GYM-MASTER

TEAM :
VYOMA MANKAD
ANSH SEMWAL
GAURAV TATPATE

MENTORS :
ADITYA VIVEKANAND

Reinforcement Learning

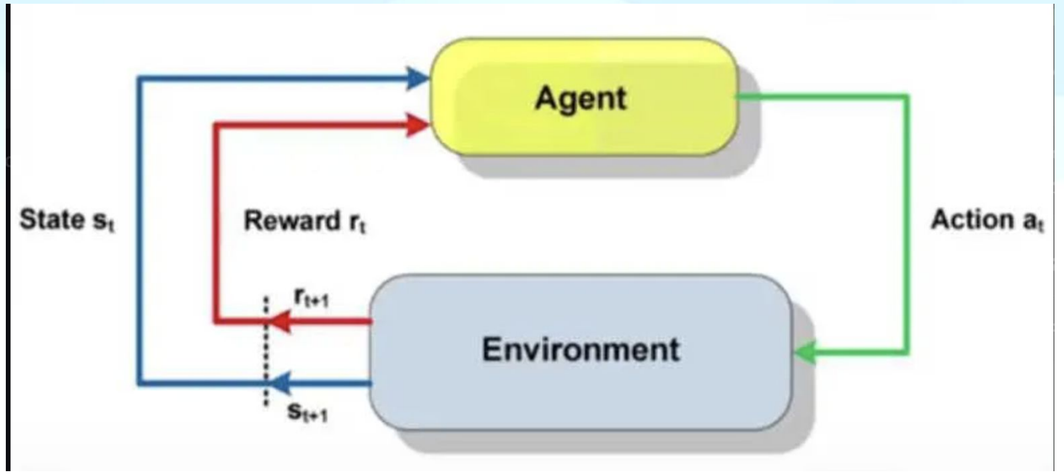
Agent and Environment

-Reinforcement Learning (RL): A type of machine learning where an agent learns to make decisions by interacting with an environment to maximise cumulative reward.

-Agent: The learner or decision-maker.

-Environment: The system with which the agent interacts.

-Reward: Feedback from the environment based on the agent's actions.



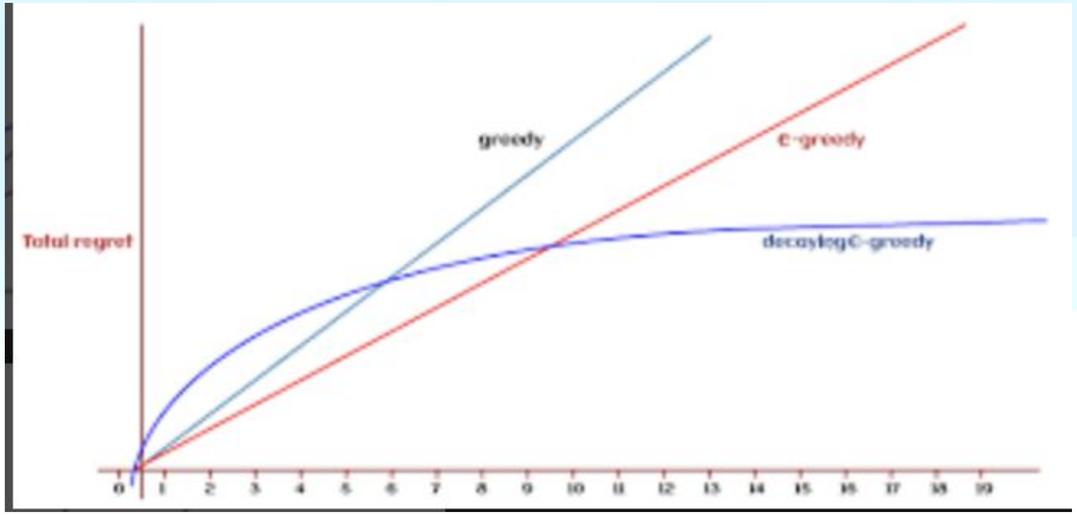
Exploration vs Exploitation

Epsilon Strategy ϵ

Exploration: Trying new actions to discover their rewards.

Exploitation: Choosing the best-known action based on past experience.

Epsilon is a parameter that decides if the agent explores new options or sticks to the best known option .



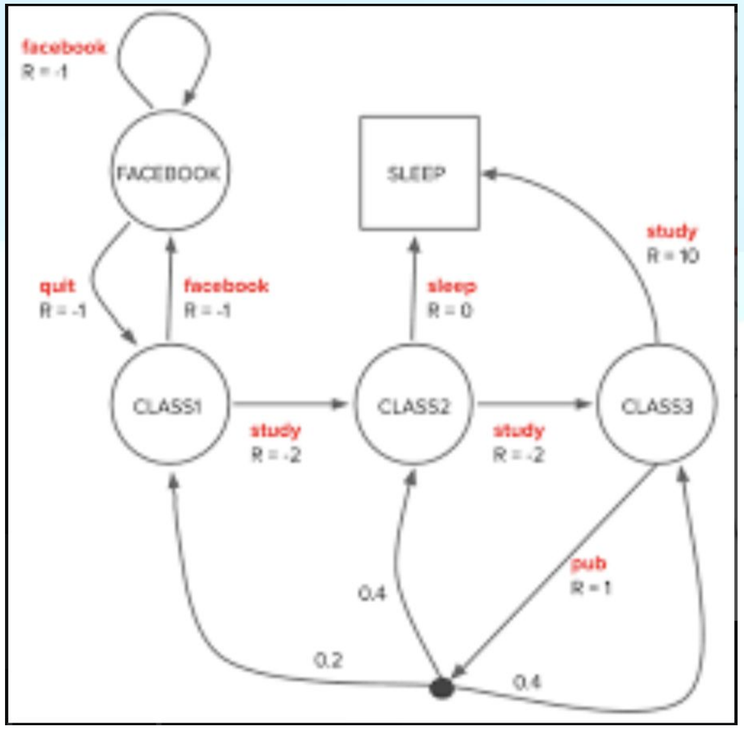
Markov Decision Process (MDP)

Understanding K-Armed Bandits

Markov Property :

“The future is independent of the past given the present”

K Arm Bandit Problem using epsilon greedy strategy .



Bellman Equation

The Foundation of Dynamic Programming

$$v(s) = \mathbb{E}\left[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s\right]$$

This equation expresses the relationship between the value of a state and the values of its successor states.

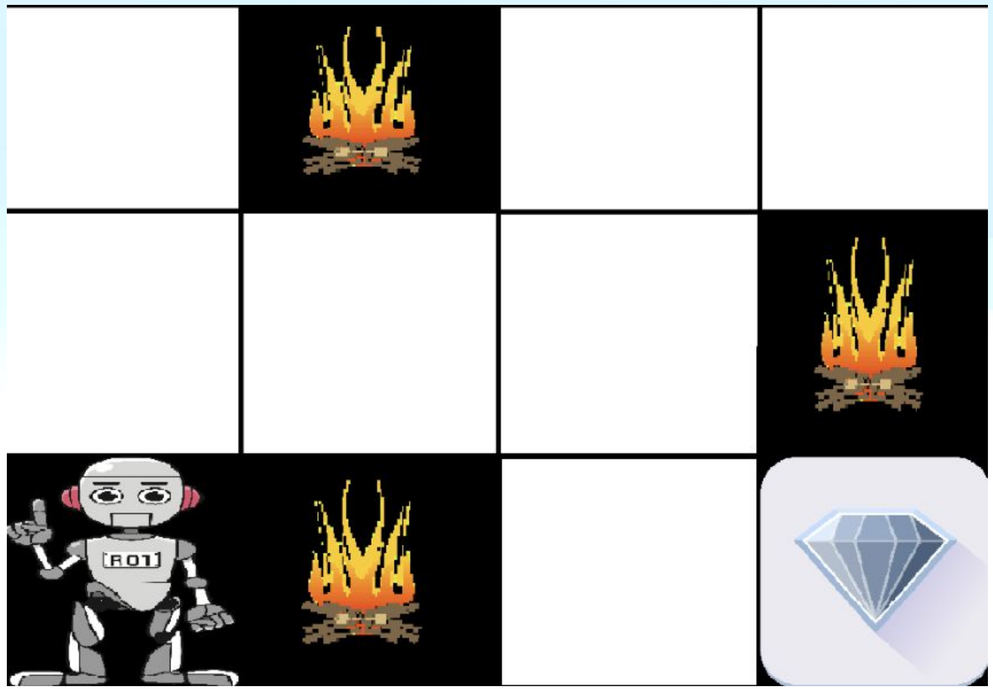
Dynamic Programming

Value Function and Policy Iteration

Value Function (V): Measures the expected cumulative reward from each state.

Policy (π): A strategy that specifies the action to take in each state.

GridWorld : Classic control problem in RL in which a agent solves a grid by itself using value and policy iteration.

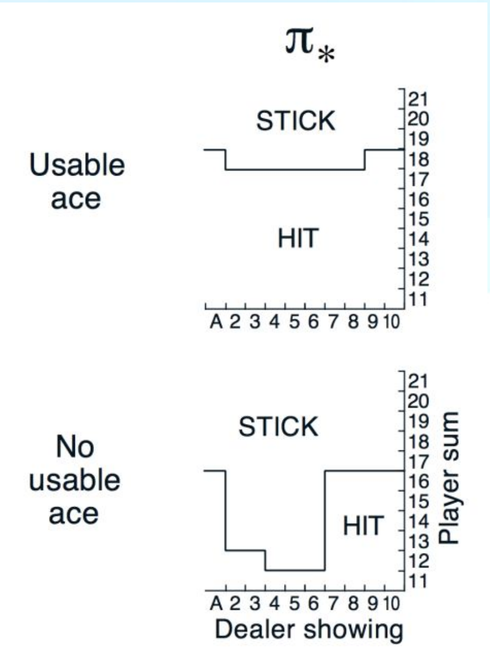


Monte Carlo Algorithm

Application to Blackjack

The term "Monte Carlo" refers to the use of random samples from the environment to estimate the expected return or value of a state-action pair.

$$V(s) = \frac{1}{N} \sum_{i=1}^{N} G_t$$



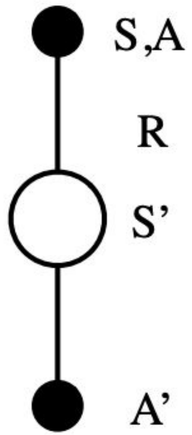
BLACK-JACK

Temporal Difference (TD) Learning

TD(λ) - A Unification of Monte Carlo and Dynamic Programming

TD(λ):
Combines TD and Monte Carlo, where λ controls the trade-off between them.

$$V(s) = V(s) + \alpha \left(R_{t+1} + \gamma V(s_{t+1}) - V(s_t) \right)$$



Q-Learning

$$Q(s,a) = Q(s,a) + \alpha \left(R_{t+1} + \gamma \max_{a'} Q(s',a') - Q(s,a) \right)$$

A model-free algorithm that learns the value of action-state pairs.

Most used RL algorithm

Problems solved with q learning :

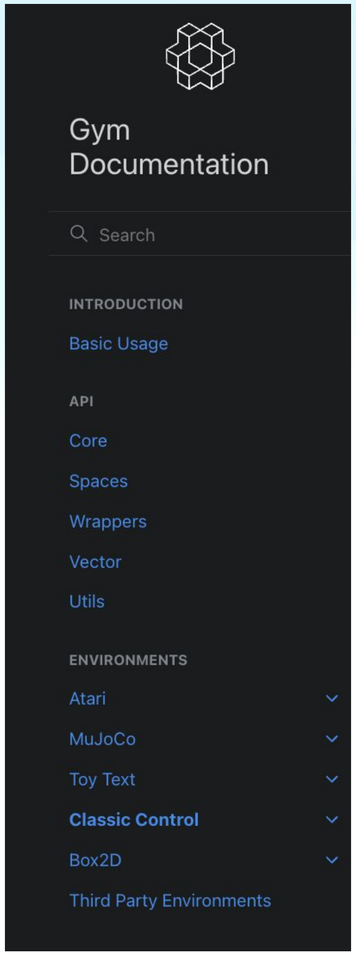
- Mountain Car
- Cartpole
- Frozen Lake

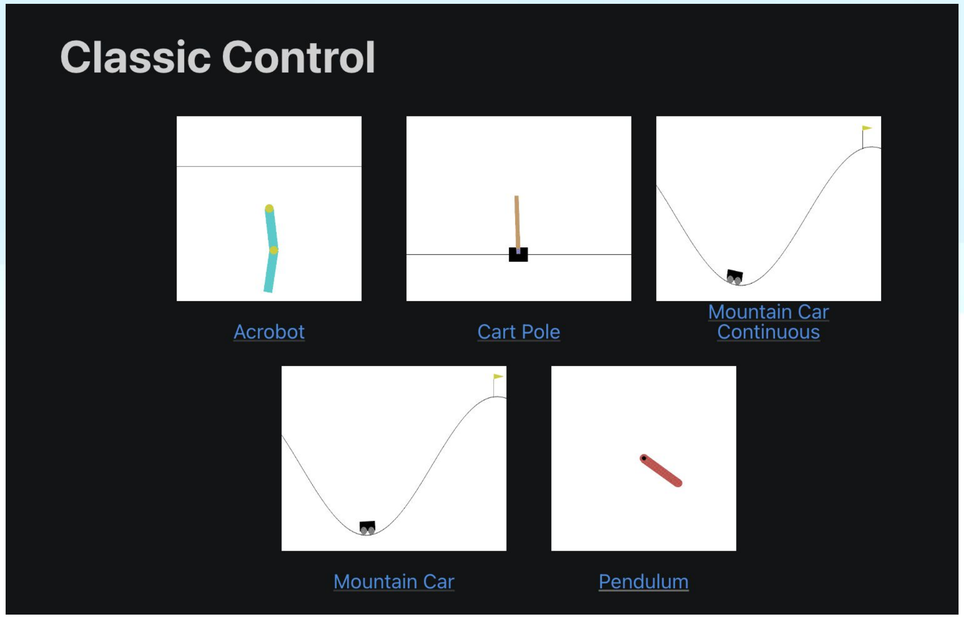
Introduction to OpenAI Gym

A Toolkit for Developing RL Algorithms

OpenAI Gym:

A platform providing a wide variety of environments for testing and developing RL algorithms.





WHAT WE HAVE IMPLEMENTED IN CLASSIC CONTROL:

MOUNTAIN CAR

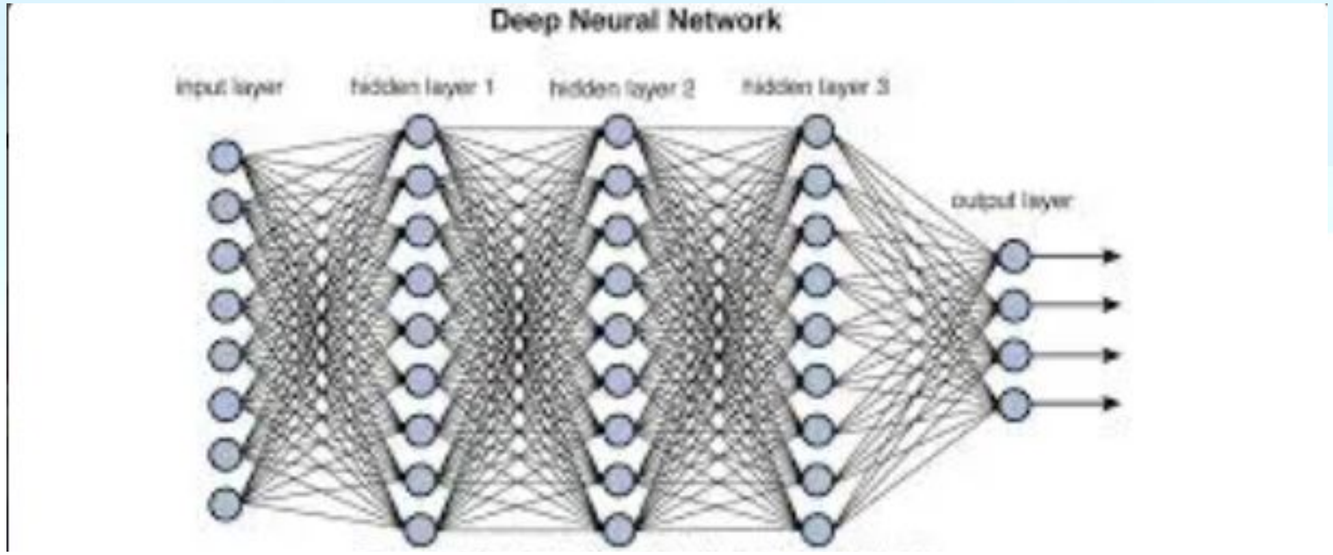
CARTPOLE

Deep Q-Learning and Neural Networks

Role of NNs and CNNs in RL

Deep Q-Learning: Extends Q-Learning by using neural networks to approximate Q-values.

CNNs: Convolutional Neural Networks process visual data, allowing agents to learn directly from pixels (e.g., in Atari games).

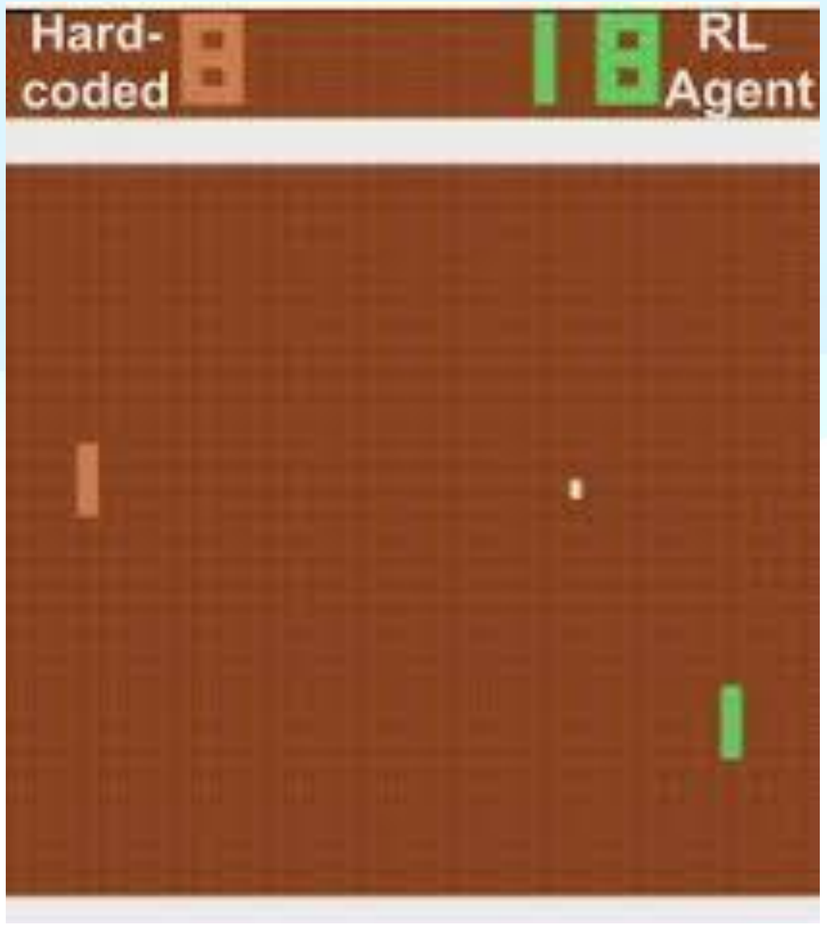


RL in Atari Games

Pong

Atari Games: RL has been successfully applied to play and master a wide range of Atari games.

We have implemented RI on Pong using Neural networks and convolutional Neural Networks .



Objective of the game is to not let the ball

cross our control paddle

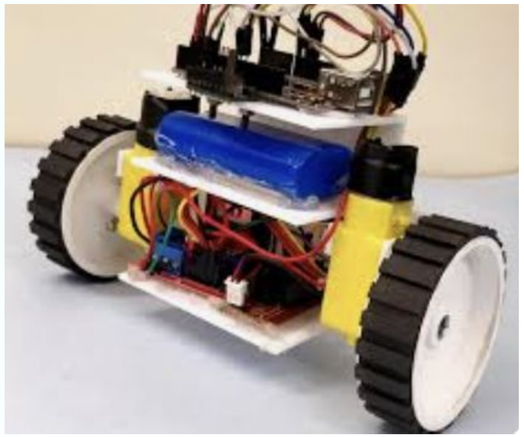
Problems faced:

Gym version not compatible with pcs

Dependency errors arise while trying to execute the pong game

Future Goals :

- Implementing pong using libraries such as keras and TensorFlow for better and efficient results
- Implementation of RL on a self-balancing bot



THANK YOU