REINFORCEMENT LEARNING OVERVIEW

From Data to Decisions

Give users, hype-driven backboxes and you feed him for a day;

Teach them how to design algorithms and you feed him for a lifetime

Prepared by : Bhupen

AGENDA

- RL course overview
  - LMS
  - Generic overview
- Example – use cases
  - Games
  - Finance (active area of research, presently)
- Define RL
- RL vs (Sup/Unsup MLs)
- Elements of RL
- Finally define RL (again)
- Quiz

ABOUT THIS COURSE

- RL seems to hold the key to unlocking the promise of AI (ChatGPT e.g.)
- Learning RL (barriers to entry)
  - Reluctance
  - Mysterious
  - advanced math
  - Complicated engineering
- real-world, RL algorithms and implementations
  - Get heavy
- RL foundations can be learned without technical complexity.

APPROACH TO LEARNING RL

Focus on the foundational theory

emphasize rigorous but simple mathematical notations and formulations

encourage you to write out the equations yourself rather than just reading from the book

bring the concepts to life with simple examples and informal descriptions

After a topic/lesson – write small pieces of python code

we will avoid messy and complicated ML/RL/Big Data tools/packages (AWS…Azure)

stick to bare-bones Python/NumPy

Coding from scratch

LET US TRY TO DEFINE REINFORCEMENT LEARNING

Reinforcement Learning (RL) is a way to teach computers to learn by trial and error.

It's like training a pet or playing a game.

The computer tries different actions, receives feedback, and learns to make better choices to get rewards.

The goal is to find the best actions for different situations to get the most rewards.

WHAT DOES THE AGENT LEARN IN REINFORCEMENT LEARNING

Game Playing: RL has been used to train agents to play games such as chess, Go, and poker. For example, DeepMind's AlphaGo used RL techniques to defeat world champion Go players by learning from massive amounts of gameplay data.

Robotics: RL is applied to train robots to perform complex tasks. Robots learn to navigate environments, manipulate objects, or even control their own movements.

Autonomous Vehicles: RL plays a crucial role in training self-driving cars. Agents learn to make driving decisions, such as lane changing, speed control, and obstacle avoidance, based on sensor inputs and desired outcomes like safety and efficiency.

Recommendation Systems: RL algorithms can be used to optimize recommendations. By learning user preferences and feedback, agents can adaptively suggest personalized content, products, or services to users.

Inventory Management: RL is employed to optimize inventory levels and replenishment decisions. Agents learn to balance stock levels, demand forecasting, and supply chain dynamics to minimize costs while meeting customer demand.

Energy Management: RL is used to optimize energy consumption in various domains. For example, in smart grids, RL can learn to control energy storage systems, allocate power resources, and manage demand response for efficient energy utilization.

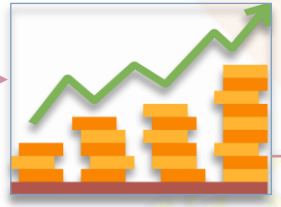
Finance: RL is utilized in algorithmic trading, portfolio management, and risk analysis. Agents learn to make trading decisions based on market conditions, historical data, and desired financial objectives.

Advertising and Marketing: RL is employed to optimize online advertising strategies. Agents learn to target specific audiences, allocate advertising budgets, and determine optimal bidding strategies to maximize the effectiveness of ad campaigns.

EXAMPLE - SALES

managing a team of sales representatives

Goal is to maximize the sales revenue



RL can be thought of as a way to train your team members to make the best decisions in different situations.

EXPLANATION

the team members are the agents,

the environment is the market or the sales context.

Each team member takes actions

such as choosing a sales strategy or

approaching potential clients, and

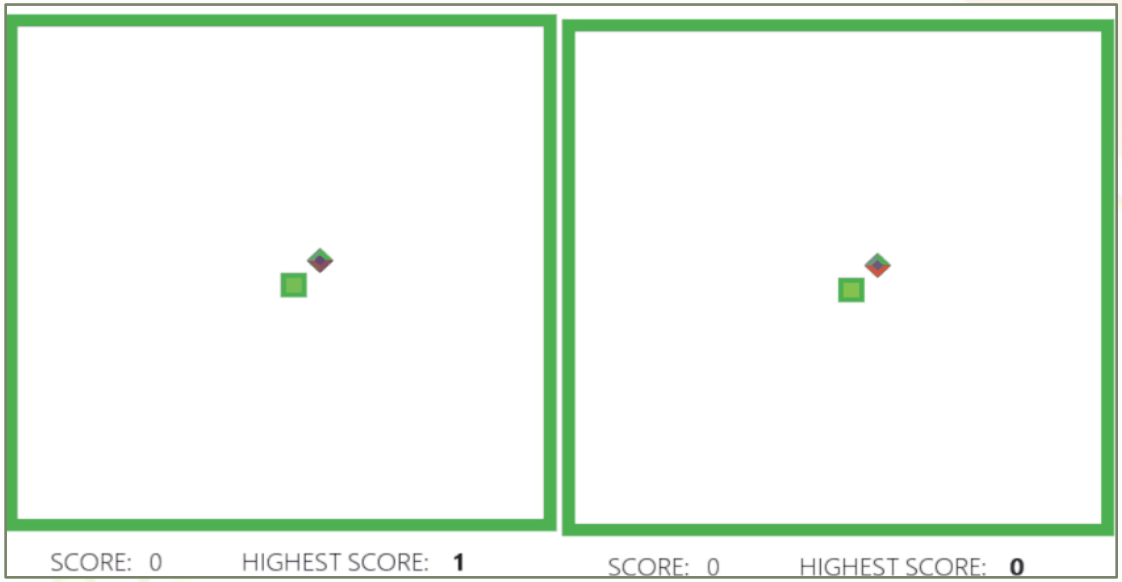
receives feedback in the form of rewards or penalties,

immediate sales,

customer satisfaction, or

other metrics

EXAMPLE - GAMES



AI does not know anything about the game

AI is trained and learnt how to play.

HOW IS RL DEVELOPED FOR GAMES

Environment: The video game serves as the environment.

Agent: The RL agent is the player or learner.

Action Space: The set of possible actions the agent can take in the game. These actions could include moving in different directions, jumping, shooting, or performing other in-game actions.

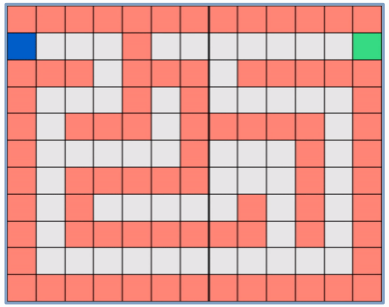
State Space : game's current state, such as the positions of the game objects, scores, or other relevant variables.

Reward Signal: The agent receives a reward signal from the environment based on its actions. In video games, rewards can be positive (e.g., gaining points, completing a level), negative (e.g., game over), or intermediate (e.g., collecting coins, defeating enemies). The goal is for the agent to learn to maximize the cumulative rewards it receives over time.

Learning Process: The RL agent learns by interacting with the game environment. It explores different actions, observes the resulting state changes, and receives rewards. The agent's goal is to discover the best actions to take in different states to maximize its long-term rewards.

EXAMPLE – GRIDWORLD

Gridworld environment
the robot can move 1 unit left, R, L, U or D at a time, to any 'safe' square (colored white).



agent cannot 'see' the entire map

Goal : robot (blue) has to reach the terminal cell (Green)

States : robot travels thru the cells, some cells robot is not allowed to go (orange cells)

Only white cells form the path

Possible decisions: the robot can go

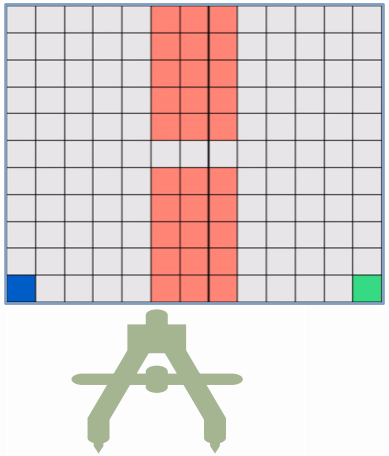
U, D, R, L

Rewards: -1 if the robot tries to enter the Orange cells, 0 for all other steps, 1 for the Goal

Training (n_iters) : randomly takes a step. For each possible step U, D, R, L, agent will compute the numeric value (goodness of the step in a particular cell (state)

Decision making (prediction) : max of the goodness values

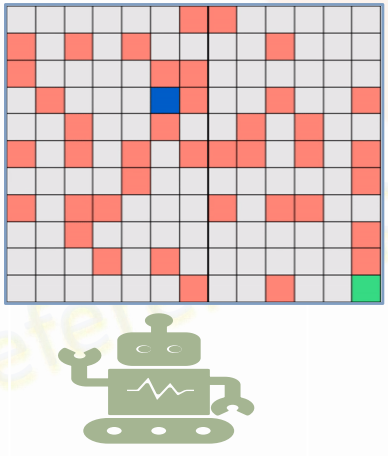
EXAMPLE – VARIATIONS OF GRIDWORLD



Example :

an instance where dangers are positioned in a **specific** manner, creating a confined pathway of secure squares that divides the world into two halves.

it must traverse a slender bridge composed of safe squares in order to reach the desired destination.



Example :

with randomly-placed hazards! In this case too, the robot must learn to navigate a hazard-free path to reach the target efficiently.

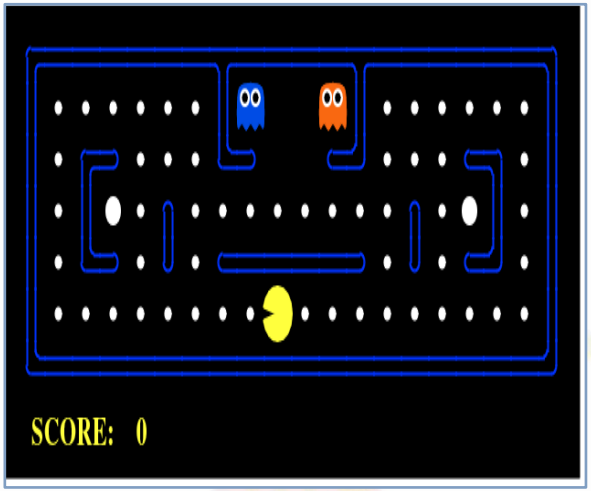
EXAMPLE - GAME PLAYING AI (CHESS)

a state : refers to a permissible arrangement of the remaining white and black pieces on the board

an action : denotes a legal move executed by any of the current pieces based on the rules of chess.

a potential reward structure : any move that does not promptly result in the goal state of checkmating the opponent incurs a negative reward of -1, whereas a move that successfully accomplishes a checkmate earns a substantial positive reward, such as 10,000.

EXAMPLE - GAME PLAYING AI (PAC-MAN)

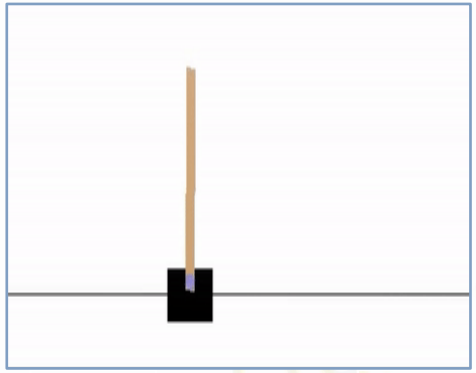


a state : refers to a permissible arrangement player/objects/enemies on the board

an action : denotes a legal move executed by any of the current.

a potential reward structure : design

EXAMPLE - GAME PLAYING AI (CART POLE)



In the cart-pole problem, a state comprises cart position, cart velocity, pole angle (deviation from vertical), and pole angular velocity.

values, continuous in nature, are discretized for practical purposes.

the agent can choose to keep the cart still or move it left or right along the horizontal axis.

reward : If the angle between the pole and the horizontal axis exceeds a specific threshold at any given state, the reward is 1; otherwise, it is 0.

The agent takes an action (keeping the cart still or moving it) from a starting state), resulting in a new state of the system.

The agent receives a reward for taking the action and transitioning to the new state.

NOW … LETS UNDERSTAND WHAT IS REINFORCEMENT LEARNING

Reinforcement learning is learning what to do

- how to map situations to actions
- so as to maximize a numerical reward signal.
- The learner is not told which actions to take, but instead must discover which actions yield the most reward by trying them.

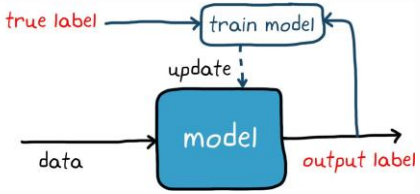
Sutton and Barto, Reinforcement Learning: An Introduction

- Machine learning (sup, unsup, RL)

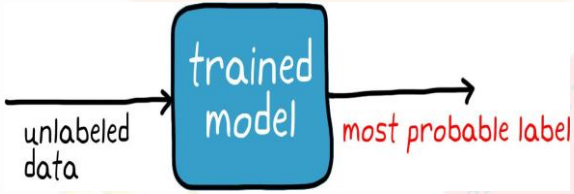
SUPERVISED LEARNING

animal dataset (labeled)

species	weight	height	num. of legs	communal living	domesticatable	…
rat	1.3	1.1	4	yes	yes	…
robin	1.2	0.8	4	no	no	…
elephant	98.5	12.2	4	yes	no	…
rabbit	2.5	2.1	4	yes	yes	…
spider	0.1	0.2	8	no	no	…
…	…	…	…	…	…	…



- a training set of labeled examples
- Each sample is a description of a situation together with a specification – label
- In RL terms - the correct action the agent should take to that situation



- model to extrapolate, or generalize, its responses so that it acts correctly in situations not present in the training set. (new state)
- Not adequate for learning from interactions.

SUPERVISED MODELS

Not adequate for learning from interactions.

In interactive problems

it is often impractical to obtain ALL examples of desired behavior (states)

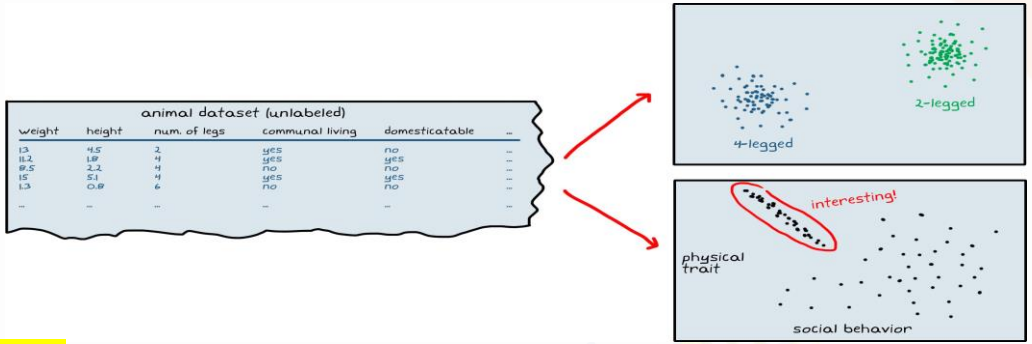
RL = uncharted territory (Stock market, Robots, Advertising..)

In uncharted territory

an agent must be able to learn from its own experience. (trial and error)

Reinforcement learning is different from supervised learning

UNSUPERVISED MODELS

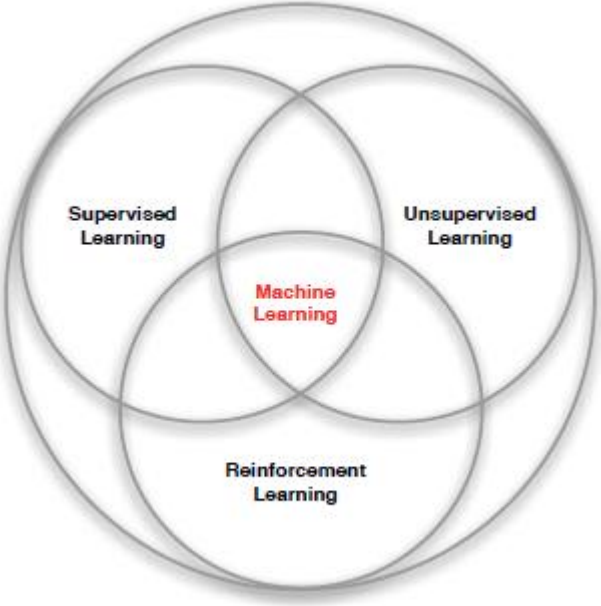


RL not same as unsup

- finding structure hidden in collections of unlabeled data.
- In this example : the structure could be based on number of legs or based on patterns that might not be as obvious, such as correlations between physical traits and social behavior

- For a new sample, we find the nearest sub-structure.
- RL seems like Unsup, as there are no labels
- RL is not meant to find hidden patterns/groups/structure

ML - PARADIGM



consider reinforcement learning to be a 3rd machine learning paradigm

- Elements of Reinforcement Learning

STATE

Chess: a state represents the arrangement of the chessboard with the positions of all the pieces. whose turn it is to play

Autonomous Driving: a state can encompass various sensor inputs, such as camera images, LIDAR data, GPS coordinates, and speed measurements.

Stock Trading: a state might involve financial data such as historical stock prices, technical indicators (e.g., moving averages), economic indicators (e.g., interest rates), and portfolio information (e.g., holdings, cash reserves).

Robotics: the state could include information about the joint angles of the robot arm, the position and orientation of objects in the scene, and the robot's gripper status (open or closed).

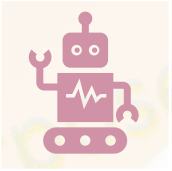
- a state refers to the current situation or configuration of an environment
- A state typically includes observable variables or features

ACTION

Chess: a move made by a player. It includes selecting a piece on the board and specifying its destination square. For instance, moving a pawn from e2 to e4 or moving a knight from g1 to f3.

Autonomous Driving: action could be the acceleration, deceleration, or steering angle of the vehicle. For example, increasing the throttle, applying the brakes, or turning the steering wheel to a specific degree.

Stock Trading: action might involve buying or selling stocks.



Robotics: movement of robot joints or the activation of specific gripper configurations. For instance, rotating a robot arm joint by a certain angle or opening/closing the gripper to grasp or release an object.

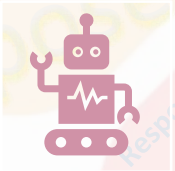
- an action refers to the specific move or decision taken by an agent in a given state.
- it represents the choices available to the agent to interact with the environment.
- The action determines how the agent will modify the state of the environment.

REWARDS

Chess: Winning the game may yield a positive reward (+1), losing may result in a negative reward (-1), and a draw could have a neutral reward (0)

Autonomous Driving: a +ve reward can be given for staying within the lanes, maintaining a safe distance from other vehicles, and adhering to traffic rules. -ve rewards can be provided for collisions, violations etc

Stock Trading: +ve reward for profitable trade, -ve reward for loss making decisions



Robotics: Grasping success = positive reward, dropping or failure = negative reward.

VALUE OR VALUE FUNCTION

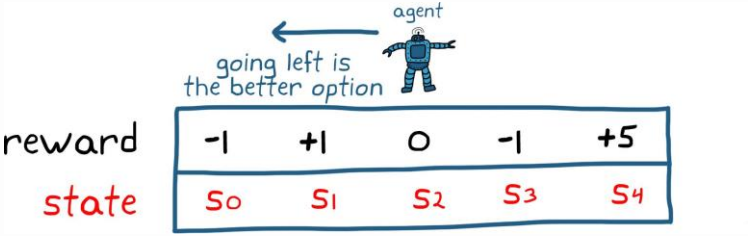
The value function is like a special power that tells the robot how good it is to be in a certain situation.

It is a numeric value.

If the value is high, it means you're doing well, and you'll probably get a lot of rewards.

It tells the robot which choices will give it more rewards or put it in a better situation.

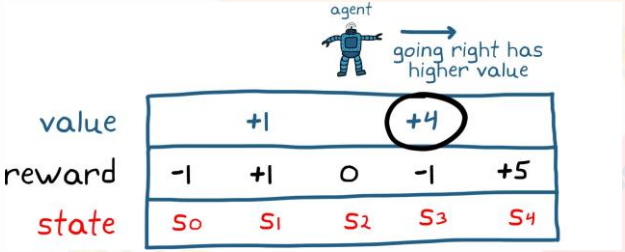
EXAMPLE



If the agent looks only at the reward for each action, it will step **left** first since that produces a higher reward than **right**.

Then it'll go back **right** since that again.

ultimately collect a total of +1



if the agent is able to estimate the value of a state, then it will see that going right has a higher value than going left even though the reward is lower.

Using value as its guide, the agent will ultimately end up with

+4 total cummulative reward.

WHAT REINFORCEMENT LEARNING AIMS TO SOLVE

RL aims to solve problems that involve making Sequential Optimal Decisions under Uncertainty.

Jargon Alert !!

UNCERTAINTY

Stochastic Environment: Uncertainty arises from random or unpredictable elements in the environment, such as sensor noise or variations in the agent's actions, leading to different outcomes even with the same action.

Unknown State Transitions: agent faces uncertainty about the probabilities of transitioning between states

Incomplete Observations: Limited or noisy observations make it challenging for the agent to have a complete understanding of the environment

Unknown Rewards: The agent may be uncertain about the precise rewards associated with different actions or state-action pairs, necessitating estimation or learning from limited feedback or historical data.

UNCERTAINTY - EXAMPLES

Stochastic Environment: Drone : The agent's actions may be influenced by unpredictable gusts of wind, leading to uncertain outcomes and making it difficult to precisely predict the drone's trajectory.

Unknown State Transitions: Playing card : playing a card game, but it can only see its own hand and a subset of the opponent's cards.

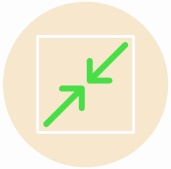
Incomplete Observations: Drone : Due to limited sensor capabilities or occlusions, the agent may have incomplete information

Unknown Rewards: The agent may be uncertain about the precise rewards associated with different actions or state-action pairs, necessitating estimation or learning from limited feedback or historical data.

evolution of random variables over time is very common
- in nature (e.g., weather) and
- in business (e.g., customer demand or stock prices*), but modeling and navigating such random evolutions can be enormously challenging*

OPTIMAL DECISIONS (OPTIMIZATION)

Optimal means : there is a well-defined quantity to be evaluated



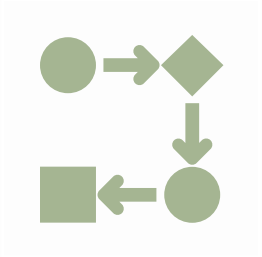
Optimization : The quantity to be maximized

Examples : might be financial (like investment value or business profitability),

Examples : or it could be a safety or speed metric (such as health of customers or time to travel),

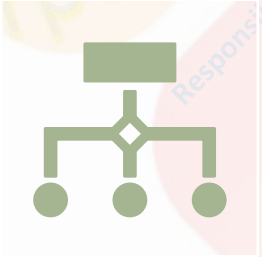
Examples : or something more complicated like a blend of multiple objectives rolled into a single objective.

SEQUENTIAL



refers to the notion that actions and **decisions** are made in a **sequential** manner over time.

an agent interacts with an environment over a series of discrete time steps,



each step involves observing the current state, taking an action, receiving a reward, and transitioning to a new state.

HENCE

- In essence,
  - the RL agent operates in an **uncertain** and **dynamic** world,
  - **continuously optimizing** its behavior through **sequential decision-making**.
  - By persistently navigating towards the goal, the agent aims to conquer uncertainty and achieve optimal outcomes in complex and evolving environments.

SUMMARY

- RL course overview
  - LMS
  - Generic overview
- Example – use cases
  - Games
  - Finance
- Define RL
- RL vs (Sup/Unsup MLs)
- Elements of RL
- Finally define RL (again)