Build COMPETENCY
across your TEAM

Reinforcement Learning using Q Learning

Chandrashekhar Deshpande

Google Cloud Platform

Windows Azure

amazon web services

APACHE kafka
A distributed streaming platform

Apache Software Foundation

APACHE Spark

Java

hadoop

Agenda

1. What is Reinforcement Learning?
2. Few important use-cases
3. How it is different than Supervised and Unsupervised Machine Learning?
4. Types and approaches
5. Common Algorithms in RL
6. How does it work?
7. The Q-Learning
8. The Q-Table
9. Implementation insight in Python

What is Reinforcement Learning?

A Machine Learning that trains the model to come to an optimal solution for a problem by taking decision by itself.

A type of Machine Learning that enables an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences.

The feedback rewards score on desired behavior or punishment on undesired one.

Is all about making decisions sequentially.

Rather than the typical ML problems such as Classification, Regression, Clustering and so on, RL is most commonly used to solve a different class of real-world problems, such as a Control task or Decision task, where you operate a system that interacts with the real world.

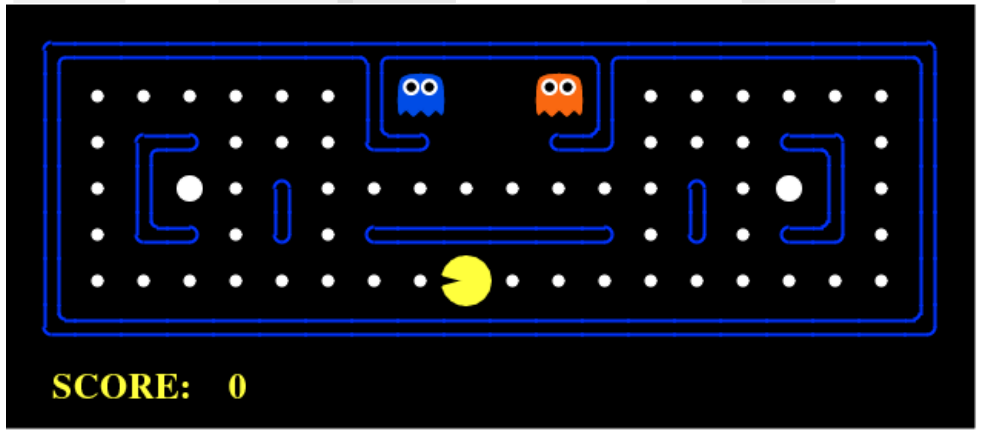
Learning from trial and error: A baby can touch a fire or milk and then learn from positive or negative reinforcement. Milk is good so mind registers score as +5 and fire is bad, so registers score as -5.

What is Reinforcement Learning?

In the Pac Man game, the goal of Pac Man is to eat the food without getting killed by ghost.

Pac Man receives a reward (As a positive score) for eating a food and punishment (May be as a negative score) when killed by a ghost which ultimately ends a game.

Pac Man must choose a path to gain rewards keeping itself away from ghosts.



Few Use Cases

Enterprise Resource Management:

- Reinforcement learning algorithms can allocate limited resources to different tasks if there is an overall goal it is trying to achieve. A goal in this circumstance would be to save time or conserve resources.

Robotics:

- Reinforcement Learning can provide robots in an Industry with the ability to learn tasks a human teacher cannot demonstrate, to adapt a learned skill to a new task or to achieve optimization despite a lack of analytic formulation available. Operating drones and auto-driven vehicles.

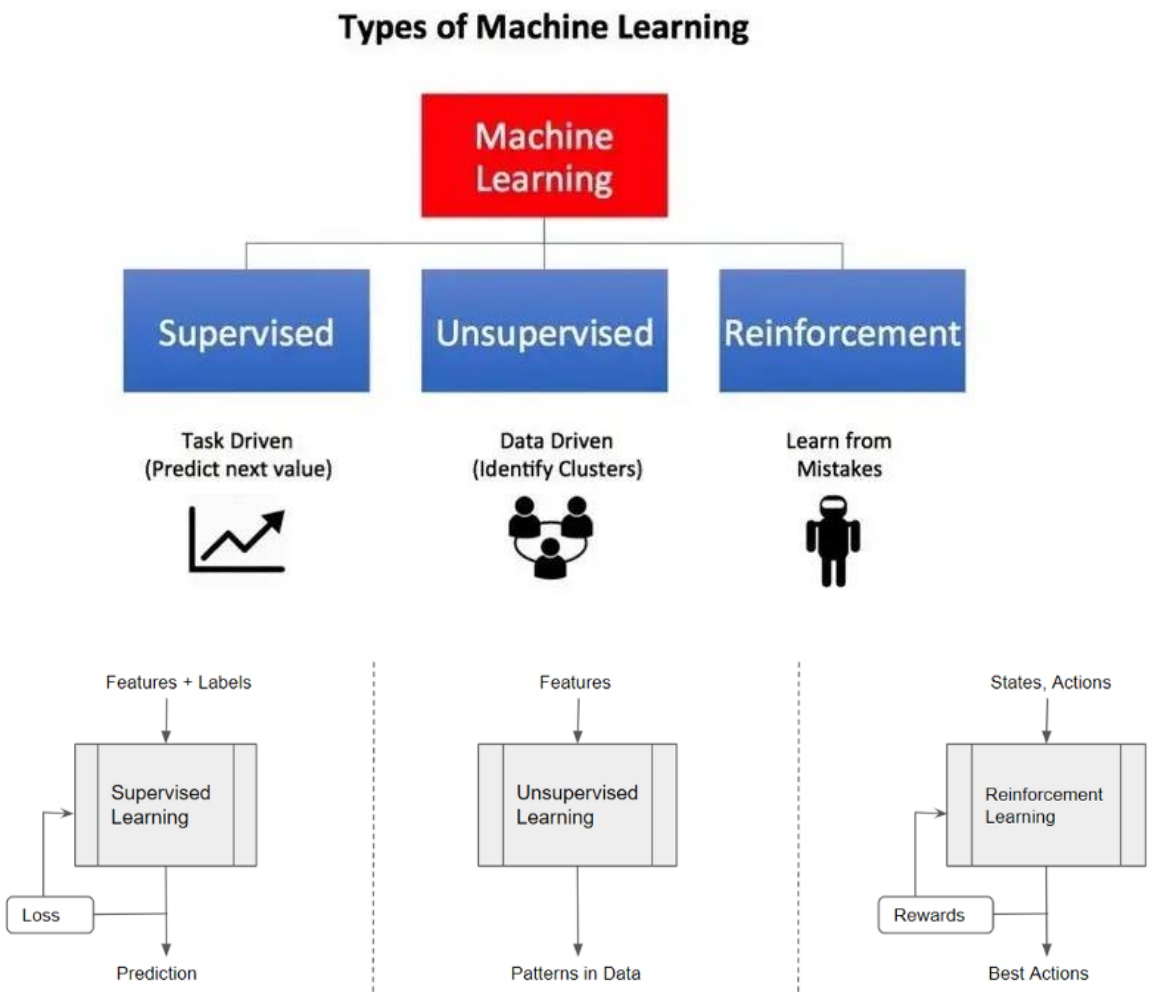
Personalized Recommendations:

- Reinforcement learning algorithms can allocate positive scores for likings/positive sentiments in tweets and obviously negative scores for disliking/negative sentiments and accordingly try to achieve overall goal of best choice. Also, Ad Recommendation system, Ads are shown based on previous purchase or recent clicks.

Other use cases…

- Operations Research
- Game theory
- Information Theory
- Control Theory
- Genetic Algorithms
- Swarm Intelligence
- Simulation based Optimization

How RL is different than Supervised ML?



The Reinforcement Learning Vs Supervised Learning

Both supervised and reinforcement learning use mapping between input and output.

The feedback provided to the agent is correct set of actions for performing a task, reinforcement learning uses rewards and punishments as signals for positive and negative behavior.

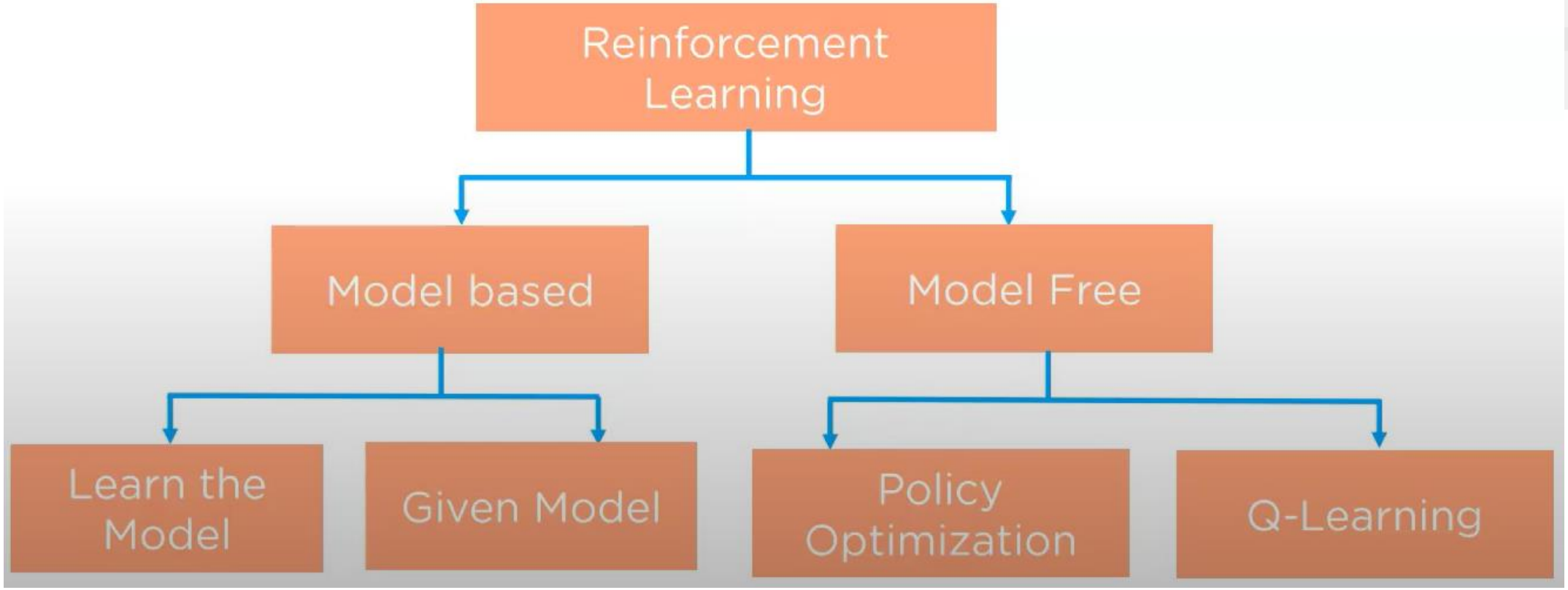
Object Recognition Vs. Chess Game

The Reinforcement Learning Vs Unsupervised Learning

The unsupervised learning is to find similarities and differences between data points.

The RL is entirely different in goals to find a suitable action model that would maximize the total cumulative reward of the agent.

Types of Reinforcement Learning



Types of Reinforcement Learning based on Score

The Positive:

- Increases the strength and the frequency of the behavior and impacts positively on the action taken by the agent.
- It helps you to maximize performance and sustain change for a more extended period.

The Negative:

- Strengthens the behavior that occurs because of a negative condition which should have stopped or avoided.

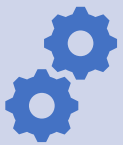
Approaches in RL

Value Based

Maximize a value function so that Agent can expect a long-term return of the current state.

Policy Based

To produce a policy to gain a maximum reward in future against the action performed.



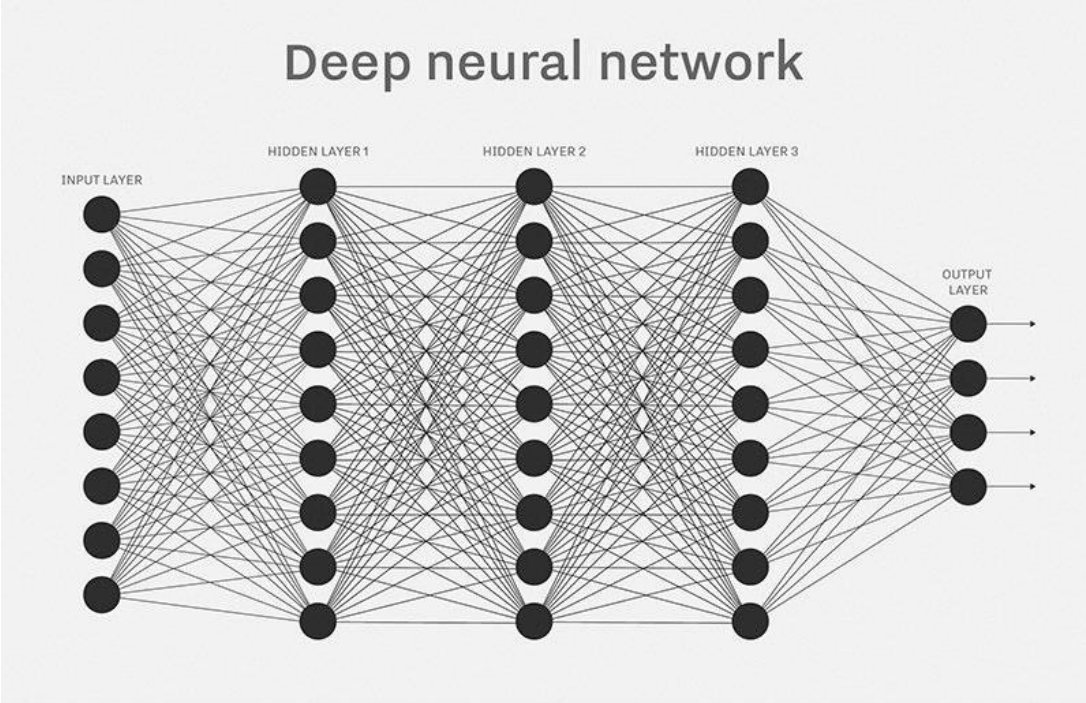
Model Based

Create a virtual model for each environment so that agent learns to perform best in a specific environment.

Common algorithms in RL

The algorithms are different mainly due to their strategies for exploring their environments.

- State Action Rewards State Action(SARSA): Starts by giving an agent a Policy.
- Q-Learning: Agent receives no policy so; its exploration of environment is more self-directed.
- Deep Q-Networks: Combines neural network and RL Techniques.



How does it work?

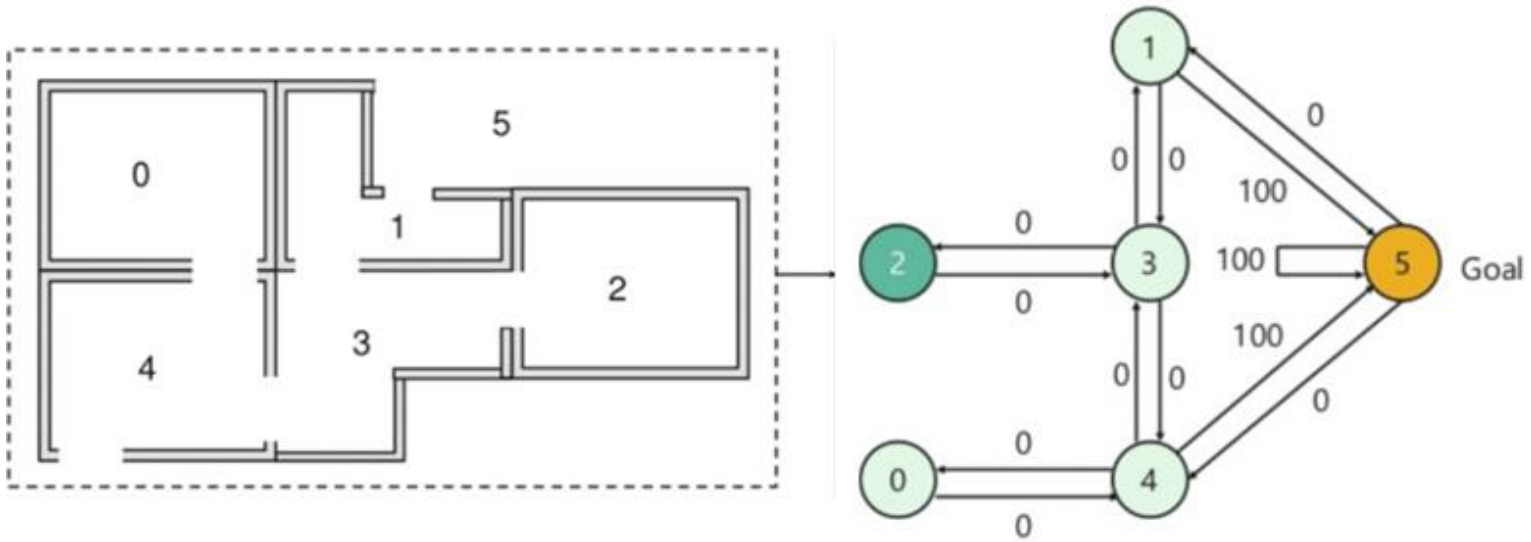
Key terms	
The Agent In the Pac Man game, the agent is Pac Man itself.	• Its goal is to eat food in the grid while avoiding the ghost on its way. This can be a Robot or a model to build and train using RL.
The Environment The Physical world in which agent operates.	• In Pac Man, the grid world is an interactive environment in which Agent operates. For a Robot, the terrain and surrounding factors (Wind, Friction) over which Robot has to navigate.
The Rewards It's a feedback from the Environment.	• Agent receives a feedback on eating a food and punishment when ghost kills it thereby losing the game.
The Action Agent interacts with Environment	• Its an action agent takes to interact with Environment. The Robot can turn right, left, forward, backward, bend, raise its hand.

How does it work?

Key terms	
The Rewards It's a feedback from the Environment.	• Agent receives a feedback on eating a food and punishment when ghost kills it thereby losing the game.
The State A current state of the agent.	• A location of the agent in the grid world.
A Value A future reward that an agent would receive by taking an action in a particular state.	• An agent is rewarded when it eats a food.
The Policy A method to map agent's state to action. Its also a probability that tell it the odd of certain actions resulting in rewards or beneficial states.	• Eat and move ahead are the methods/actions on an agent.
The Exploration	• In order to build an optimal policy, the agent faces the dilemma of exploring new states while maximizing its overall reward at the same time.

How may it work?

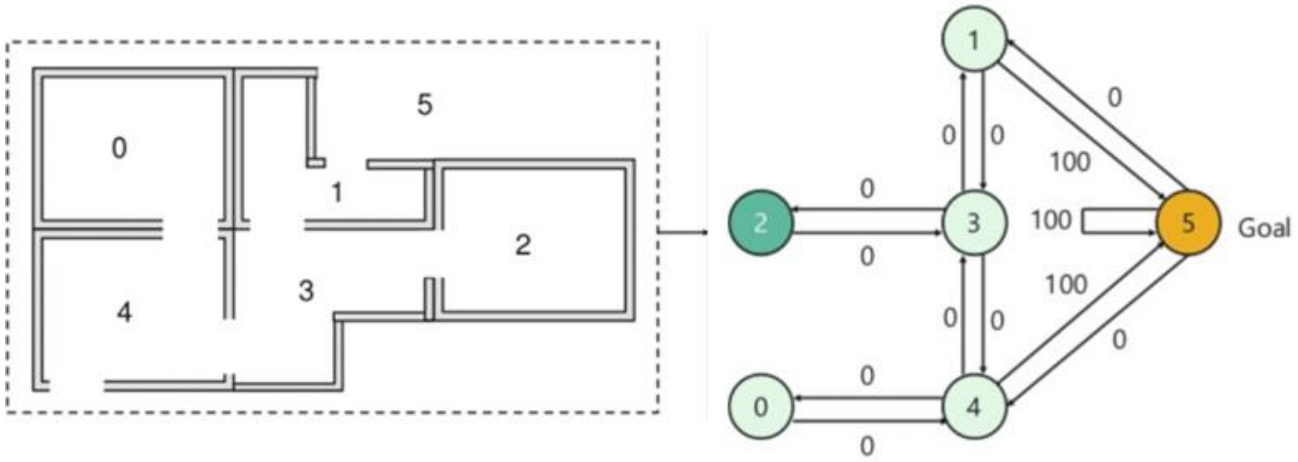
- There are 5 rooms in a building numbered as 0 to 4.
- Outside of the building is numbered as 5.
- Doors of room 1 and 4 are opening towards outside (5).
- Doors directly leading to Goal have reward 100. All other doors have reward of 0.
- As doors are two ways, bi-directional access is shown.



How may it work?

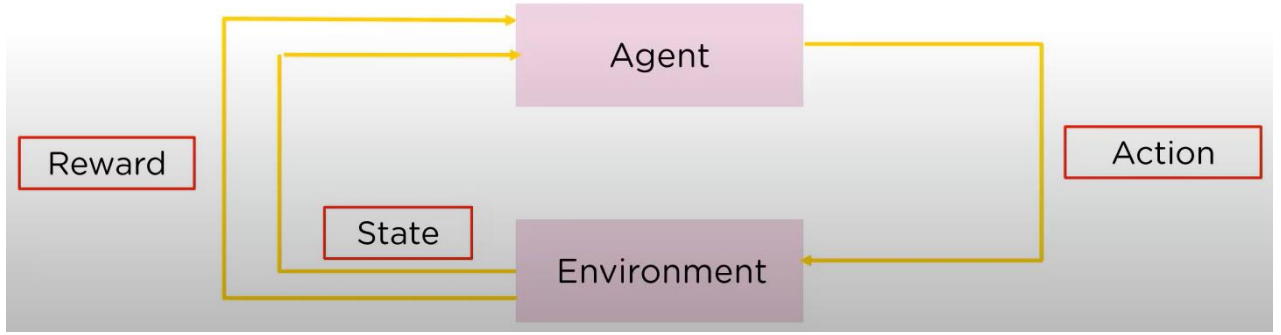
- Key terms
  - An Agent: A person trying to come outside building.
  - The State: An agent reaching a specific room represents a state of an Agent.
  - The Action: Moving from one room to another. Arrows are showing Action.
  - The Environment: Whole building along with outside represents Environment
  - The rewards: Movement across rooms may reward either 0 or 100.
  - The Value: The score either 0 or 100.

- Assume initial state as Room 2.
  - From State 2
    - to State 3(Score 0)
  - From State 3
    - To State 1(Score 0)
      - From State 1 to 5 (Score 100)
    - To State 4(Score 0)
      - From State 4 to 5 (Score 100)
      - From State 4 to 0 (Score 0)

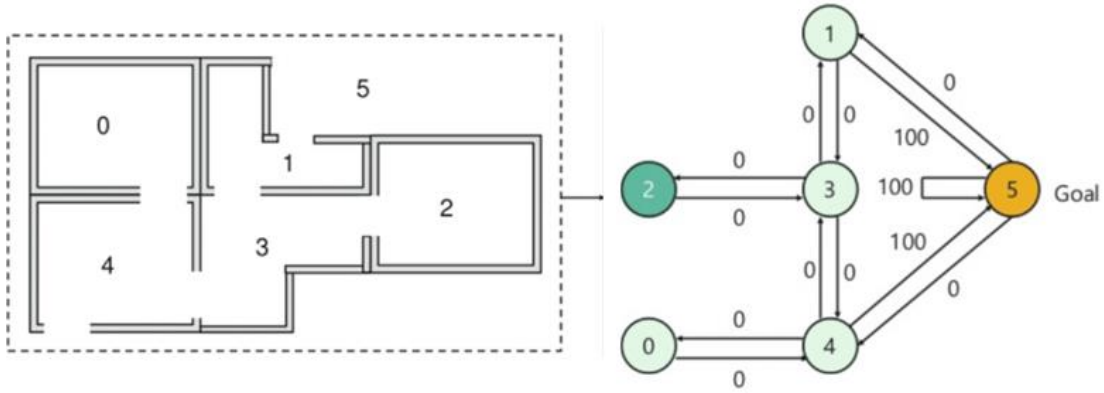


The Q-Learning

- A model free but value-based method which decides the best action an agent should take. The best action obviously is decided by maximum score. It chooses this action at random to maximize the reward.
- Based on Stochastic Transition (Based on probability of every action).
- Uses a Lookup table called as a Q-Table (State-Action Values).



Q-Table



Action → / Rooms ↓	0	1	2	3	4	5
0	NA	NA	NA	NA	0	NA
1	NA	NA	NA	0	NA	100
2	NA	NA	NA	0	NA	NA
3	NA	0	0	NA	0	NA
4	NA	NA	NA	NA	0	100
5	NA	0	NA	NA	0	NA

LAB and Implementation

Q-Learning using Python

Refer QLearn01.ipynb

Steps followed…

1. Import necessary libraries
2. Create an environment
3. Initialize Q-Table
4. Initialize Hyper Parameters
5. Create Episodes, iterate, calculate rewards and performance
6. Observe the performance.

References

- Online reference:
  - chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/http://people.eecs.berkeley.edu/~pabbeel/nips-tutorial-policy-optimization-Schulman-Abbeel.pdf
  - https://towardsdatascience.com/reinforcement-learning-101-e24b50e1d292
  - https://www.techtarget.com/searchenterpriseai/definition/reinforcement-learning
  - Web pages by Mr. Ketan Doshi on Reinforcement Learning.

Thank You