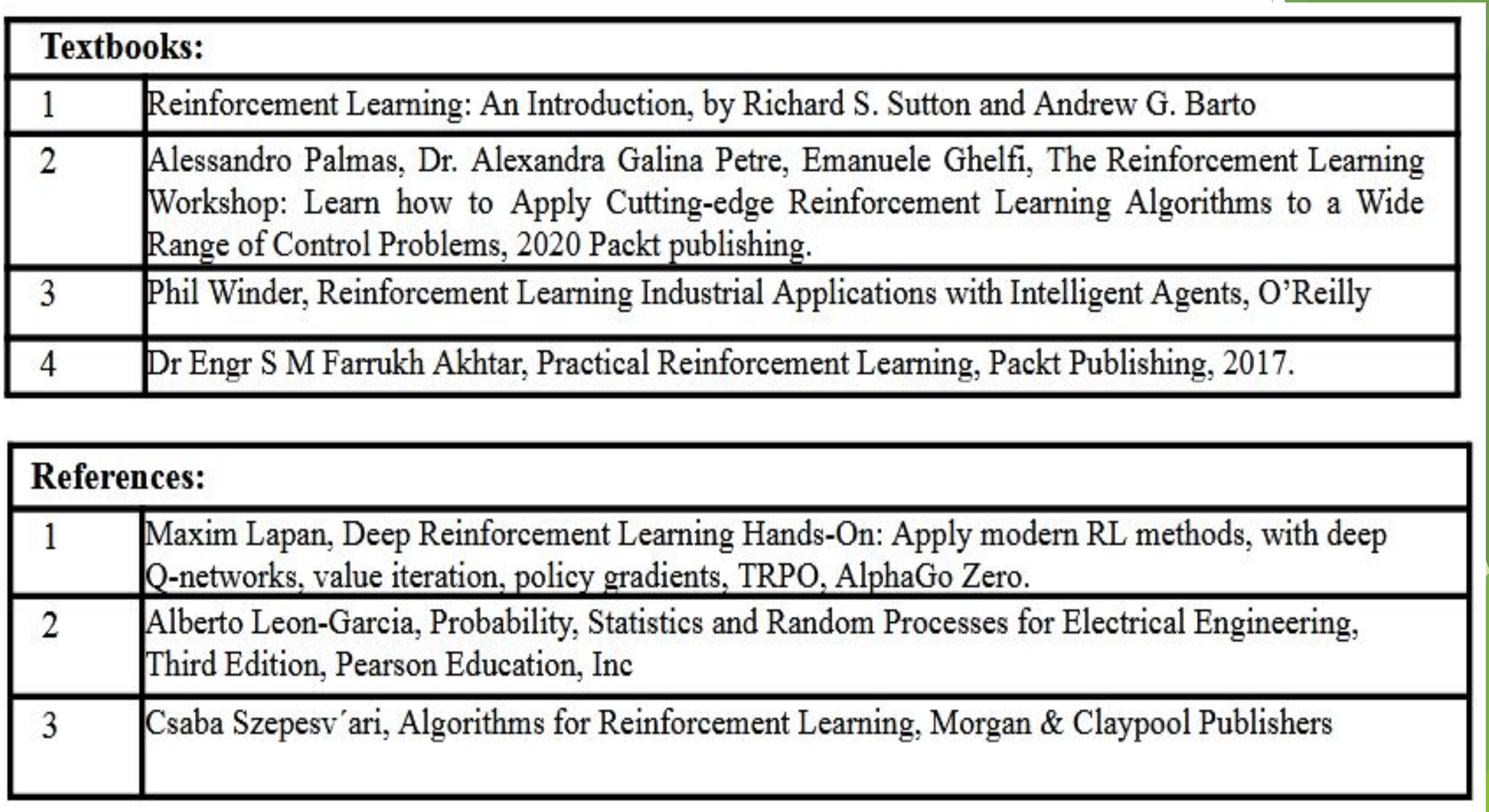
Syllabus

Module	Content	Hrs
1	Introduction to Reinforcement Learning:	4
1.1	Reinforcement Learning: Key features and Elements of RL, Types of RL, rewards. Reinforcement Learning Algorithms: Q-Learning, State Action Reward State action (SARSA)	
2	Bandit problems and online learning:	7
2.1	An n-Armed Bandit Problem, Action-Value Methods Tracking a Non stationary Problem, Optimistic Initial Values Upper-Confidence-Bound Action Selection Gradient Bandit	
3	Markov Decision Processes:	7
3.1	The Agent–Environment Interface, The Agent–Environment Interface, Goals and Rewards, Returns, Markov properties, Markov Decision Process, Value Functions and Optimal Value Functions	
4	Dynamic Programming:	7
4.1	Policy Evaluation (Prediction), Policy Improvement, Policy Iteration, Value Iteration, Asynchronous Dynamic Programming, Generalized Policy Iteration	
5	Monte Carlo Methods and Temporal-Difference Learning	8
5.1	Monte Carlo Prediction, Monte Carlo Estimation of Action Values, Monte Carlo Control, TD Prediction, TD control using Q-Learning	
6	Applications and Case Studies	6
6.1	Elevator Dispatching, Dynamic Channel Allocation, Job-Shop Scheduling	



Textbooks:	
1	Reinforcement Learning: An Introduction, by Richard S. Sutton and Andrew G. Barto
2	Alessandro Palmas, Dr. Alexandra Galina Petre, Emanuele Ghelfi, The Reinforcement Learning Workshop: Learn how to Apply Cutting-edge Reinforcement Learning Algorithms to a Wide Range of Control Problems, 2020 Packt publishing.
3	Phil Winder, Reinforcement Learning Industrial Applications with Intelligent Agents, O'Reilly
4	Dr Engr S M Farrukh Akhtar, Practical Reinforcement Learning, Packt Publishing, 2017.

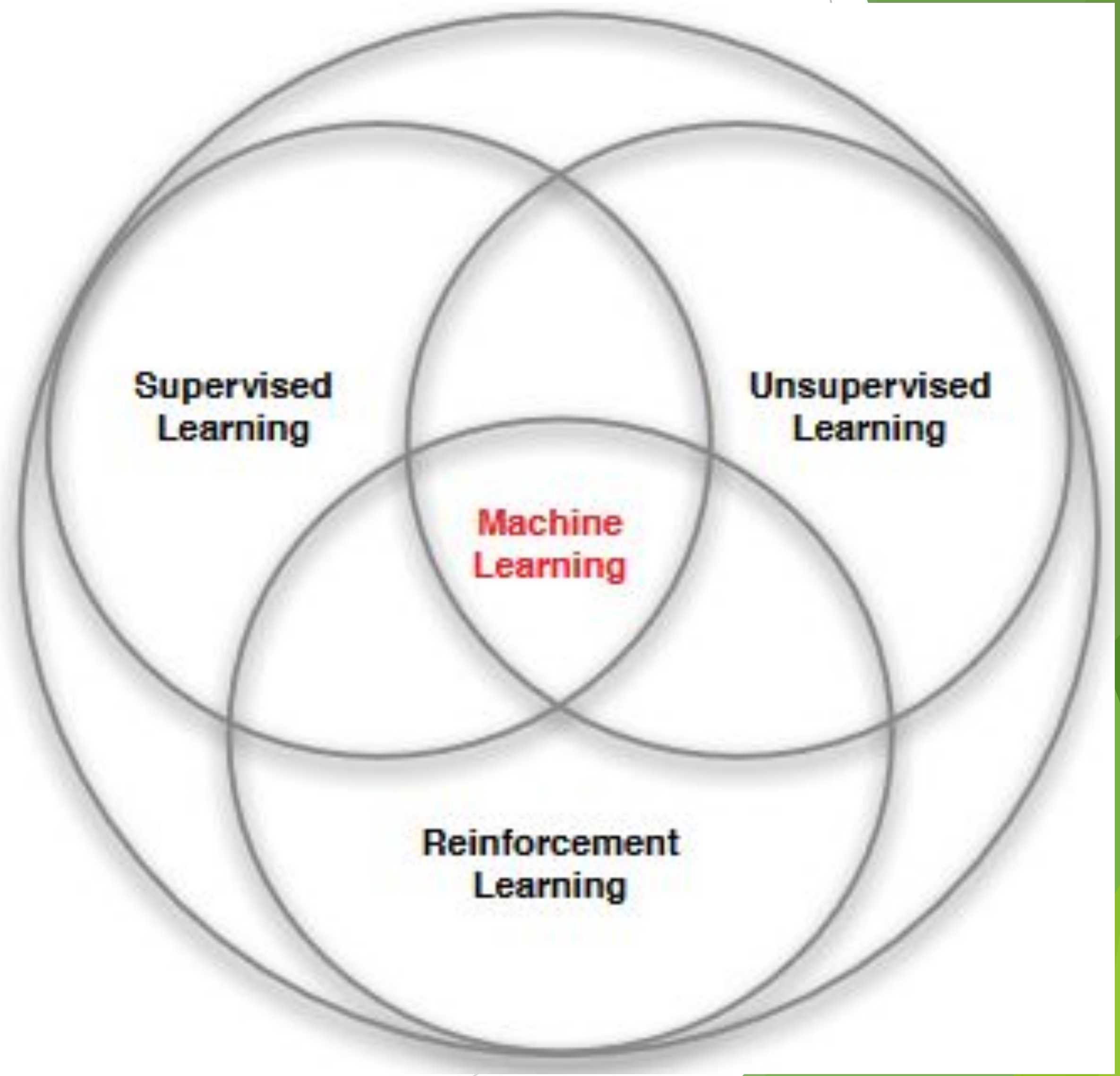
References:	
1	Maxim Lapan, Deep Reinforcement Learning Hands-On: Apply modern RL methods, with deep Q-networks, value iteration, policy gradients, TRPO, AlphaGo Zero.
2	Alberto Leon-Garcia, Probability, Statistics and Random Processes for Electrical Engineering, Third Edition, Pearson Education, Inc
3	Csaba Szepesv´ari, Algorithms for Reinforcement Learning, Morgan & Claypool Publishers

What is learning?

- Learning takes place as a result of interaction between an agent and the world, the idea behind learning is that
  - Percepts received by an agent should be used not only for acting, but also for improving the agent's ability to behave optimally in the future to achieve the goal

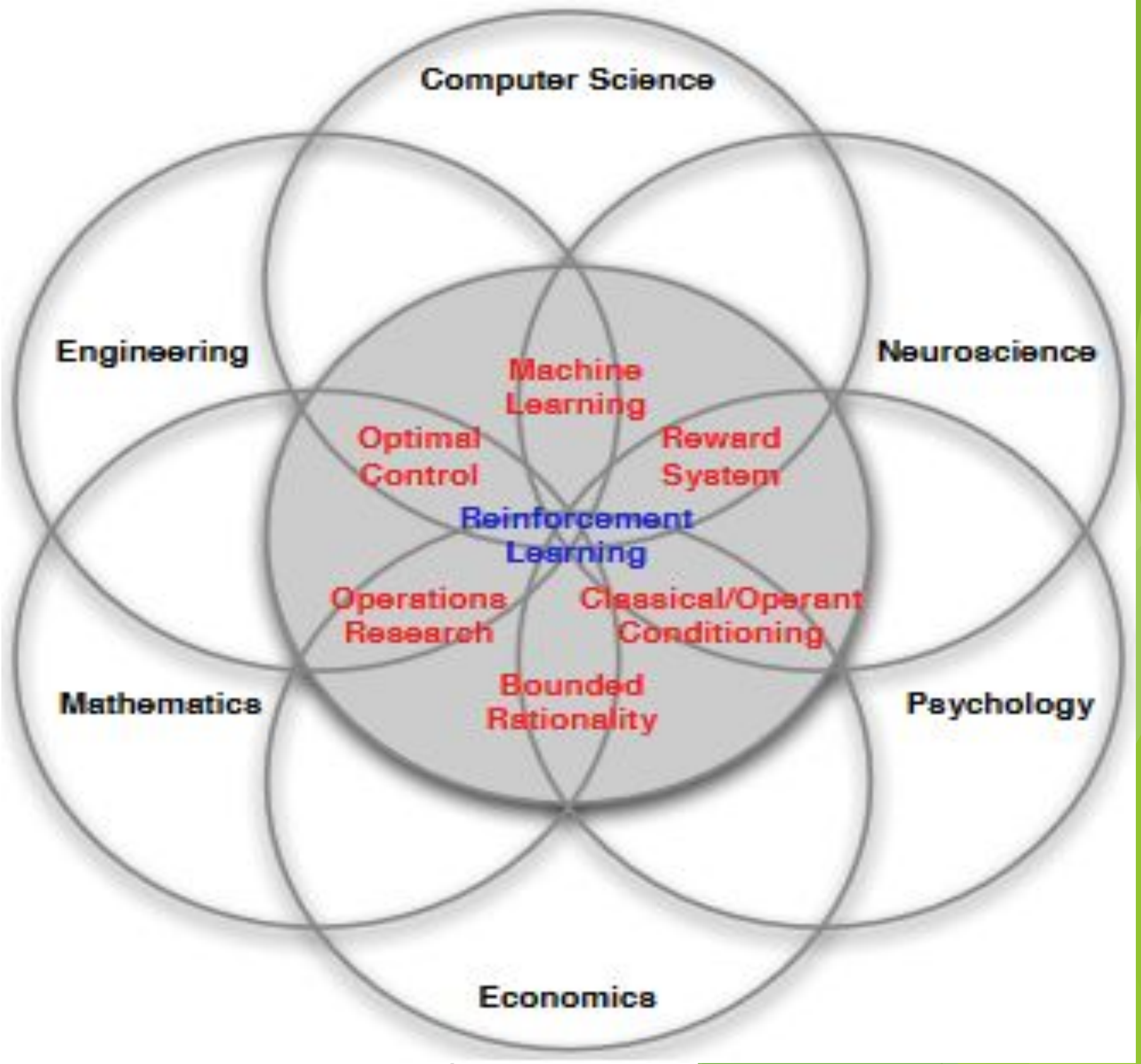
Learning types

- Supervised learning
- Unsupervised learning
- Reinforcement learning

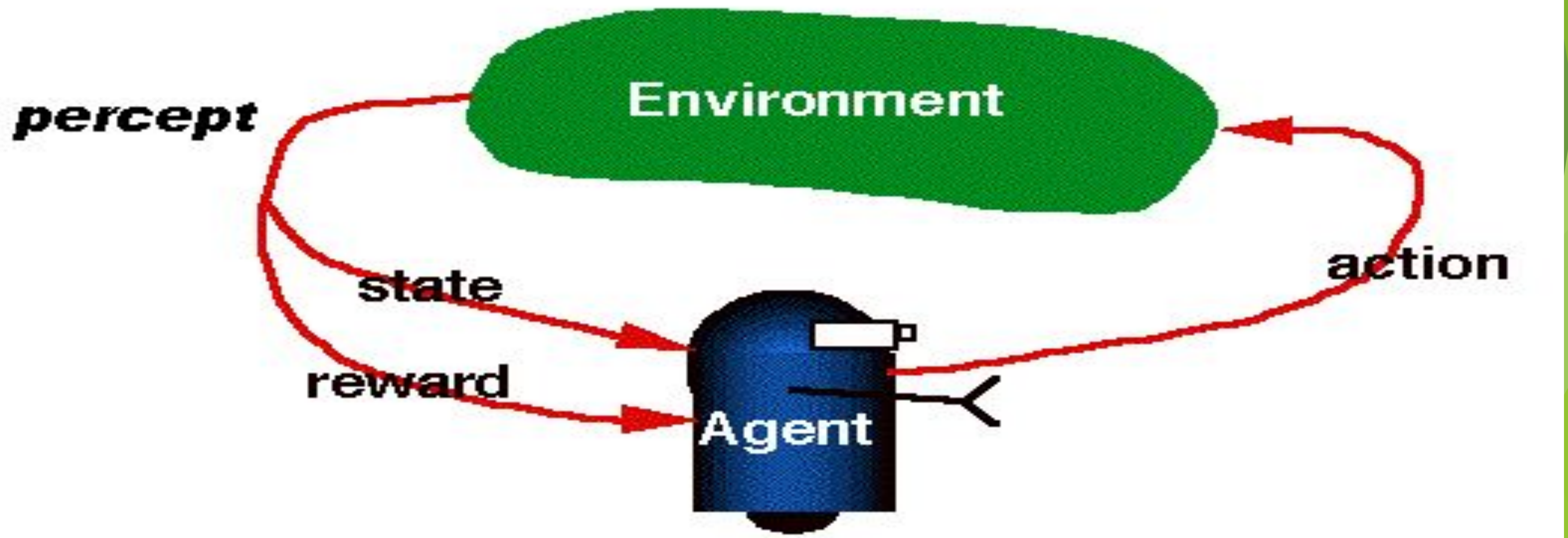


Reinforcement learning

- Task
  - Learn how to behave successfully to achieve a goal while interacting with an external environment
  - Learn via experiences!
- Examples
  - Game playing: player knows whether it win or lose, but not know how to move at each step
  - Control: a traffic system can measure the delay of cars, but not know how to decrease it



RL is learning from interaction



Cont…

- **Reinforcement Learning is a feedback-based Machine learning technique** in which an agent learns to behave in an environment by performing the actions and seeing the results of actions. For each **good action**, the agent gets **positive feedback**, and for each **bad action**, the agent gets **negative feedback** or penalty
- In Reinforcement Learning, the agent **learns automatically** using feedbacks without any labeled data, unlike supervised learning. Since there is no labeled data, so the agent is bound to **learn by its experience** only
- RL solves a specific type of **problem where decision making is sequential**, and the **goal is long-term**, such as game-playing, robotics, etc. The agent interacts with the environment and **explores** it by itself
- Reinforcement learning problems involve learning what to do—how to map situations to actions—so as to **maximize a numerical reward** signal
- The **primary goal** of an agent in reinforcement learning is to improve the performance by getting the **maximum positive rewards**

Advantages of RL

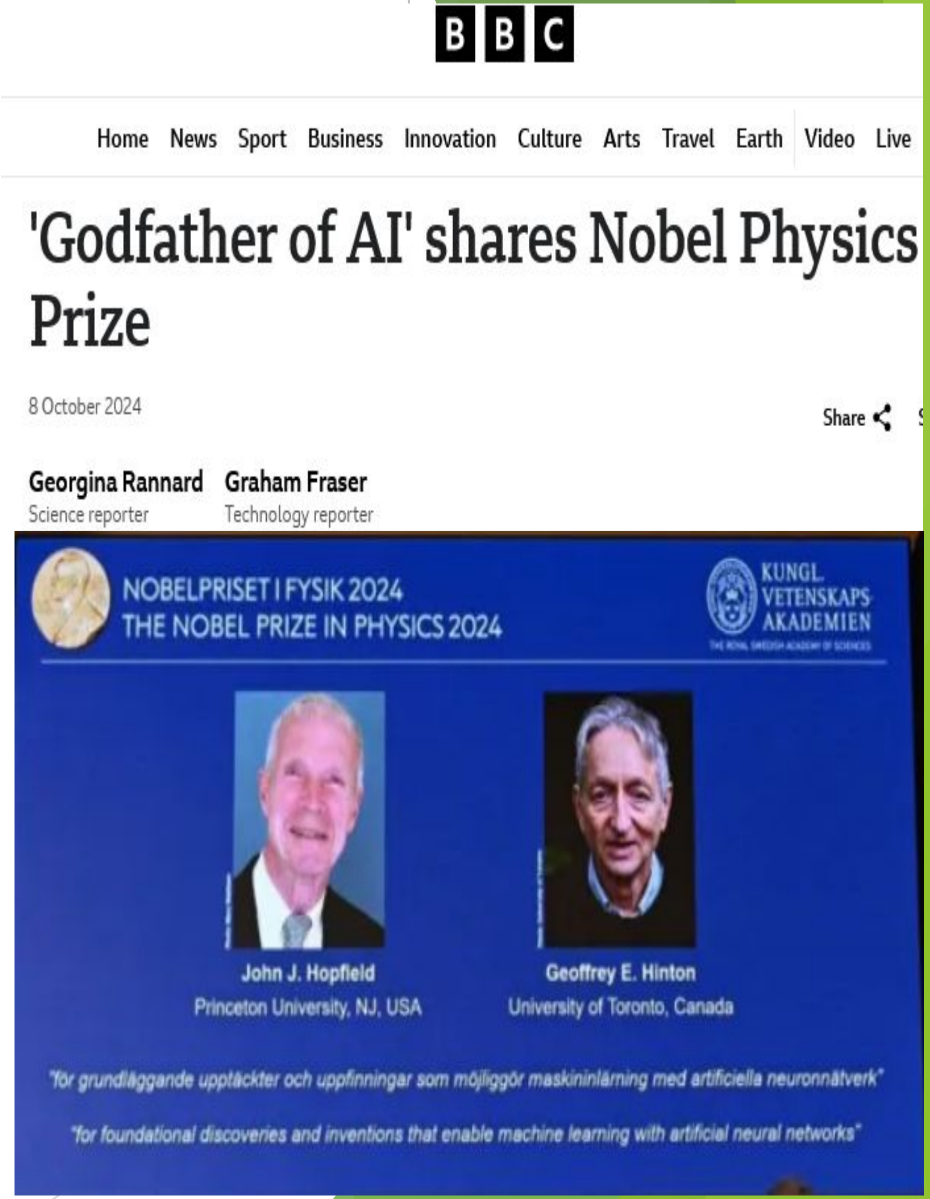
- RL can be used to solve very complex problems that cannot be solved by conventional techniques
- The model can correct the errors that occurred during the training process
- In RL, training data is obtained via the direct interaction of the agent with the environment
- RL can handle environments that are non-deterministic / stochastic, outcomes of actions not always predictable
- This is useful in real-world applications where the environment may change over time or is uncertain
- RL is a flexible approach that can be combined with other machine learning techniques, such as deep learning, to improve performance

Disadvantages of RL

- RL is not preferable to use for solving simple problems
- RL needs a lot of data and a lot of computation
- RL is highly dependent on the quality of the reward function
- If the reward function is poorly designed, the agent may not learn the desired behavior

AI Ethics

- Prof **Hinton**, **76**, is a professor at University of **Toronto** in Canada.
- Prof Hinton's pioneering **research** on **neural networks** paved the way for current AI systems like **ChatGPT**.
- He **resigned** from **Google** in **2023**, and has warned about the dangers of **machines** that could **outsmart** humans.
- “It’s going to be like the **Industrial Revolution** - but instead of our **physical capabilities**, it’s going to exceed our **intellectual capabilities**," he said.
- "We're **biological systems** and these are **digital systems**. And the big difference is that with digital systems, you have many **copies** of the same set of weights, the same model of the world.
- "And all these copies can learn separately but **share** their **knowledge instantly**. So it's as if you had **10,000** people and whenever one person learnt something, everybody automatically knew it. And that's how these **chatbots** can know so much more than any **one person**.“
- In reply, he said he would do the **same work again**, "but I **worry** that the overall consequences of this might be **systems** that are **more intelligent** than us that might eventually take **control**".
- He was “very worried about AI taking **lots** of mundane **jobs**”.
- He added that while AI would increase **productivity** and **wealth**, the money would go to the **rich** “and not the people whose jobs get lost and that’s going to be very bad for society”.


BBC
Home
News
Sport
Business
Innovation
Culture
Arts
Travel
Earth
Video
Live
Share

'Godfather of AI' shares Nobel Physics Prize

8 October 2024

Georgina Rannard
Science reporter

Graham Fraser
Technology reporter

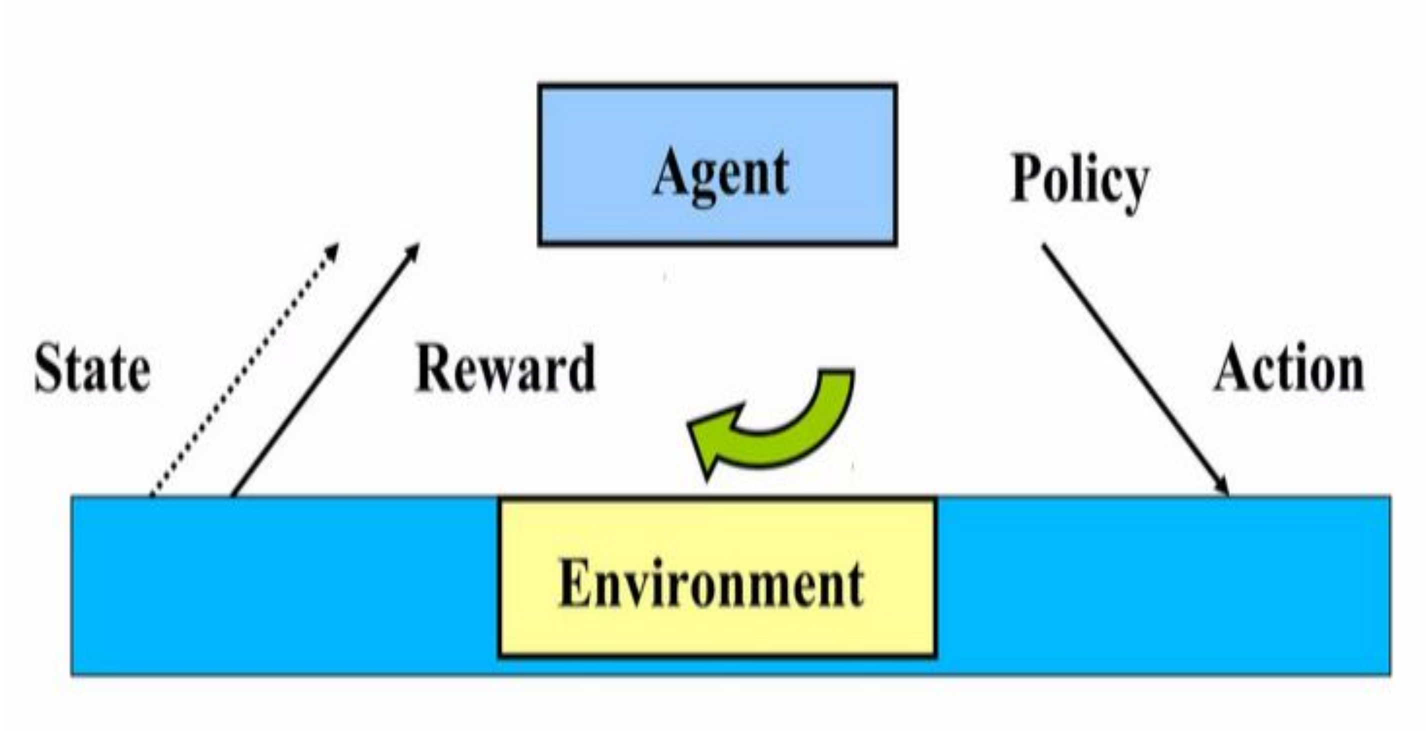
NOBELPRISET I FYSIK 2024
THE NOBEL PRIZE IN PHYSICS 2024
KUNGL. VETENSKAPS AKADEMIEN
John J. Hopfield
Princeton University, NJ, USA
Geoffrey E. Hinton
University of Toronto, Canada
"för grundläggande upptäckter och uppfinningar som möjliggör maskininlärning med artificiella neuronnätverk"
"for foundational discoveries and inventions that enable machine learning with artificial neural networks"

Key Features of Reinforcement Learning

- The key characteristics/ features of reinforcement learning problems are:
  - **Closed-loop System**: The agent's actions influence the environment and the environment's feedback (like rewards) influences the agent’s future decisions
  - **No Direct Instructions**: The agent doesn't receive explicit instructions on what actions to take; it must figure out the best actions through trial and error.
  - **Delayed Consequences**: The effects of the agent's actions, including rewards, might not be immediate and can unfold over a long period

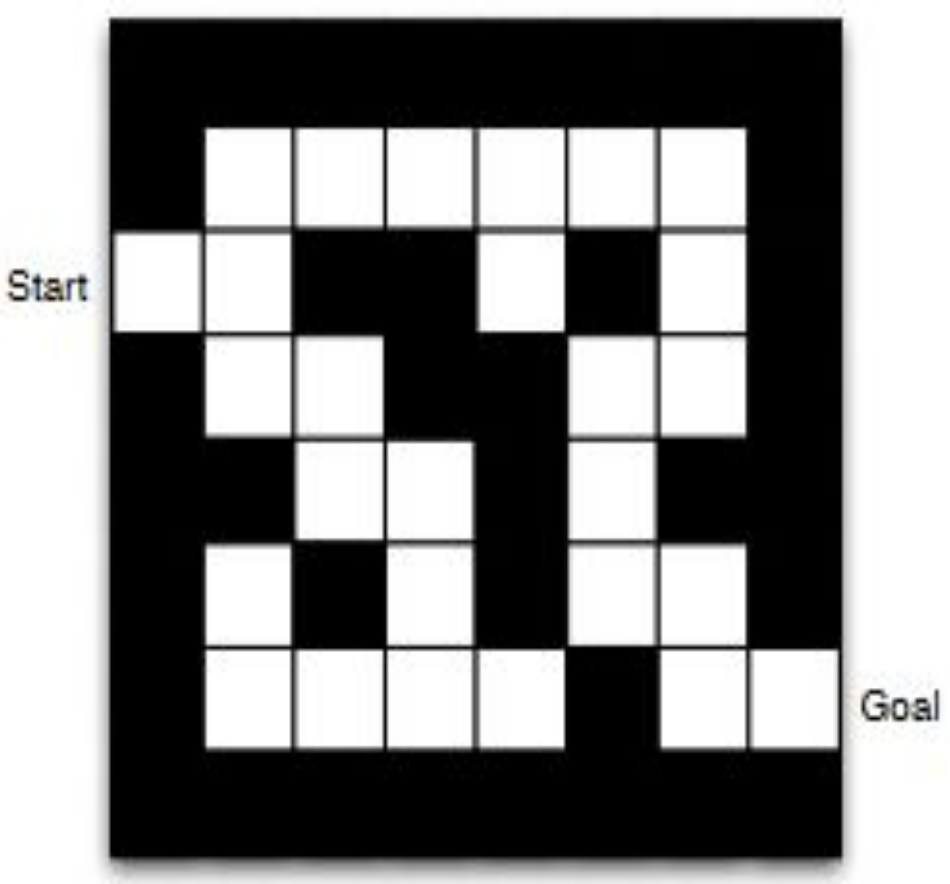
Elements of Reinforcement Learning

- The main elements of Reinforcement Learning are:
  - Agent
  - Environment
  - Action
  - State
  - Reward Signal
  - Policy
  - Value Function
  - Model



Maze Example

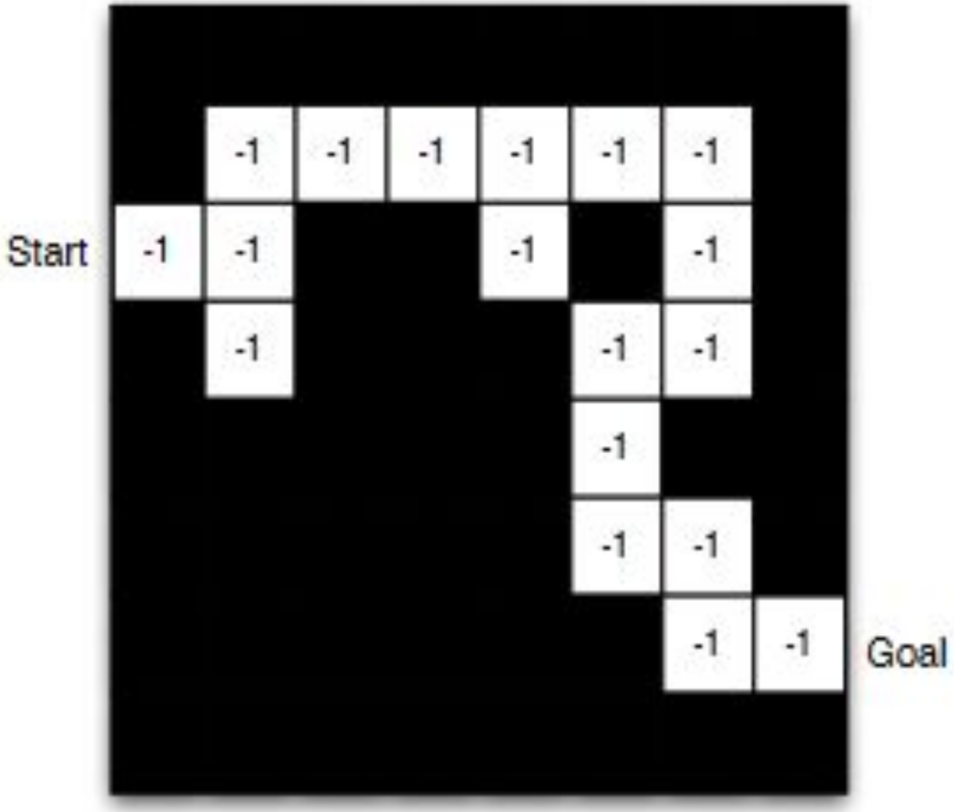
- Agent: An entity that can perceive/explore the environment and act upon it
- Environment: A situation in which an agent is present or surrounded by. In RL, we assume the stochastic environment, which means it is random in nature
- Action: Actions are the moves taken by an agent within the environment
- State: State is a situation returned by the environment after each action taken by the agent



Actions: N, E, S, W
States: Agent’s location

Reward Signal

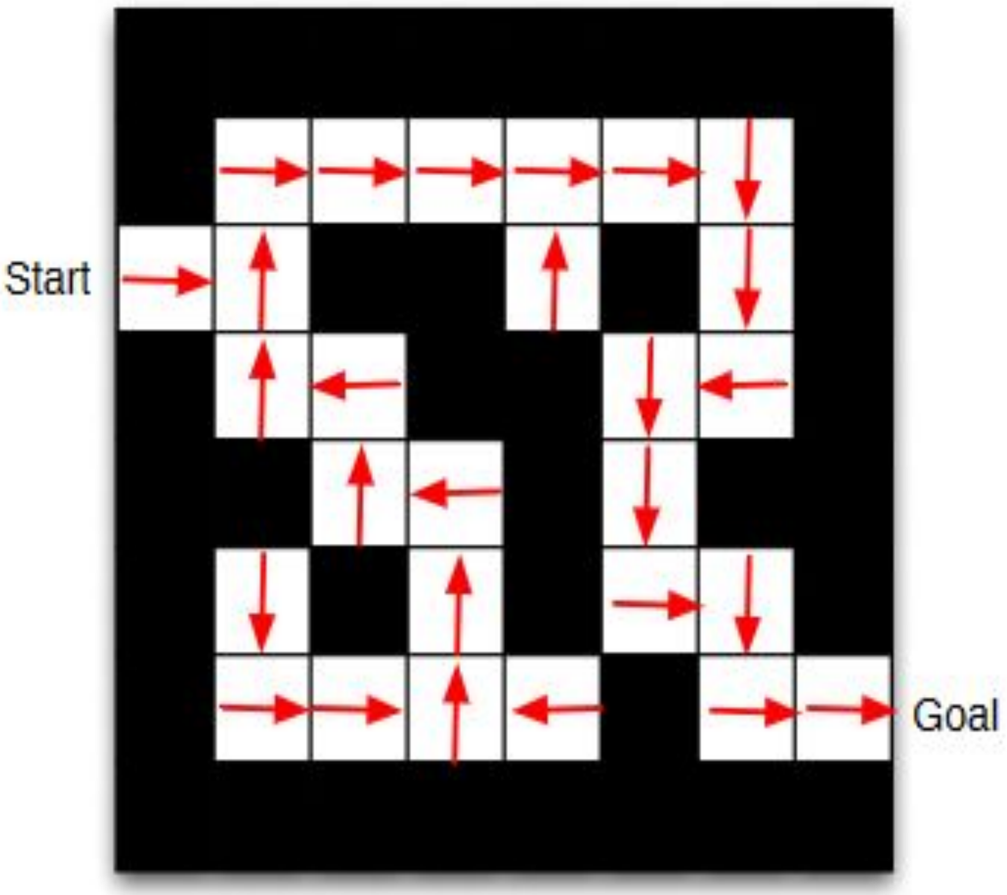
- The goal of reinforcement learning is defined by the reward signal
- At each state, the environment sends an immediate signal to the learning agent, and this signal is known as a reward signal
- These rewards are given according to the good and bad actions taken by the agent
- The agent's main objective is to maximize the total number of rewards for good actions
- The reward signal can change the policy, such as if an action selected by the agent leads to low reward, then the policy may change to select other actions in the future



Rewards: -1 per time-step

Policy

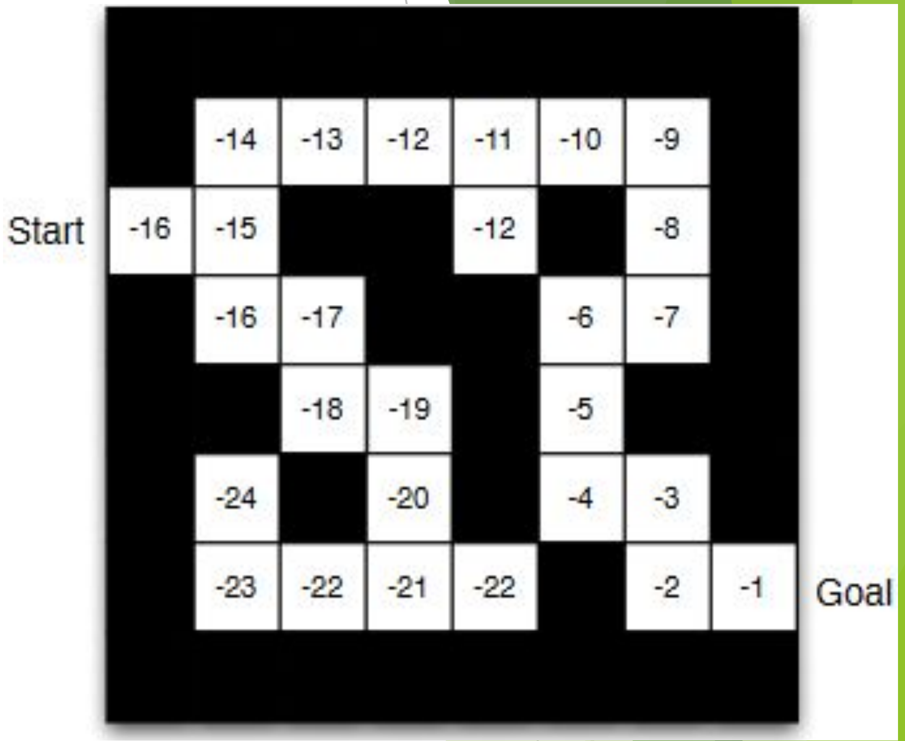
- A policy can be defined as a way how an agent behaves at a given time
- It maps the perceived states of the environment to the actions taken on those states
- A policy is the core element of the RL as it alone can define the behavior of the agent
- In some cases, it may be a simple function or a lookup table, whereas, for other cases, it may involve general computation as a search process
- It could be deterministic or a stochastic policy:
    - For deterministic policy: a = π(s)
    - For stochastic policy: π(a | s) = P[At =a | St = s]



Arrows represent policy π(s) for each state s

Value Function

- The value function gives information about how good the situation and action are and how much reward an agent can expect
- A reward indicates the immediate signal for each good and bad action, whereas a value function specifies the good state and action for the future
- The value function depends on the reward as, without reward, there could be no value
- The goal of estimating values is to achieve more rewards
- The Bellman equation: V(s) = max [R(s,a) + γV(s')]
  - V(s)= value calculated at a particular point
  - R(s,a) = Reward at a particular state s by performing an action
  - γ = Discount factor
  - V(s') = The value at the previous state



		-14	-13	-12	-11	-10	-9		
Start	-16	-15			-12		-8		
		-16	-17			-6	-7		
			-18	-19		-5			
		-24		-20		-4	-3		
		-23	-22	-21	-22		-2	-1	Goal

Numbers represent value V(s) of each state s

Example 1(May_2024) 10M

- Imagine a simple game where a player controls a character to navigate through a maze to reach a treasure chest. The player receives a reward of +10 points upon reaching the treasure chest and -1 point for each move taken. Assume the player starts at the entrance of the maze.
- If the player reaches the treasure chest in 15 moves, what is their total reward?
- If the player reaches the treasure chest in 20 moves, what is their total reward?
- What is the maximum possible reward the player can achieve in this game?
- What would be the reward if the player gets stuck in the maze indefinitely?

Model

- The last element of reinforcement learning is the model, which mimics the behavior of the environment
- With the help of the model, one can make inferences about how the environment will behave
- Such as, if a state and an action are given, then a model can predict the next state and reward

Approach for Implementation of RL

- **Value-based:** The value-based approach is about to find the optimal value function, which is the maximum value at a state under any policy. Therefore, the agent expects the long-term return at any state(s) under policy π
- **Policy-based:** Policy-based approach is to find the optimal policy for the maximum future rewards without using the value function. In this approach, the agent tries to apply such a policy that the action performed in each step helps to maximize the future reward. The policy-based approach has mainly two types of policy:
    - Deterministic: The same action is produced by the policy (π) at any state
    - Stochastic: In this policy, probability determines the produced action
- **Model-based:** In the model-based approach, a virtual model is created for the environment, and the agent explores that environment to learn it. There is no particular solution or algorithm for this approach because the model representation is different for each environment

Reinforcement Learning Algorithms

- Q-Learning
- State Action Reward State action (SARSA)

Q-Learning

- Q-Learning is a model-free, off-policy reinforcement learning algorithm
- It operates on the principle of maximizing the expected cumulative future reward over time
- Q-value (action-value): The Q-value, Q(s,a), represents the expected future reward for an agent when taking a particular action ‘a’ in a given state ‘s’

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\ (r_{t+1} + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t))$$

- Off-policy: The agent does not need to follow the policy that it is currently learning
- Greedy Policy: In Q-Learning, the policy is updated to choose actions that maximize the Q-value

Strengths and Weaknesses

- **Strengths:**
  - Simple and efficient: Q-Learning is a straightforward algorithm and has been widely applied to problems like game-playing and robotics
  - Model-free: Does not require a model of the environment
- **Weaknesses:**
  - Exploration-exploitation trade-off: Balancing between exploring unknown states and exploiting known ones is challenging
  - State and action space explosion: Q-Learning struggles with large state spaces, as it requires a table to store Q-values for each state-action pair

SARSA (State-Action-Reward-State-Action)

- SARSA is model-free, on-policy reinforcement learning algorithm
- On-policy: SARSA updates the Q-values based on the actually action taken by the agent

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\ (r_{t+1} + \gamma\ Q(s_{t+1},\ a_{t+1}) - Q(s_t,a_t))$$

- The policy is updated according to the current behavior of the agent and the agent's exploration is guided by this policy
- Exploration-exploitation: SARSA usually uses an epsilon-greedy policy, balancing exploration and exploitation

Strengths and Weaknesses

- **Strengths:**
  - On-policy: SARSA updates the Q-value based on the actual action taken and follows the current policy, making it more stable in environments where exploration is important
  - Simplicity: SARSA is simple and easy to implement
  - Exploration: The policy is refined as the agent explores, so the learning process is directly influenced by the agent’s behavior
- **Weaknesses:**
  - Less optimal : Since SARSA uses the current policy to update the Q-values, it might converge to suboptimal solutions
  - Slow convergence: Since SARSA is on-policy and the agent’s behavior is involved in the update, it may take longer to converge

Differences Between Q-Learning and SARSA

Aspect	Q-Learning	SARSA
Type	Off-policy	On-policy
Update Rule	Updates based on the greedy action (max Q-value) of the next state	Updates based on the action actually taken in the next state
Policy Used for Updates	Uses the policy that maximizes Q-value	Uses the current policy
Exploration Strategy	Typically epsilon-greedy, but more focused on exploitation in the long run	Typically epsilon-greedy, with exploration based on the current policy
Convergence	Faster convergence	May converge slower as it follows the actual actions taken
Suitability	Better when exploration can be separated from the learning phase	More suitable for environments where exploration is part of the learning process