog.openai.com/openai-baselines-ppo/" \t "https://learn.upgrad.com/course/1688/segment/14428/89171/267127/_blank)in 2017
* How [Deep Blue beat a chess grandmaster](https://en.wikipedia.org/wiki/Deep_Blue_versus_Garry_Kasparov" \t "https://learn.upgrad.com/course/1688/segment/14428/89171/267127/_blank)

These achievements have in turn inspired other researchers and companies to turn to reinforcement learning. The most noticeable is the field of ****driverless cars****. Several automobile companies are hard at work for building cutting-edge-technologies for self-driving cars. [Tesla's](https://www.autotrader.com/tesla-cars.jsp" \t "https://learn.upgrad.com/course/1688/segment/14428/89171/267127/_blank)[Autopilot](https://www.tesla.com/autopilot" \t "https://learn.upgrad.com/course/1688/segment/14428/89171/267127/_blank) is one such system. Its 'driver assistance system' offers features such as lane-centring, adaptive cruise control, self-parking, etc.  Alphabet's [Waymo](https://waymo.com/" \t "https://learn.upgrad.com/course/1688/segment/14428/89171/267127/_blank), [Ford's](https://corporate.ford.com/innovation/autonomous-2021.html" \t "https://learn.upgrad.com/course/1688/segment/14428/89171/267127/_blank) self-driving car (which is due out in 2021) are locked in the competition to reach the final level of autonomous driving.

There have been interesting developments in the field of ****robotics****as well, where robots are trained for different tasks such as finding defects in objects, carrying an object from one place to other. [Fanuc](https://www.technologyreview.com/s/601045/this-factory-robot-learns-a-new-job-overnight/" \t "https://learn.upgrad.com/course/1688/segment/14428/89171/267127/_blank) has deployed a robot that uses RL to pick a device from one box and put it in a container.

Another domain where RL is used is ****finance.****RL is turning out to be a robust tool for evaluating trading strategies. Many companies are leveraging the 'Q-Learning' algorithm of RL with the simple objective of maximising the 'rewards' i.e. profits. You will study Q-learning and some of these applications in this course.

## ****The Evolution of RL****

The roots of Reinforcement Learning, acronymed as RL, go back to a psychologist, Edward L. Thorndike who talked about ****learning by trial and error****.  He studied cats in puzzle boxes. The cat was motivated to come out of the box. The cat would fall around and eventually stumble upon the latch that would open the box. Once the cat managed to get out, the same cat would be put in the same box again. After successive runs, he observed that cats were getting faster in finding and pulling the latch. And on the basis of this [behavioural experiment](https://www.youtube.com/watch?time_continue=3&v=fanm--WyQJo" \t "https://learn.upgrad.com/course/1688/segment/14428/89171/267127/_blank), Thorndike put forward the 'Law of Effect':

*“Responses that produce a satisfying effect in a particular situation become more likely to occur again in that situation, and responses that produce a discomforting effect become less likely to occur again in that situation.”*

In the first part of this course, you will study classical reinforcement learning techniques - ****dynamic programming****, ****Monte Carlo methods****, ****Q-learning**** etc. These algorithms are used to teach ‘agents’ how to perform a task.

In the second part, you will study ****deep reinforcement learning**** where you use classical reinforcement learning techniques along with deep learning to manage large state spaces.

## In this session:

We will cover the following topics:

Different Elements of a Reinforcement Learning Agent

Markov Decision Process

Value Functions

## Prerequisites

There are no prerequisites for this session apart from knowledge of the course 'Machine Learning'.

## Guidelines for in-module questions

The in-video and in-content questions for this module are not graded. Note that graded questions are given on a separate page labelled 'Graded Questions' at the end of this session. The graded questions in this session will adhere to the following guidelines:

|  |  |  |  |
| --- | --- | --- | --- |
|  | First Attempt Marks | |  | | --- | | Second Attempt Marks | |
| Questions with 2 Attempts | 10 | 5 |
| Questions with 1 Attempt | 10 | 0 |

## People you will hear from in this session:

****Subject Matter Expert:****

[G. Srinivasaraghavan](https://www.linkedin.com/in/gopalakrishnan-srinivasaraghavan-43b4b9?authType=NAME_SEARCH&authToken=uGg0&locale=en_US&trk=tyah&trkInfo=clickedVertical:mynetwork,clickedEntityId:1239599,authType:NAME_SEARCH,idx:1-1-1,tarId:1476708729573,tas:Srinivasaraghavan iiit " \t "https://learn.upgrad.com/course/1688/segment/14428/89171/267127/_blank)

****Professor, IIIT-Bangalore****

The International Institute of Information Technology, Bangalore, commonly known as IIIT Bangalore, is a premier national graduate school in India. Founded in 1999, it offers Integrated M.Tech., M.Tech., M.S. (Research) and PhD programs in the field of Information Technology.

**Report an error**

# **What is Reinforcement Learning?**

Ever noticed how an infant learns to walk? She puts a step forward. If she falls, she realises 'that was probably the wrong way'. She keeps trying and relies on the feedbacks (e.g. falling, being able to walk easily, etc.) to judge whether ‘the current technique is correct or not’.

You learn from your interactions with the world aka 'environment'. When you are learning to drive a car (assuming you have no instructor), you completely rely on the feedback you get from the surroundings. If you cross a lane or come too close to a tree, you change your technique (or actions, such as putting more breaks, turning the steering a little lesser, etc.).

Let’s first start with an overview of the RL problem and look at some examples which will help build an intuition of  RL.

Play Video

1569002

****Note:****

* The ****agent**** is any robot that is trying to learn the task,
* While the ****environment**** is the world around it that gives it the feedback.

When you are learning to walk, you are the agent and the surroundings are the environment.

Reinforcement learning is similar to 'human learning'. Remember the first time you were trying to learn to ride a bicycle? Learning how to balance and manoeuvre comes with experience. Maybe, when you had a fall (a negative experience), you learnt that the action which led to the fall was wrong and you should not do that again. Similarly, when you had a positive experience, you learnt what actions (how to keep your feet on the pedal, how much to turn the handlebar, etc.) led to a happy ride. Let’s look at a few more examples:

Play Video

1569002

Let’s now solve some basic questions to get a clear understanding of reinforcement learning, agent and environment; their interaction.

In the next segment, you will learn about Agent-Environment Interaction.

**Question 1/5**

Mandatory

#### **Reinforcement Learning**

A mobile robot has the job of collecting trash in an office. It has sensors for detecting cans, and an arm that picks it up. It runs on a rechargeable battery. The robot’s control system has components for interpreting sensory information, for navigating, and for controlling the arm.

It makes decisions depending upon the charge levels of battery. Consider there are three levels of charge: ‘high’, ‘medium’ and ‘low’. The agent needs to decide whether to (1) actively search for a can, (2) remain stationary and wait for someone to bring it a can, or (3) head back to its home base to recharge its battery.

Who is the agent in this case?

Robot’s control system

**✓ Correct**

**Feedback:**

It’s a high-level agent which is responsible for interpreting sensory information, for navigating, and for controlling the arm and gripper

Robot’s Battery level

Office environment

None of the above

**Attempt 1 of 2**

**Question 2/5**

Mandatory

#### **Reinforcement Learning**

A mobile robot has the job of collecting trash in an office. It has sensors for detecting cans, and an arm that picks it up. It runs on a rechargeable battery. The robot’s control system has components for interpreting sensory information, for navigating, and for controlling the arm.

It makes decision depending upon the charge levels of battery. Consider there are three levels of charging: ‘high’, ‘medium’ and ‘low’. The agent needs to decide whether to (1) actively search for a can, (2) remain stationary and wait for someone to bring it a can, or (3) head back to its home base to recharge its battery.

What is environment in this case?

Robot’s control system

Robot’s Battery level

Office environment

**✓ Correct**

**Feedback:**

Environment presents a new state and reward to the agent basis its actions

None of the above

**Attempt 2 of 2**

**Question 3/5**

Mandatory

#### **Reinforcement Learning**

A mobile robot has the job of collecting trash in an office. It has sensors for detecting cans, and an arm that picks it up. It runs on a rechargeable battery. The robot’s control system has components for interpreting sensory information, for navigating, and for controlling the arm.

It makes decision depending upon the charge levels of battery. Consider there are three levels of charging: ‘high’, ‘medium’ and ‘low’. The agent needs to decide whether to (1) actively search for a can, (2) remain stationary and wait for someone to bring it a can, or (3) head back to its home base to recharge its battery.

What are actions in this case?

Agent’s decisions: (1) actively search for a can, (2) remain stationary and wait for someone to bring it a can, or (3) head back to its home base to recharge its battery

**✓ Correct**

**Feedback:**

Actions are the tasks that an agent can perform

Robot’s Battery level: high and low

Office environment

None of the above

**Attempt 1 of 2**

**Question 4/5**

Mandatory

#### **Reinforcement learning**

Investment Management: You want to manage your investments such that the profit is maximised. Given that you can either buy, hold or sell a stock, what are the possible consequences (or outcomes)?

% Profit or Loss or no-profit-no-loss

**✓ Correct**

**Feedback:**

any action whether it’s buying or selling would either result in a profit or a loss or a no-profit-no-loss

the stock is being sold

Pressure and temperature change

**Attempt 1 of 2**

**Question 5/5**

Mandatory

#### **Reinforcement Learning**

Which of the following is the best possible explanation of RL?

RL is a learning problem where an agent learns from environment with help of a teacher telling what is right or wrong

RL is a learning problem where an agent learns from the environment by taking actions and controlling the consequences in its favor

RL is a learning problem where an agent is trying to learn from its environment by taking an action and understanding the consequences; in order to achieve an objective

**✓ Correct**

**Feedback:**

Environment doesn't tell the agent what is right or wrong. It tells the agent what is the consequence of its action.

All of the above

**Attempt 1 of 2**

Continue

## ****Humanoid robot example - additional reading****

Let’s take one more example: Say, an engineer is learning to design a humanoid robot and his objective is to make it stand. He can do that by turning some joints. A wrong turn of a joint can make it fall. By trial and error experience, he will learn which sequence of turning joints is making robot stand and which is not. This is a reinforcement learning problem. Here, the engineer is the agent and the robot on which he is taking the actions is the environment; actions are 'turning the joints' and consequences could be robot falling or standing up.

You can read this [paper](https://www.cc.gatech.edu/~isbell/reading/papers/peters-ICHR2003.pdf" \t "https://learn.upgrad.com/course/1688/segment/14428/89171/267127/_blank)to read more about this.

# **Agent-Environment Interaction**

In the last segment, several terms were introduced such as ****agent, environment, actions**** and ****consequences****. In the next few segments, let’s build an intuition for these terms. Before jumping into the equations of RL, it is important that you understand these concepts intuitively.

Let’s first start with the agent-environment interaction.

Replay

Unmute

Current Time 6:46

/

Duration 6:46

Loaded: 100.00%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

To summarise, agent ****observes**** the environment, then takes an action which he thinks is good. After taking the action, the environment tells him how good that action was, in the form of ****rewards****; this also presents a new observation to the agent.

Let’s take the example of a student learning to maximise grades in his training. He has grades of the exam that happened two weeks back. He ****observes**** the subjects in which he scored lower. And then studies ****(action)**** only for those subjects. For the remaining time, he plays or surfs. After a week, he goes through the exams again. His grades improve and now only in one subject his marks are a little less than the average. So, the action of studying had a positive consequence as his marks increased.  Apart from this, he observes the subject in which he scored low marks - that becomes his new observation.

****Reward****, in this case, is the increase in marks. However, note that the reward is not enough to judge his action. What if he had failed the subjects for which he did not study in the second attempt? Reward only tells you how well you are carrying out the task. It does not guarantee that this is the best action. In other words, reward is an indicator, or a 'weak signal', which roughly indicates whether the agent is taking actions in the right direction.

Unlike supervised learning, which classifies each observation as 'right' or 'wrong'; reward in reinforcement learning is just a number indicating how well you are performing the action. The robot needs to try and find out which actions are better than the others, if not the best. The objective here is to maximise the cumulative reward as the sequence of actions is taken.

Note that there are ****two types of tasks****:

* ****Continuous:**** tasks that do not have a definite end - e.g. learning to walk, controlling a chemical plant, driving a car
* ****Episodic tasks:**** tasks that have a definite end - e.g. most games (videos games, Chess, Ludo) etc. are episodic since at the end of the game the agent either wins or loses.

Let’s solve some questions now.

**Question 1/5**

Mandatory

#### **Reinforcement Learning**

Rewards are under the control of the agent, i.e., after taking an action, it can modify the rewards obtained

True

False

**✓ Correct**

**Feedback:**

Rewards are the property of the envirornment

**Your answer is Correct.**

**Attempt 1 of 1**

**Question 2/5**

Mandatory

#### **Reinforcement Learning**

Which of following are correct matches? Multiple options are correct

Automated stock trading - Continuous task

**✓ Correct**

**Feedback:**

There is no termination to this task, till the time you tell the agent to stop

Solving a maze - Episodic task

**✓ Correct**

**Feedback:**

There is an end to this task. When the agent exits the maze.

Playing Tic-tac-toe - episodic task

**✓ Correct**

**Feedback:**

There is an end to this task. The game will terminate in a win or a lose or a tie

**Your answer is Correct.**

**Attempt 1 of 2**

**Question 3/5**

Mandatory

#### **Reinforcement Learning**

Which of following is true about the episodic task. More than one options may be correct.

Episodic tasks end in a terminal state, i.e., they terminate after time T.

**✓ Correct**You missed this!

**Feedback:**

Episodic tasks have a terminal state.

Episodic tasks do not end. It goes on forever

**✕ Incorrect**

**Feedback:**

Episodic tasks have a terminal state.

Continuing tasks end in a terminal state

**Question 4/5**

Mandatory

#### **Reinforcement Learning**

Objective of RL agent is to:

Find sequence of actions basis some observation from environment

Find sequence of actions that accumulate maximum rewards

**✓ Correct**

**Feedback:**

The objective of RL agent is to carry out task well.  By doing the task well, it implies that its actions should be more aligned to accumulate maximum rewards

Understand how environment responds to its actions

Control the observations and rewards for its actions

**Question 5/5**

Mandatory

#### **Reinforcement Learning**

Which of following correctly describes what reward is? More than one options can be correct.

Reward is the property of an agent

Reward is just a parameter indicating how good or bad the action is.

**✓ Correct**

**Feedback:**

In RL, reward is a number telling the agent how good or bad his action is.

Reward is the  most important to judge an action. It tells whether the action is best or not

Reward is the property of the environment

**✓ Correct**

**Feedback:**

Reward is a property of the environment. Agent can't control how much reward to get for a particular action

# **State Vectors**

Till now, you were given an overview of the ****RL problem**** and the agent-environment interaction. Let’s now understand theoretically what is meant by an ****observation**** and how we formally represent a ****reward****.

Replay

Unmute

Current Time 3:22

/

Duration 3:22

Loaded: 100.00%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

So, a ****state**** is a representation of the environment at any point in time. The environment will give all the signals, but how ****relevant**** those signals are for the agent to take an action is what you have to decide. You can consider state vector as a list of features that help the agent to take an action. For each RL problem, state vector would be different.

Let’s take the example of a humanoid robot where your objective is to make him stand. Now, the environment offers you values for the following:

* pressure & temperature at some chemical plant
* robot’s joint positions
* the angle at the knee joint
* Bitcoin’s current price

Would all these variables impact the agent’s decision making? No. The pressure and temperature at the chemical plant would not help you decide which joint to turn. The action you need to make is on the basis of ‘which joint of the robot should be turned’. So, your 'state' will be the representation of robot’s joint positions and the angle at knee joint. These two are enough to take the next action.

State is ****your**** representation of the environment. Perhaps the environment would have a lot of things, but the state that you want to take will determine which parameters in the environment really matter to you.

Let’s have the professor illustrate a few more examples in the upcoming video.

Replay

Unmute

Current Time 4:19

/

Duration 4:19

Loaded: 100.00%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

So, the representation of the environment which is necessary for the agent to take an action is called state. In any real-life scenario, it will be left on your judgement to decide what variables are good for the agent to take an action.

Let’s now solve some questions on state. There is no right or wrong in these questions. It is about which option is the best among the options.

**Question 1/5**

Mandatory

#### **State Vector**

Which of the following statements are ****incorrect****? More than one options may be correct.

Agent interacts with the environment in the form of actions, and the environment responds to the agent by rewarding these actions and by presenting a new set of state vectors to the agent

**✕ Incorrect**

**Feedback:**

Agent interacts with the environment by taking actions. And actions result in some consequence, which is two-fold: change in state and a reward

Reward is in the form of a vector

**✓ Correct**You missed this!

**Feedback:**

Environment rewards the agent's action as a number. And it is just a weak signal of telling how good or bad an action is

State is usually represented by a number

**✓ Correct**

**Feedback:**

State represents what features agent wants to observe from the environment. It is usually represented in form of a vector.

Agent observes the state variables and basis that takes some action

**✕ Incorrect**

**Feedback:**

Agent observes what the state vector and then takes an action

**Attempt 2 of 2**

**Question 2/5**

Mandatory

#### **State Vector**

Self-driving cars: You are designing a self-driving car and you want the car to observe the values of the state vector to take the next action. Which of following options do you think makes a good state vector for this problem.

(Pressure, temperature, wind speed in process control system)

(Current position of car on road, speed of car, the angle of the steering wheel, distance of obstacles (other vehicles, buildings, pedestrian), the current position of obstacles (other vehicles, pedestrian) image in rear-view mirror, road condition, white marks on road, current traffic signal)

**✓ Correct**

**Feedback:**

These features can help the agent to decide the next action

(Current position of car on road, speed of car, angle of steering wheel, distance of obstacles (other vehicles, buildings, pedestrian), direction of obstacles’ movement (other vehicles, pedestrian) image in rear-view mirror, road condition, white marks on road, current traffic signal, stock market rates, current performance in office)

(Current position of car on road, speed of car, angle of the steering wheel, distance of other vehicles from it, weather, stock market rates)

**Attempt 2 of 2**

**Question 3/5**

Mandatory

#### **State Vector**

Inventory Management: Say you are the owner of a retail shop. Your objective is to maximise the profits earned in a day. So, you need to place an order to meet the demand. On any day, you observe the ‘state vector’ and take an action for placing an order (from the supplier) which arrives later the next day. Assume that your demand is a function of the day of the week. Which of the following is the most relevant ****state vector****?

(Current inventory level, number of people working in the retail shop)

(Current inventory level, the day of the week, number of people working in the retail shop)

(Current inventory level, the day of the week)

**✓ Correct**

**Feedback:**

These features are good enough for the agent to decide the next action

**Attempt 2 of 2**

**Question 4/5**

Mandatory

#### **State Vector**

Deciding cover for each movie/series on Netflix homepage: Say, Netflix wants to customize the home page for each of the customer. It wants to show very relevant cover photo for each movie or series it recommends. Example, a person watching a lot of horror movies, will be interested if he is shown some intense scene from the movie as a cover photo. So, the action is to decide the cover photo, basis some state vector. What could be the state vector in this case? Choose the most appropriate option.

(Customer’s past preferences of genres, actors, directors)

(Today’s weather, current market condition, customer’s past preferences of genres)

(Customer’s past preferences of genres, actors, directors, ratings of movie he has watched)

**✓ Correct**

**Feedback:**

This state vector seems to be more appropriate than other options

**Attempt 1 of 2**

**Question 5/5**

Mandatory

#### **Tic-Tac-Toe**

Consider a game of Tic-Tac-Toe in which an RL agent is playing against a human. (If you haven't played the game before, you can refer to the link [here](https://www.youtube.com/watch?v=5SdW0_wTX5c" \t "https://learn.upgrad.com/course/1688/segment/14428/89171/267127/_blank)).

What is the state-vector of the RL agent?

The current position of all Xs on the board

(value of cell, position)

Value of a cell can null of filled with X or O

**✓ Correct**

The current position of all Os on the board

**Attempt 1 of 2**

# **Objective of RL Agent**

So far, you have learnt that the agent is interacting with the environment in an 'optimal manner' to achieve the objective. But, how do you define the ****objective of an RL problem****? Can you say that the objective is to gather maximum rewards?

Take an episodic task, say a game of tic-tac-toe. How will you calculate the reward for each ****O (or X)**** you marked? You will get a reward after you win or lose the game. What is the agent’s end-objective in this case? Similarly, for a continuing task, say for stock market trading, you can define your reward as how much profit you earned in a month or a day. There is no end to continuing tasks, so how are you going to parametrise your objective?

Let’s hear prof. Raghavan explain the objective of the RL agent in the following video.

Replay

Unmute

Current Time 3:30

/

Duration 3:30

Loaded: 100.00%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

You can summarise the objectives of the RL agent in the following manner:

* The objective of ****episodic tasks**** is to find such a sequence of actions that will make the majority of episodes successful.
* For ****continuing tasks****, break it into multiple episodes and then find out actions that maximise the average rewards earned from those episodes.

In the next segment, you will learn about the actions and policies.

**Question 1/2**

Mandatory

#### **RL Objective**

Which of the following is/are incorrect? More than one options may be correct.

The objective is defined after observing the state vector from the environment

**✓ Correct**

**Feedback:**

The objective of the RL agent is pre-decided. It doesn't depend in what state agent is in.

Just like the reward, the objective is also a property of the environment

**✓ Correct**

**Feedback:**

RL objective is not decided by the environment

The objective of RL agent in an episodic task is to find such sequence of actions that will make a majority of episodes successful.

**Attempt 1 of 2**

**Question 2/2**

Mandatory

#### **RL Objective**

A self-driving car gets rewarded for every 1km of a ride. And after every 8 hours, it needs to charge itself.

Which of following is the correct objective of a self-driving car?

Find out how much traffic is there today

Find out an action sequence that could accumulate maximum rewards in maximum rides

**✓ Correct**

**Feedback:**

Since the car needs to charge itself after every 8 hours, this makes it an episodic task. And the objective then would be to find a sequence of actions that would accumulate maximum rewards

Find out an action sequence that would make the average hourly ride smooth.

None of these

# **Actions & Policy**

So far, you have two pieces of information:

1. An action leads to a reward and a change in state.
2. The objective of the RL agent is to find the sequence of actions to maximise overall rewards.

Both have something in common which we have still not discussed in detail – ****actions.****

How does an RL agent take an action from a given state? Are there some rules defined for each scenario?  Or is there some mathematical model of the environment which the agent learns over time and then takes an action? We will handle all these questions in this segment.

Replay

Unmute

Current Time 3:45

/

Duration 3:45

Loaded: 100.00%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

Agent needs to learn about the environment before it behaves in an optimal manner. Learning essentially means that the agent interacts with the environment by trying out different actions and understanding their consequences.

The consequence is two-fold, one in form of ****rewards****, other in form of ****change of state****. It is like a child learning how to walk. He needs to understand which foot to put forward, to keep both the legs straight, etc., and the consequence for each of these. So, he will remember what action he took and what consequence it led to. This memory of action and consequence is called ****knowledge-base**** or ****history**** for an RL agent.

Now, the agent can look up in its knowledge base and see which action leads to the best consequence when in a given state. You could also build a ****mathematical model**** of the environment rather than storing all possible (action, consequence) pairs. We will discuss how to do that in a while.

All this is good for smaller problems where there are ****very few states and actions****. In more realistic situations, it is very difficult to explore all possible states and actions and therefore makes it difficult to build a knowledge base or a model. You will learn about both the scenarios in detail later in the course.

Watch the video given below to learn about some more terms which are frequently used in reinforcement learning.

Replay

Unmute

Current Time 8:54

/

Duration 8:54

Loaded: 100.00%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

A ****policy**** is a set of rules which helps the agent decide the action that it should take in a given state such that the agent can maximise its rewards in the long run. There are two types of policies:

* A ****deterministic policy****: π(s)→a
* A ****probabilistic policy:**** π(a|s)

A probabilistic policy becomes deterministic when π(a|s)=1

For example, for a novice doing investment portfolio management, the policy could be: whenever the stock price reaches a certain threshold, he will sell all the stock. This is the deterministic case. He has fixed the action for a state. A probabilistic policy, on the other hand, could be: whenever the stock price reaches a certain threshold, sell the stock 60% of the times, retain the stock 35% of the times, and for rest of the times, buy the stock. What if the stock price keeps on increasing and he holds the stock rather than selling it? Well, he could earn more profit by selling it later.

Let’s solve some questions to understand the idea of policy better. Consider the cab service scenario: The current state of cab is (09:00am, 1st January, Point A). He gets a request in the form of (pickup, drop). Let's say he has gotten two requests which he has to choose from: (Point A, Point B) and (Point C, Point B).

In the next segment, you will learn about Exploration vs Exploitation.

**Question 1/4**

Mandatory

#### **Reinforcement Learning**

Consider, he services the second request. What will be the consequent state after completing the ride?

(12:00pm, 1st January, Point A)

(12:00pm, 1st January, Point B)

**✓ Correct**

(12:00pm, 1st January, Point C)

**Question 2/4**

Mandatory

#### **Reinforcement Learning**

Consider, the reward is calculated as (amount of cash earned during the trip – amount of fuel burnt). Assume Point A, B and C are equidistant from each other, so the amount of cash earned from Point A to Point B and Point C to Point B is the same. In which of the following cases, the reward earned is more?

From Point A to Point B

**✓ Correct**

**Feedback:**

If the cab driver is initially at Point A and he services 2nd request, he'll have to waste extra fuel in going from Point A to Point C. Cash earned in both the trips is same. But fuel burnt in second case is more

From Point C to Point B

It is same in both cases

**Attempt 1 of 2**

**Question 3/4**

Mandatory

#### **Reinforcement Learning**

Consider two cab drivers are in a state s = (09:00 am, 1st January, Point A) and they could service two actions:  
Action 1: (Point A, Point B)  
Action 2: (Point C, Point B)  
The policy for one cab driver is deterministic in the sense that no matter the time or day, he will never pick a customer from Point A. That implies π(s) -> action 2.  
And for other cab driver, it is stochastic: π(action1|s) = 0.3, π(action2|s) = 0.7. And given that rewards are calculated as per the previous question, ****which driver could get a chance to earn more profit ?****(Assume Point A, B and C are equidistant from each other, so the amount of cash earned from Point A to Point B and Point C to Point B is the same.)

First cab driver

Second cab driver

**✓ Correct**

**Feedback:**

Though, the policy of both the drivers is inclined to action 2. But driver 2  has a chance of earning more profit than driver-1 on some days.

Both will get an equal chance

**Question 4/4**

Mandatory

#### **Reinforcement Learning**

Which of following correctly defines a policy?

A mapping from state that helps the agent to figure out what action needs to be taken

**✓ Correct**

A set of variables that define what is the current situation of the environment

Policy is another terminology for defining RL objective

**Attempt 1 of 2**

# **Exploration vs Exploitation**

In the last segment, you studied two types of policies - ****deterministic****and ****stochastic****. You also understood why a stochastic policy is preferred over a deterministic one in most practical cases. In this segment, we will discuss another reason why a stochastic policy is better.

Let’s first build an intuition of this discussion. Let's say you love Indian food. You have the following policy (based on, say, Zomato reviews) to choose a restaurant when in a given mood:

π(“SmokeHouseDeli”|happy)=0.6;π(“Chinabistro”|happy)=0.35;  
π(“DelhiHouse”|happy)=0.05

This is your current policy. Say, you are happy, and you decide to go out for a family dinner. The most obvious choice is “Smoke House Deli” as per the policy. Now, on some day, you thought of giving “Delhi House” a try. You went there, and you loved the food much more than that of “Smoke House Deli”. If you wouldn’t have ‘explored’ this option, probably, you would have never had that tasty Indian food!

(P.S. This example is just for illustration. Not to point out whether Smoke House Deli is better or Delhi House.)

This concept of exploration will set the grounds for why a stochastic policy is often better than a deterministic one.

Replay

Unmute

Current Time 5:09

/

Duration 5:09

Loaded: 100.00%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

****Exploration-Exploitation Tradeoff:****Exploiting an action is fine if you have exhaustively explored all the actions from a given state. But, this is generally not the case in real-life situations. In most scenarios, you would have explored only a small fraction of all possible actions. What if there exists an action that can get you a lottery? Wouldn’t you go exploring more? But at the same time, you also don’t want to lose out on the benefits of the current action, in case you don’t find good options while exploring.

So, to handle this problem, typically, a small window of exploration is set. We will go into the mathematical formulation later, but for now, let’s stick only to the theoretical concept of exploration vs exploitation.

**Question 1/2**

Mandatory

#### **Reinforcement Learning**

What is exploitation?

Exploiting the current option which you already know is reaping benefits

**✓ Correct**

Finding out other options

**✕ Incorrect**

Both a and b

**Attempt 2 of 2**

**Question 2/2**

Mandatory

#### **Reinforcement Learning**

Investment Management Portfolio: Consider you know which stocks will always reap benefits. And you invest your money there. Will you always risk your money there? Will you never explore? What if there exists some stock which is reaping 200% profits? (Text question)

Word Count **5**Word Limit **1 - 100**

**Attempt 1 of 1**

# **Markov State**

In the previous segment, you learnt that when in a given state, an agent takes an action according to a policy (which is learnt during training - we will discuss training algorithms later). The action leads to a change in state and possibly generates a reward. One bruteforce way to learn a policy is to actually remember all the possible pairs of state, action and reward. But that is often not feasible. For example, in a game of (say) Chess, this set may comprise of a million possible combinations. In more complex problems (such as playing Go, driving a car etc.) this may be further intractable. Therefore, in the RL problem, we make a ****Markovian**** assumption.

Let's now understand what is Markov assumption and how can we define the Markov assumption for RL problems.

Replay

Unmute

Current Time 6:21

/

Duration 6:21

Loaded: 100.00%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

The ****Markov assumption**** states that the current state contains all the necessary information about all the past states the agent was in and all the past actions the agent took. It assumes that the current state is sufficient for taking the next action.

You can consider a Markov state as some function of the knowledge base that captures all relevant information from the knowledge base. And once ‘Markov state’ is known, the knowledge base can be thrown away. What action the agent needs to take next or what possible state an agent can land on given he has taken an action - all of this can be determined from the ****Markov state****.

Let’s consider a robot learning to fly a plane. It has the knowledge base of ****position, the speed**** at each and every time step. Now, at some position, the robot makes a right turn. The next state of the plane will be dependent on what the current position and speed of the plane are, and the robot has taken the right turn. It doesn’t need to know how it arrived at the current position or how it gained its current speed. Its current state vector (current position, current speed) and action satisfy the Markovian assumption.

Now, let’s revisit some examples we have discussed earlier and see whether the Markov assumption holds in all those cases.

Pause

Unmute

Current Time 2:04

/

Duration 10:01

Loaded: 30.65%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

All these processes that work in accordance to Markov property are called ****Markov Decision**** ****Processes**** (popularly called ****MDPs****). The word ‘Decision’ in MDP takes into account actions taken by the agent in a given Markov state. MDP is the formal name of a sequential decision-making process. All the RL problems set its ground on MDPs, i.e., work on the assumption of the Markov property. We will uncover more on MDPs in later segments. For now, let’s solve some questions on Markov property.

**Question 1/2**

Mandatory

#### **Reinforcement Learning**

Which of following statements are correct about Markov state?

Markov state contains all necessary information that helps to predict the future state.

**✓ Correct**

**Feedback:**

According to the Markovian assumption, the current state is sufficient to take the next action. Information about past is not required. Now, it depends on how you define your state

Markov state doesn’t contain necessary information to take the next action

All the above

**Attempt 1 of 2**

**Question 2/2**

Mandatory

#### **Reinforcement Learning**

Inventory Management Problem: Say you are the owner of a retail shop. Your objective is to maximise the profits earned in a day. So, you need to place an order to meet the demand. On any day, you observe the ‘state vector’ and take an action for placing an order (from supplier) which arrives later the next day. Assume that your demand is a function of day of the week.  
State for this problem is (current stock level, day-of-week). Is this state Markov?

Yes

**✓ Correct**

**Feedback:**

The demand trend is incorporated in the ‘day-of-week’ and current stock level observations are sufficient to take the next action

No

**Attempt 1 of 1**

# **Markov Decision Process (MDP)**

So far, everything has been explained to develop an intuition of all the important ideas. From this segment onwards, we will build the mathematical formalisation of all that you have studied. Before we do that, let’s encapsulate the learning so far.

The following points apply to an RL problem:

* An agent learns how to behave in an environment by taking actions.
* Then observing the consequences (rewards and next state), of the action taken.
* The control objective of the agent is to learn a policy to accumulate maximum cumulative rewards over a period of time.
* All of RL problems are based on the Markov assumption: the current state contains all relevant information to take the best action.

**Question 1/6**

Mandatory

#### **Reinforcement Learning**

A policy is a mapping from perceived states of the environment to actions to be taken when in those states

True

**✓ Correct**

**Feedback:**

Policy defines what action RL agent will take when in a given state

False

**Attempt 1 of 1**

**Question 2/6**

Mandatory

#### **Reinforcement Learning**

Rewards are the property of agent.

True

False

**✓ Correct**

**Feedback:**

The environment rewards the agent’s actions. Even though agent might know how much reward it can get for its action, but it cannot control the reward awarded Agent cannot decide how much reward it should get for its actions

**Attempt 1 of 1**

**Question 3/6**

Mandatory

#### **Reinforcement Learning**

Stochastic policy is generally preferred over deterministic. Which of following options justify the statement? More than one options may be correct.

Stochastic policy gives a chance to explore, whereas deterministic does not

**✓ Correct**

**Feedback:**

In stochastic policy, the agent has an option to choose among different actions possible. In the case of deterministic policy, only one action is possible. So, stochastic policy is inherently exploratory, whereas deterministic is not

Stochastic policy provides an edge over deterministic by allowing to choose some less probable actions

**✓ Correct**

**Feedback:**

In stochastic policy, the agent has an option to choose among different actions possible. In the case of deterministic policy, only one action is possible. So, stochastic policy is inherently exploratory, whereas deterministic is not

Deterministic policy helps to exploit the best possible action known. There’s no need to explore

Stochastic policy is easy to handle as only one action can be taken from a state

**Attempt 1 of 2**

**Question 4/6**

Mandatory

#### **Reinforcement Learning**

What is Markov in Markov Decision Process?

The state vector

**✓ Correct**

**Feedback:**

Markov assumption is for states. That state should be sufficient to make the current decision

Actions

Policy

Rewards

**Attempt 1 of 2**

**Question 5/6**

Mandatory

#### **Reinforcement Learning**

Say an RL agent is trying to learn to play the game of tic-tac-toe. What could be the Markov state in this case? Choose the most appropriate option.

The current position of all X's

Positions of all X’s and O’s

**✓ Correct**

The current position of all O's

**Attempt 1 of 2**

**Question 6/6**

Mandatory

#### **Reinforcement Learning**

Say an RL agent is trying to learn to play the game of tic-tac-toe.  The opponent is playing with O. What is the objective of RL agent?

Find a sequence of actions that will help him win maximum number of games

**✓ Correct**

**Feedback:**

This is an episodic task where the agent should win maximum games possible

Find the next best state after taking an action

None of these

**Attempt 1 of 2**

# **Value Function**

Let’s say an agent is learning to play a game where it has to fight enemy's drones in a battlefield. Assume that his current location and the number of bullets in his gun defines his state. His action could be - to move forward or to shoot. The reward is the number of drones he shot without getting killed. Let’s say he found a position behind a wall and he can shoot as many drones as he wants without getting killed.

The position behind the wall has offered him an advantage and is more valuable to him. On the other hand, the agent once found himself out in the middle of the battlefield where he was an easy target for the drones.

This implies that some states are inherently more ****valuable**** than others, i.e. the agent prefers some states over others. It will be useful if the agent learns the 'values' of all these states. In the upcoming lecture, Professor Raghavan will explain the concept of 'value' in detail.

Replay

Unmute

Current Time 7:56

/

Duration 7:56

Loaded: 100.00%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

## **State value function**

Value function helps in evaluating a state. Consider the following table.

|  |  |  |
| --- | --- | --- |
| 3 | 4 | 9 (Target) |
| 2 | 5 (Current) | 8 |
| 1 | 6 | 7 |

Assume that you want to reach cell 9 from the current state (which is cell 5). Assuming you can take actions only from the following set {Left, Up, Down, Right}, the possible next states you can be in, starting from cell 5, are - cell 2, cell 4, cell 6, cell 8. You evaluate all the states, based on how valuable each of these states would be in achieving the final goal of reaching state/cell 9.

Clearly, if you move to cell 4 (or 8), the chances of reaching cell 9 in the next step are higher than if you were in cell 2 or 6.

The positional advantage you have simply by being in a particular state is the intrinsic ****value of the state.**** For example, it is far more valuable to be in state 4 as opposed to state 3, because the immediate step after 4 will result in achieving the target i.e. state 9. So, state 4 is inherently more valuable than state 3. So, you can select an action that helps you achieve the state with the maximum 'value'.

A value function tells you how good it is for the agent to be in that particular state. It is also known as the ****state value function****. The agent would want to be in a state such that its ****total rewards (immediate + expected future rewards) are maximised.****

## 

## ****Total rewards****

Consider that the agent starts from state S0, takes an action A0 and gets an immediate reward of R0 and ends up in state S1. From there he takes action A1 and so on. So, his episode is:

(S0, A0, R0), (S1, A1, R1), (S2, A2, R2), (S3, A3, R3), (S4, A4, R4), S5

where S5 is the terminal state.

Now, you want to calculate total rewards earned from State S2, that will be: (R2)+ (R3+R4). Here R2 is an immediate reward and (R3+R4) are future rewards.

## **Action value function**

On a similar note, we can define a value function for action, i.e., define how valuable it is to perform an action in a particular state. Consider the game of chess - you moved your rook to take the opponent’s queen and got an immediate high reward for this move. But taking this action might have been useful in some other state, but in this state (where the King gets endangered) this action is quite unfavorable.

Let's understand ****action value function**** in detail.

Replay

Unmute

Current Time 4:45

/

Duration 4:45

Loaded: 100.00%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

You take actions that bring about states with high value, i.e. actions that fetch high immediate reward + expected future rewards because these actions bring the greatest amount of reward over the long run.

|  |  |  |
| --- | --- | --- |
| 3 | 4 | 9(Target) |
| 2 | 5(Current) | 8 |
| 1 | 6 | 7 |

Consider the above example: If you (the agent) are in state 5, let’s say you have the choice of performing 4 actions in that state. You can go either left, right, up or down. All actions are not equally valuable. Some actions can make you closer to the target. So, in state 5, actions ‘up’ and ‘right’ will take you closer to the target cell 9, while the actions ‘down’ and ‘left’ will take you away from the target cell. So when you are in state 5, actions ‘up’ and ‘right’ are more valuable.  
   
This function calculates the intrinsic value of performing an action when you are in state ‘s’.

Also known as the ****q-function, q(s, a)**** is the ****reward you can expect**** if you take an action ‘a’ in state ‘s’.  
   
For example, as Professor mentioned, if you are near the airport at 6:30 pm in the evening, the action of going to the airport by 8 pm is very valuable. Why? Because many flights land at around 8 pm in the evening, so the cab driver will be expected to get many long-distance rides at 8 pm if he is at the airport at that time.

To summarise,

* The ****state-value function**** v(s) is the total reward an agent can expect if it is in state s
* The ****action-value function**** q(s, a) is the total reward an agent can expect if it performs an action ‘a’ when it is in the state ‘s’

**Question 1/5**

Mandatory

#### **Reinforcement Learning**

State True/False - “Value function Q(s,a) represents how good it is for an agent to be in state ‘s’”.

True

False

**✓ Correct**

**Feedback:**

Value function Vπ(s) represents how good is a state for an agent to be in

**Attempt 1 of 1**

**Question 2/5**

Mandatory

#### **Reinforcement Learning**

Reinforcement learning agent aims to maximise ****a numerical value**** which represents the long term reward of a state. What is that numerical value?

Reward

Value function

**✓ Correct**

**Feedback:**

Value function indicates the long-term desirability of states after taking into account the states that are likely to follow and the rewards available in those states

Policy

All of the above

**Attempt 2 of 2**

**Question 3/5**

Mandatory

#### **Reinforcement Learning**

Action choices are made based on immediate reward obtained

True

False

**✓ Correct**

**Feedback:**

Action choices are made based on values and not just on immediate rewards. At each step, the RL agent aims to maximise the value function, which considers the future rewards as well, not just the immediate reward

**Attempt 1 of 1**

**Question 4/5**

Mandatory

#### **Reinforcement Learning**

Value function V(s) represents how good an action a is when in state s:

True

**✕ Incorrect**

**Feedback:**

Value function Vπ(s) represent how good it is for an agent to be in a state, irrespective of what action it takes.

False

**✓ Correct**

**Feedback:**

Value function Vπ(s) represent how good it is for an agent to be in a state, irrespective of what action it takes.

**Attempt 1 of 1**

**Question 5/5**

Mandatory

#### **Value Function**

The state-value function and action-value function reflects the expected total reward and not the immediate reward. True or False

True

**✓ Correct**

**Feedback:**

Yes, the value functions reflect the total expected reward the agent can get from that state/state-action pair.

False

**Attempt 1 of 1**

Once you fix the policy, the value and the q-function are fixed. Let’s understand this better in the following video.

Video Player is loading.

Pause

Mute

Current Time 0:00

/

Duration 0:00

Loaded: 0%

2x

Playback Rate

Picture-in-PictureFullscreen

1569002

So, if there are two cab drivers at the airport at 8 pm. Of which, one has the policy of earning maximum profit, while the other’s policy is to finish his day’s job by 9 pm. If you compute value function for each, they would be different for both, as the action they both would give priority to would be different, though being in the same state. One cab driver will be earning the profit, others will drive towards his home.

Therefore, we subscript both state and action-value functions with policy π:   
vπ(s),qπ(s,a)

# **Optimal Policy**

The objective of an RL agent is to find the best action to take in a given state, i.e. to learn an optimal policy. In the next video, the professor will explain what is meant by ****optimal policy****.

Replay

Unmute

Current Time 1:59

/

Duration 1:59

Loaded: 100.00%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

A policy π∗ is called optimal only if ∀  π:π∗≥π

But, π is just a mapping (of what actions to take in a given state). How can we compare two mappings? We must have some metric to say one policy is better than another. In the following video, the professor will explain how to compare two policies.

Pause

Unmute

Current Time 3:46

/

Duration 6:40

Loaded: 71.42%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

Policy π is better than π′ if, for all states, the value function is higher if you follow policy π than if you follow policy π′ . Mathematically,

π≥π′ if   ∀  s:vπ(s)≥vπ′(s)

We will not get into its formal proof now. But you will get an intuition of this later in the module while studying the conditions of optimality.

**Question 1/1**

Mandatory

#### **Reinforcement Learning**

A policy P is said to be optimal

If it has exactly one state-value greater than that state in every other policy

If it has all state-values  greater than corresponding state-values in every other policy

**✓ Correct**

**Feedback:**

Policy π is better than π′ if, for all states, value function is higher if you follow policy π than if you follow policy π′ .

**Attempt 1 of 1**

# **Model of the Environment**

There are broadly two types of frameworks in RL: ****model-based**** and ****model-free****.

* In ****model-based methods****, it is possible to learn what is called a model of the environment, i.e. a model which maps the consequences (next state, reward) of taking an action in a state.
* In ****model-free methods****, it is not possible to learn an explicit model (which is a more realistic case).

Let's learn about them.

Pause

Unmute

Current Time 6:21

/

Duration 9:07

Loaded: 80.70%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

In most cases, the environment is stochastic, i.e., most of the times you will see different rewards and states after taking a particular action in a particular state. Therefore, the model is represented as:

p(s′,r|s,a)

It is the probabilistic distribution of finding the agent in state 's' and reaping the reward r, given a particular action a is taken in a particular state s. This is known as ****the model**** ****of the environment.****

In most real-world scenarios, you wouldn’t know what exactly the model of the environment is. You implicitly infer about the model from the observations and the RL techniqes used to solve such problems are called ****model-free****.

So, the objective of an RL agent is to find the optimal policy either using the explicit model (model-based) or by implicitly inferring the model from the actions taken from various states (model-free).

In the next session, you will learn about the RL equations and how to solve them using Model-based or Model-free methods.

**Question 1/1**

Mandatory

#### **Reinforcement Learning**

Which of following correctly explains the difference between policy and model of the environment?

Policy defines which state the agent will land into and what reward it will get for taking an action from state s; whereas model of the environment defines what action the agent should take from a state

Model of the environment defines which state the agent will land into and what reward it will get for taking an action from state s; whereas policy defines what action the agent should take from a state

**✓ Correct**

Environment is stochastic and an agent can see any future state s and reward r after taking action from state. To handle this we make our policy deterministic

None of these

**Attempt 1 of 2**

# **RL vs Supervised Learning**

Before we start with the mathematics of Reinforcement Learning, it is essential to understand what makes RL different from supervised learning.

Let’s hear prof. Raghavan explaining these differences in the next video.

Pause

Unmute

Current Time 4:16

/

Duration 7:03

Loaded: 74.41%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

Let’s summarise the differences you learnt till now:

* In ****reinforcement learning****, you deal with the ****processes**** where the ****agent actively interacts**** with the environment. Whereas in ****supervised learning****, you deal with ****objects or datasets****. There is ****no interaction**** with the environment and given a dataset, you are required to predict the target.
* RL is an active learning, where the agent learns only by interacting. While supervised learning is passive learning, where the agent learns only by extracting features from a given dataset.

**Question 1/4**

Mandatory

#### **Reinforcement Learning**

Unlike reinforcement learning, in supervised learning there is no concept of agent taking actions and observing the consequence of its actions

True

**✓ Correct**

False

**Attempt 1 of 1**

**Question 2/4**

Mandatory

#### **Reinforcement Learning**

In reinforcement learning, we are given a dataset and the objective is to classify the observations as 0 or 1?

True

False

**✓ Correct**

**Attempt 1 of 1**

**Question 3/4**

Mandatory

#### **Reinforcement Learning**

In supervised learning, current observation helps in classifying the next observation to make sure Markov assumption holds

True

False

**✓ Correct**

**Attempt 1 of 1**

**Question 4/4**

Mandatory

#### **Reinforcement Learning**

In supervised learning, there is a ‘target’ or ‘label’ which is the best tag for that observation. Whereas in reinforcement learning, an agent only gets a reward for its action, but the reward doesn’t tell the agent whether this is the best action or not

True

**✓ Correct**

False

**Attempt 1 of 1**

Let’s look at few more differences in the next video.

Replay

Unmute

Current Time 5:23

/

Duration 5:23

Loaded: 100.00%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

The next major difference between the two learning methods is how the ****performance of the two is measured****.

* In supervised learning, there is a ****teacher (ground-truth)**** which tells you whether the result for a given observation is correct or not. And, then the model can be improved by minimising the error term.
* On the other hand, in reinforcement learning, there is no teacher. The environment acts only as a ****critic****, where it tells you how good or bad the action is by giving rewards. It doesn’t tell whether the action taken is the ultimate best or not.

****Classify**** the following questions as supervised or reinforcement learning

**Question 1/5**

Mandatory

#### **Reinforcement Learning**

A drone is learning to classify an image as human face or not. Is this supervised or RL?

Supervised

**✓ Correct**

RL

**Attempt 1 of 1**

**Question 2/5**

Mandatory

#### **Reinforcement Learning**

Your manager tells you to design a model to predict future sales to maximise the profit. He asks you to use last 3 years dataset and come up with a model.

Supervised

**✓ Correct**

RL

**Attempt 1 of 1**

**Question 3/5**

Mandatory

#### **Reinforcement Learning**

You decide to improve grades in class by observing your current grades and then deciding how much time you need to put to improve the current grades. Your objective is to maximise total grades earned in a semester

Supervised

RL

**✓ Correct**

**Your answer is Correct.**

**Attempt 1 of 1**

**Question 4/5**

Mandatory

#### **Reinforcement Learning**

Suppose, one day, you decide to learn to fly an aeroplane. So, learning to fly is:

Supervised

RL

**✓ Correct**

**Attempt 1 of 1**

**Question 5/5**

Mandatory

#### **Reinforcement Learning**

You are the owner of a retail shop and you want to learn how much order to place each day to meet the demand so that you can earn maximum profit? The order that you’ll place will be judged on how much profit you make. And it is for you to decide how much profit is good for you.

Supervised

RL

**✓ Correct**

**Attempt 1 of 1**

# **Inventory Management (MDP) -I**

In this segment, you are going to formulate an MDP for an ****inventory management**** problem.

A major concern of a warehouse owner is how much stock to order from the supplier to meet the customer demand and at the same time minimise his costs. The warehouse owner needs to ****learn the demand pattern**** and order accordingly. What if he was given a reinforcement learning agent who would learn from history how the demand has been and place orders accordingly?

Thought to ponder: Why not apply time series prediction and predict the future demand?

We could. But every few months the market conditions change. Will you apply time-series every month to adjust to the latest conditions? Reinforcement learning agent will constantly learn from the environment and update its demand curve.

With this let's start detailing out the problem statement.

Pause

Unmute

Current Time 6:02

/

Duration 6:10

Loaded: 100.00%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

**Question 1/3**

Mandatory

#### **Inventory Management**

What is the action of the storekeeper in this case?

Predicting the demand

Placing the order

**✓ Correct**

**Feedback:**

The way of interaction with the environment is by placing the order

**Attempt 1 of 1**

**Question 2/3**

Mandatory

#### **Inventory Management**

What is the objective of the storekeeper?

Predicting the demand

Maximise his profits

**✓ Correct**

**Attempt 1 of 1**

**Question 3/3**

Mandatory

#### **Inventory Management**

The storekeeper can control the Demand in order to maximise his profits

No

**✓ Correct**

**Feedback:**

Demand is not under agent's control. It is just an observation the environment is throwing to the agent. The agent can learn it but can't control it.

Yes

**Attempt 1 of 1**

The objective of the agent is to maximise profits and minimise the costs. Let's understand what all costs the owner incurs.

Replay

Unmute

Current Time 4:21

/

Duration 4:21

Loaded: 100.00%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

The revenue made by the warehouse manager depends upon the number of items he is able to sell. And the costs that will be incurred are:

* Buying cost from the supplier (b)
* Holding cost (d)
* Return cost (r)
* Delivery cost (s)

**Question 1/1**

Mandatory

#### **Inventory Management**

Will the storekeeper bear return cost in the following case? If yes, how much does he need to return?

Say, on Monday evening, the final inventory is 30. He places an order of 40 that will arrive Tuesday evening, in order to meet the demand. Let's say on Tuesday, the demand was 10.

Assume that the store's total holding capacity is 50.

No, 0

Yes, 10

**✓ Correct**

**Feedback:**

He does incur a return cost. His total inventory on Tuesday evening will be

Inventory on Monday-Demand of Tuesday +Action of Monday = 30-10+40 > 50

So, he needs to return 10

**Attempt 1 of 1**

In the next segment, you will define the ****MDP parameters****: state, action, transition probabilities and the reward structure.

# **Inventory Management (MDP) -II**

Let's continue defining our MDP for the inventory management problem.

Replay

Unmute

Current Time 8:18

/

Duration 8:18

Loaded: 100.00%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

So, the state in this MDP is: ****(Inventory size, day-of-the-week).****Basis these two features, the storekeeper can decide how much order to place. We will consider this state as Markovian.

Let's now define what could be the transition probability in this case.

Replay

Unmute

Current Time 7:08

/

Duration 7:08

Loaded: 100.00%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

**Question 1/2**

Mandatory

#### **Inventory Management**

On Wednesday evening (after meeting the demand of the day and stock of Tuesday has arrived), the inventory was 10 and the storekeeper places an order of 20. The demand on Thursday was 15. What will be inventory size on Thursday evening?

Note: By evening, it means the demand of the day is met and the previous day's stock has arrived.

15

25

20

**✓ Correct**

**Feedback:**

The inventory before the order arrives on Thursday will be 0 and after the action arrives, 20 will be final inventory on Thursday evening

30

**Attempt 2 of 2**

**Question 2/2**

Mandatory

#### **Inventory Management**

On Wednesday evening (after meeting the demand of the day and stock of Tuesday has arrived), the inventory was 30 and the storekeeper places an order of 20. What could be the range of inventory on Thursday evening

Note: By evening, it means the demand of the day is met and the previous day's stock has arrived.

(20,50)

**✓ Correct**

**Feedback:**

If the demand was high on Thursday, say around 30, the entire stock of 30 would be consumed. And then the order size of 20 arrives. So, the minimum inventory size would be 20. Similarly, if the demand is 0, the entire stock of Wednesday is left and the extra order of 20 will pile up to make it 50.

(30,50)

(0,50)

**Attempt 2 of 2**

Let's now look at the reward structure. Remember the following notations for the profit and the costs:

* Buying cost from the supplier (b)
* Holding cost (d)
* Return cost (r)
* Delivery cost (s)
* Opportunity cost (o)
* Profit (p)

Replay

Unmute

Current Time 6:41

/

Duration 6:41

Loaded: 100.00%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

****Correction:**** ****Holding cost**** is (inventory size)\*(holding cost per unit)

**Question 1/3**

Mandatory

#### **Inventory Management**

On a Saturday evening, the storekeeper started with a stock of 25. He anticipated that the demand on Sunday would be 30 and demand on Monday would be 20. How much order should he place?

assume: total warehouse capacity is 50

~20

**✓ Correct**

**Feedback:**

The demand on Sunday will consume the entire stock of Saturday evening. So, the storekeeper needs to order to meet Monday's demand. That means he needs to order ~20.

~30

~50

~25

**Attempt 1 of 2**

**Question 2/3**

Mandatory

#### **Inventory Management**

On a Saturday evening, the storekeeper started with a stock of 25. He anticipated that the demand on Sunday would be 30 and demand on Monday would be 20. Let's say he orders 22. Will he incur a return cost on Sunday evening when the action of Saturday arrives

assume: total warehouse capacity is 50

Yes

**✕ Incorrect**

**Feedback:**

On Sunday evening, his stock will be ~(max(25-30),0)+22).

22 <50, so he wouldn't incur a return cost

No

**✓ Correct**

**Feedback:**

On Sunday evening, his stock will be ~(max(25-30),0)+22).

22 <50, so he wouldn't incur a return cost

**Attempt 1 of 1**

**Question 3/3**

Mandatory

#### **Inventory Management**

On a Saturday evening, the storekeeper started with a stock of 25. He anticipated that the demand on Sunday would be 30 and demand on Monday would be 20. Let's say he orders 22. Will he incur an opportunity cost on Sunday evening

assume: total warehouse capacity is 50

Yes

**✓ Correct**

**Feedback:**

He will. He started with a stock of 25. But the demand was 30. So, there were 5 customers for whom he couldn't fulfil the demand. So, he would incur an opportunity cost in this case

No

**Attempt 1 of 1**

Later in the module, you will learn how to apply RL algorithms to solve the Inventory Management MDP.

Let's summarise what all you have learnt so far, in the next segment.

# **Summary**

****Reinforcement learning**** is about learning from interaction with the environment. The objective is to learn how to behave in order to achieve a goal. In this session, you learnt the following:

* In a reinforcement learning problem, ****an agent**** learns how to behave in an ****environment****  
  by taking ****actions**** and ****seeing the consequences**** - ****rewards and change in state.****
* The ****control objective**** of the agent is to ****learn a policy to accumulate maximum rewards****over a period of time.
* The entire reinforcement learning problem is based on the ****Markov assumption****: the current state contains all relevant information to take the future action.

You also learnt about deterministic and stochastic policy, and how the stochastic policy gives a window for ****exploration.****

Then, you learnt about****value function**** and ****action-value (q-value)****functions. These are the expected return from that state, or state–action pair, given that the agent uses the policy.

A policy for which value functions are optimal is an ****optimal policy****. While the optimal value functions for states and state–action pairs are unique for a given MDP, though there can be many optimal policies.

Then, you learn about the ****model of the environment.****In problems, where the agent has a complete knowledge about the environment’s dynamics we call them as ****model-based problems.**** In the case when a complete model of the environment is not available, we call those problems as ****model-free.****

In the next session, we will formulate the equations required to solve an RL problem.

**Question 1/1**

Mandatory

#### **Key Learnings**

What are your 3 key takeaways from this session?

Word Count **6**Word Limit **3 - 50**

**Attempt 1 of 1**