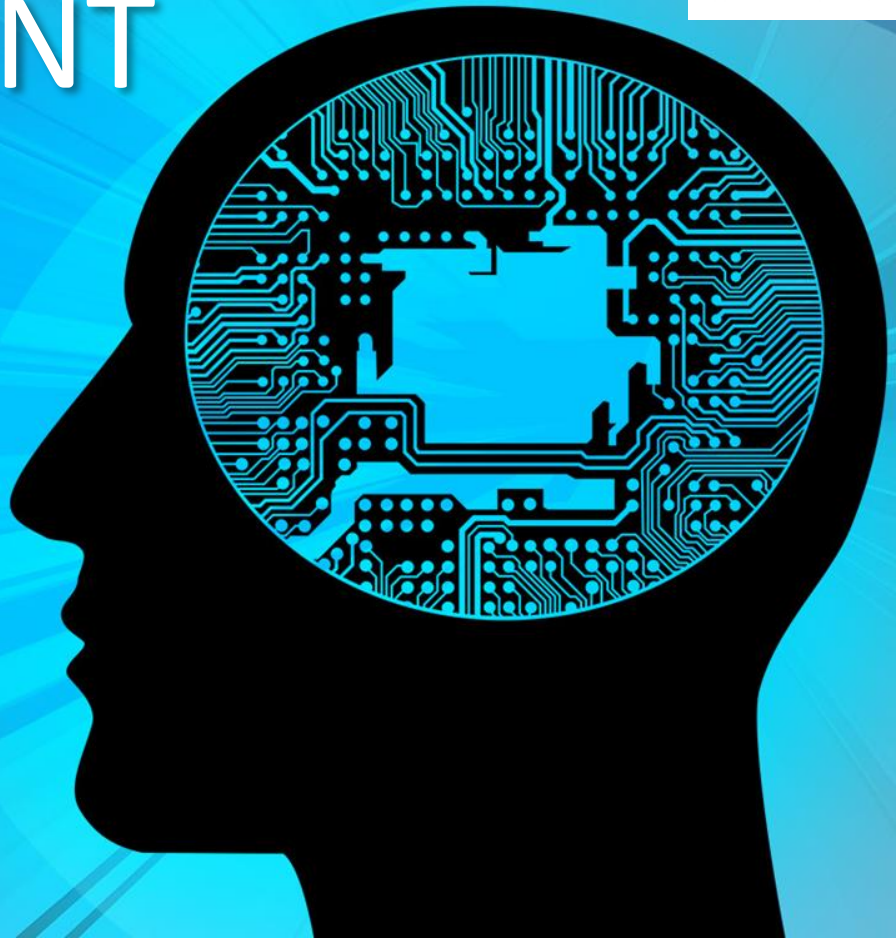
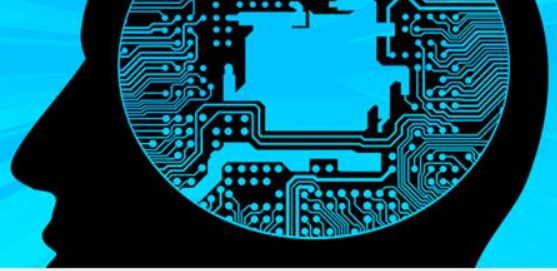
REINFORCEMENT LEARNING



Thakshila Dasun
BSc. Hons in Mechanical Engineering (Mechatronics)
CIMA, UK
Academy of Innovative Education

REINFORCEMENT LEARNING



"Reinforcement is a class of machine learning where an agent learns how to behave in the environment by performing actions and thereby drawing intuitions and seeing the results"

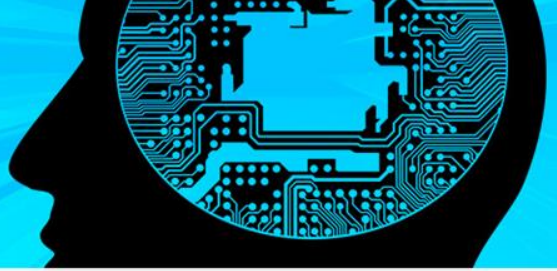
-Invented by Rich Sutton and Andrew Barto, It has taken its form in the 1980s but was archaic then. Later, Rich believed in its promising nature that it'll eventually be recognized-

- Labeled data
- Direct feedback
- Predict outcome/future

- No labels
- No feedback
- "Find hidden structure"

- Decision process
- Reward system
- Learn series of actions

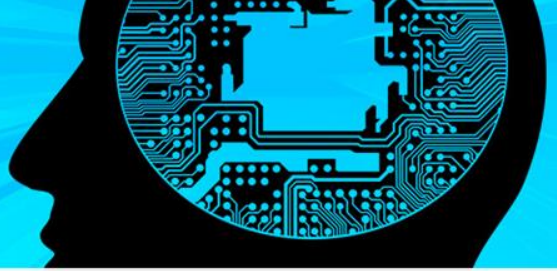
The Story



- In 2013, paper from Vlad Mnih, Atari games with Reinforcement Learning.
- It uses a technique called DQN, for deep queue network.
- This is a strategy that most human players eventually

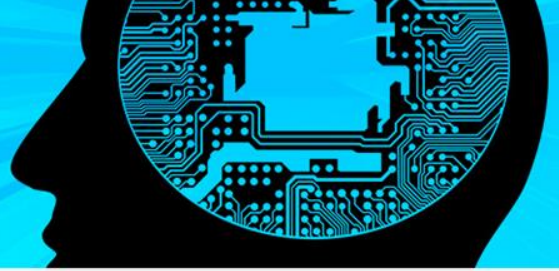
- https://www.youtube.com/watch?v=TmPfTpjtdgg

The Story



- In 2016, Deep Mind developed their original Alpha Go agent,
- which played against, and beat, the 18-time world champion,
- https://www.youtube.com/watch?v=HT-UZkiOLv8

The Story



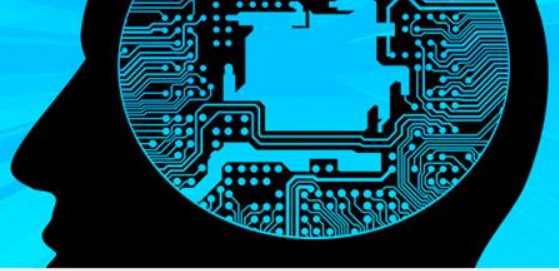
- In this 2017 paper, David Silver and others explained a new approach for training reinforcement learning agents to play adversarial games.
- They create a new agent for **Go called AlphaGo Zero, that is totally self-trained.**
- When fully trained, it beat the original AlphaGo agent 100 games to zero.

- https://youtu.be/tXlM99xPQC8

Applications

- Self Driving Cars
- Gaming
- Robotics
- Recommendation System
- Advertising and Marketing

Why Reinforcement Learning



- where has this Reinforcement Learning come from when we have a good number of Machine Learning and Deep Learning techniques available at hand?
- Reinforcement Learning supports automation by learning from the environment it is present in,
- So does Machine Learning and Deep Learning, So, why Reinforcement Learning?

Supervised Learning

- Makes machine Learn explicitly
- Data with clearly defined output is given
- Direct feedback is given
- Predicts outcome/future
- Resolves classification and regression problems

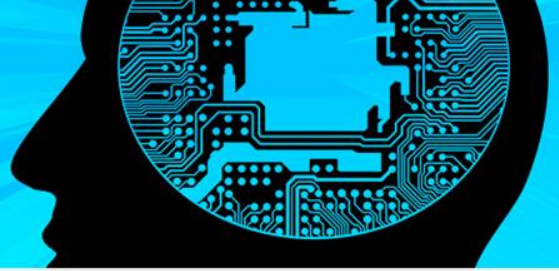
Unsupervised Learning

- Machine understands the data (Identifies patterns/structures)
- Evaluation is qualitative or indirect
- Does not predict/find anything specific

Reinforcement Learning

- An approach to AI
- Reward based learning
- Learning form +ve & +ve reinforcement
- Machine Learns how to act in a certain environment
- To maximize rewards

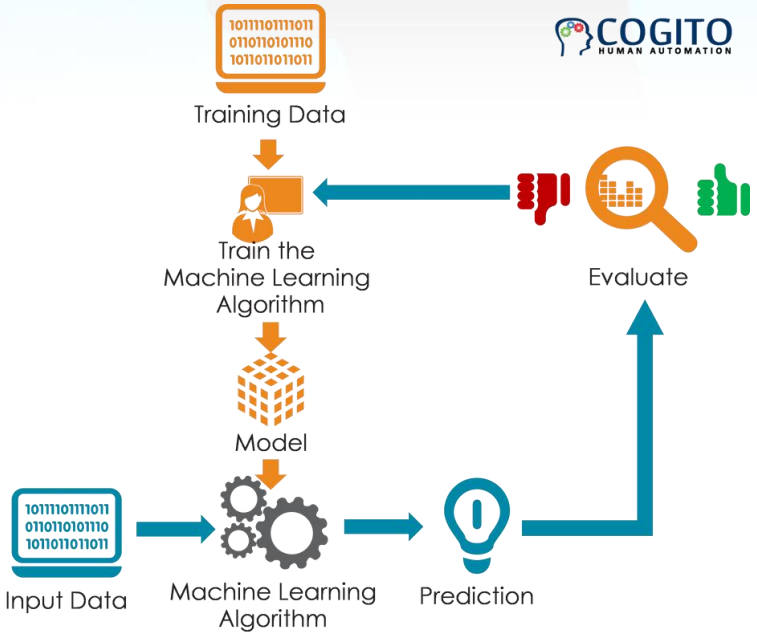
Let’s Go Deep !



- It’s very much like the natural learning process wherein, the process/the model would be receiving feedback as to whether it has performed well or not.

Let’s Go Deep !

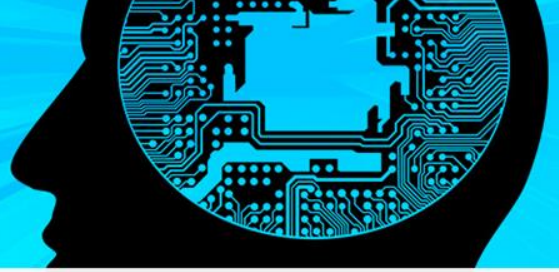
- Deep Learning and Machine Learning, are most focused on finding patterns in the existing data.
- Reinforcement Learning, does this learning by trial and error method, and eventually, gets to the right actions or the global optimum.



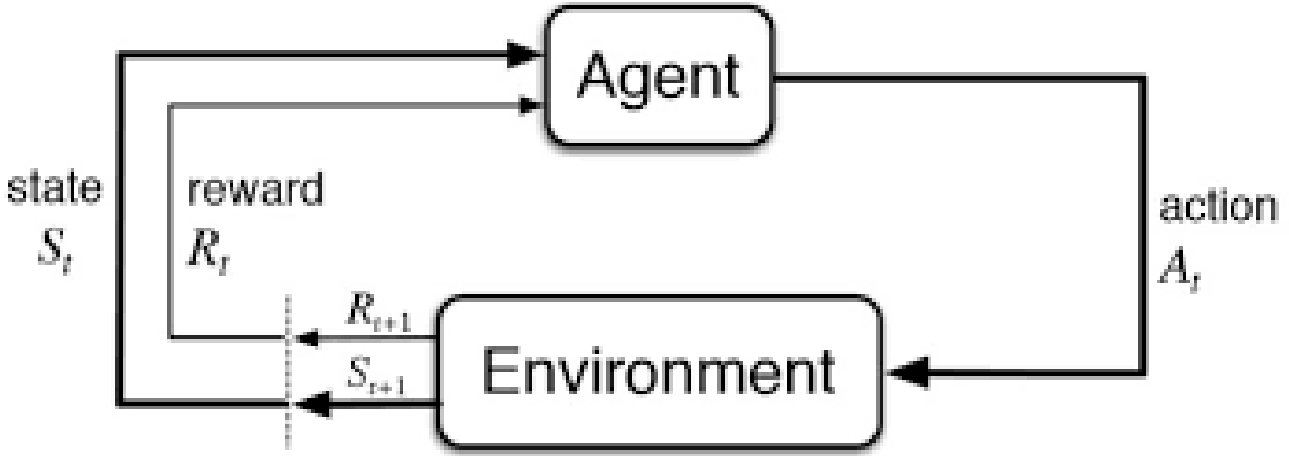
Let’s Go Deep !

- The significant additional advantage of Reinforcement Learning is that we need not provide the whole training data as in Supervised Learning. Instead, a few chunks would suffice.

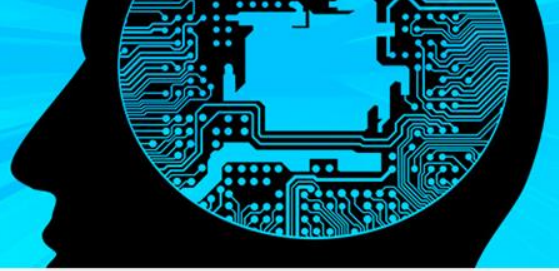
Understanding Reinforcement Learning



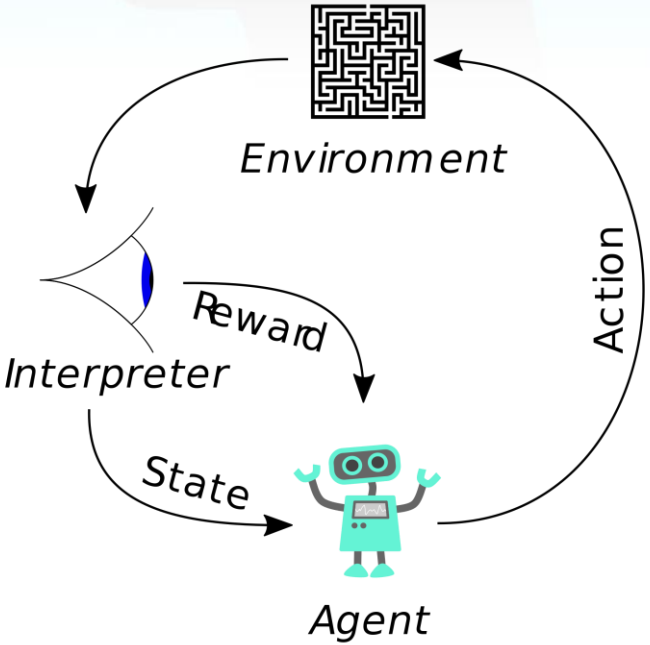
We give the machines a few inputs and actions, and then, reward them based on the output.



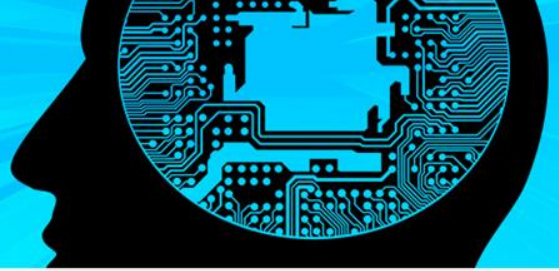
Reinforcement Learning Terminology



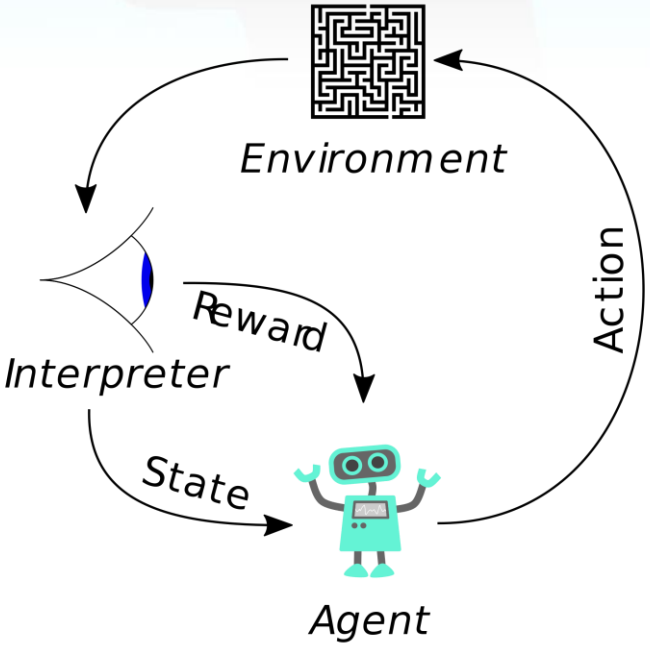
- **States:** The state is a complete description of the world, they don't hide any pieces of information that is present in the world. It can be a position, a constant or a dynamic. We mostly record these states in arrays, matrices or higher order tensors.
- **Action:** Action is usually based on the environment, different environments lead to different actions based on the agent. Set of valid actions for an agent are recorded in a space called an action space. These are usually finite in number.
- **Environment**: This is the place where the agent lives and interacts with. For different types of environments, we use different rewards, policies, etc.
- **Reward and Return:** The reward function R is the one which must be kept tracked all-time in reinforcement learning. It plays a vital role in tuning, optimizing the algorithm and stop training the algorithm. It depends on the current state of the world, the action just taken, and the next state of the world.
- **Policies:** Policy is a rule used by an agent for choosing the next action, these are also called as agents brains.



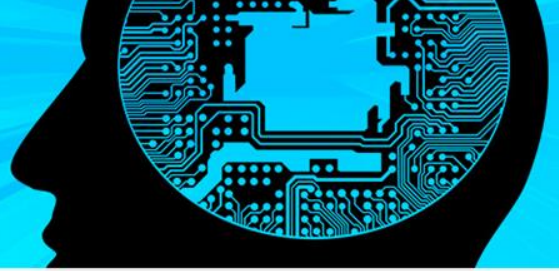
Reinforcement Learning Terminology - Simplified



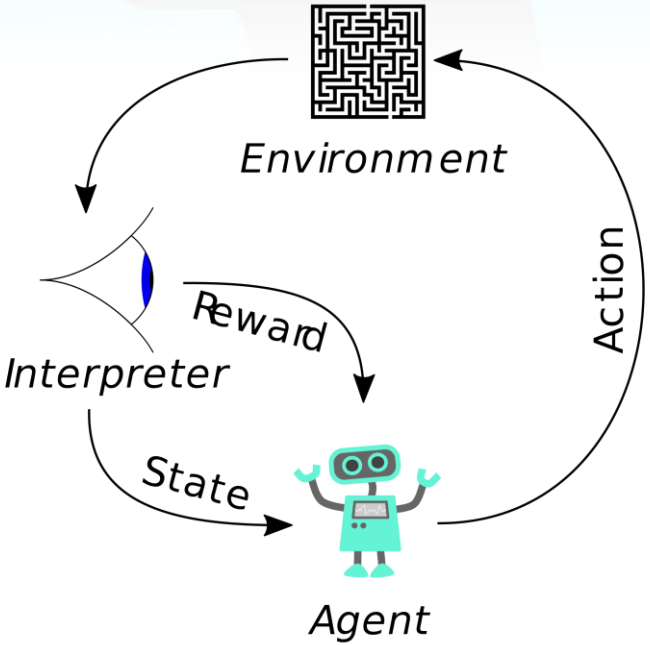
- **Agent:** The RL Algorithm that learns from trial and error
- **Environment:** The world through which the agent moves
- **Action(A):** All the possible steps that the agent can take.
- **State(S):** Current condition returned by the environment.



Reinforcement Learning Terminology - Simplified



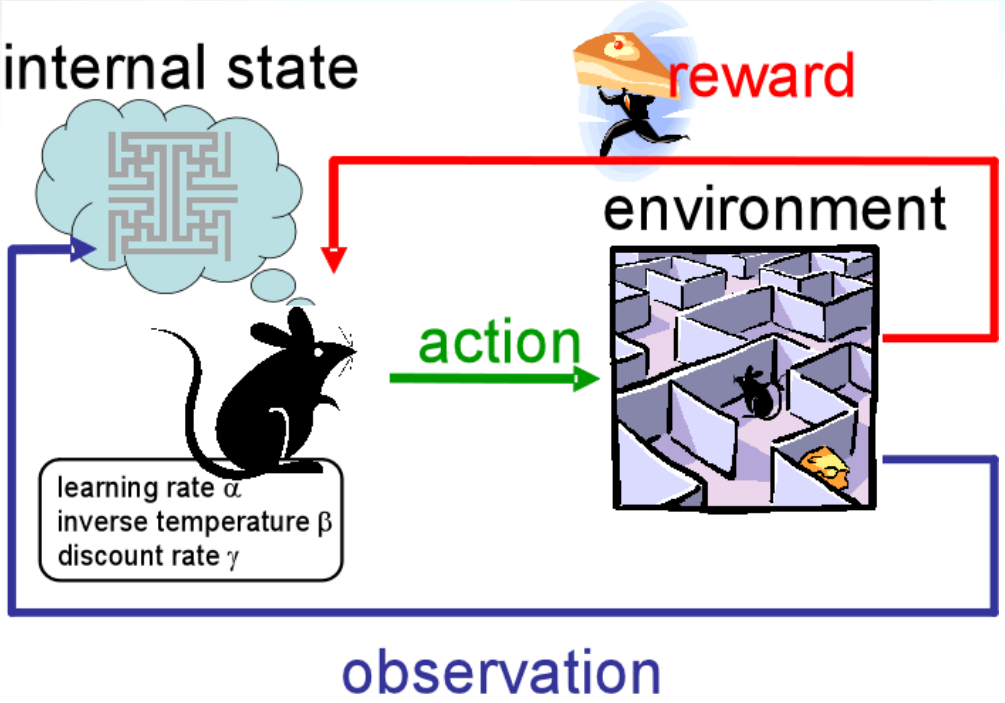
- **Reward(R):** An instant return from the environment to appraise the last action.
- **Policy:** The approach that the agent uses to determine the next action based on the current state
- **Value(V):** The expected long-term return with discount, as opposed to the short-term reward (R)



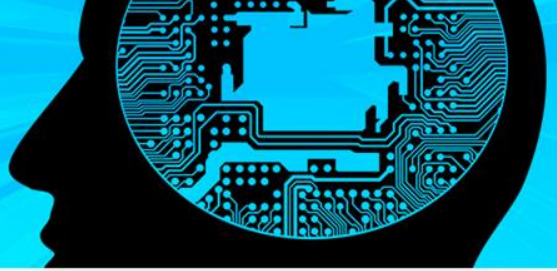
Optimization

"Reward maximization will be our end goal"

RL agent must be trained in such a way that, agent should take the best action so that the reward is maximum.



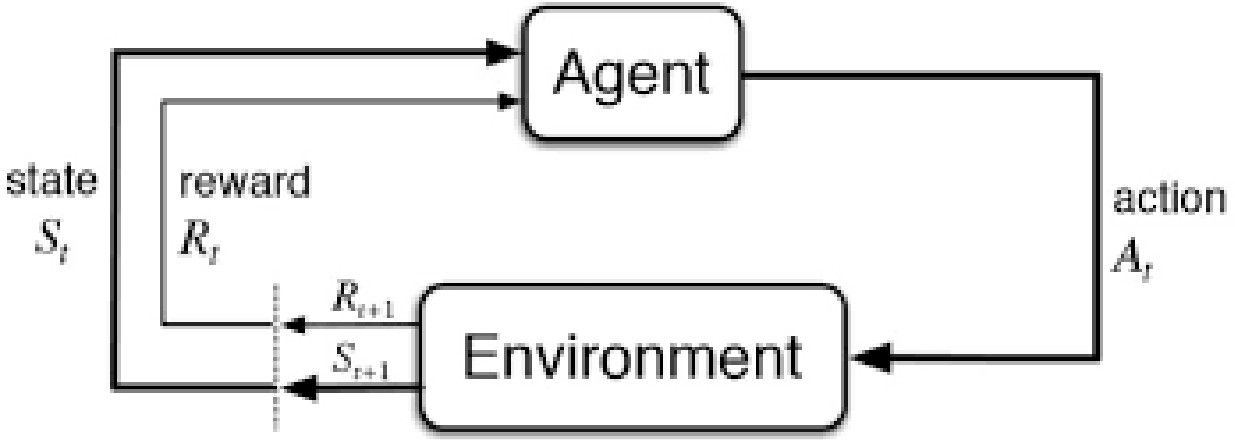
Exploration & Exploitation



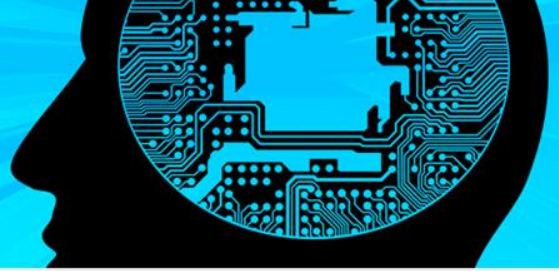
- **Exploration:** Using the already known exploited information to heighten the rewards.
- **Exploration**: Exploring and Capturing more information about an environment.

Markov Decision Process(MDP)

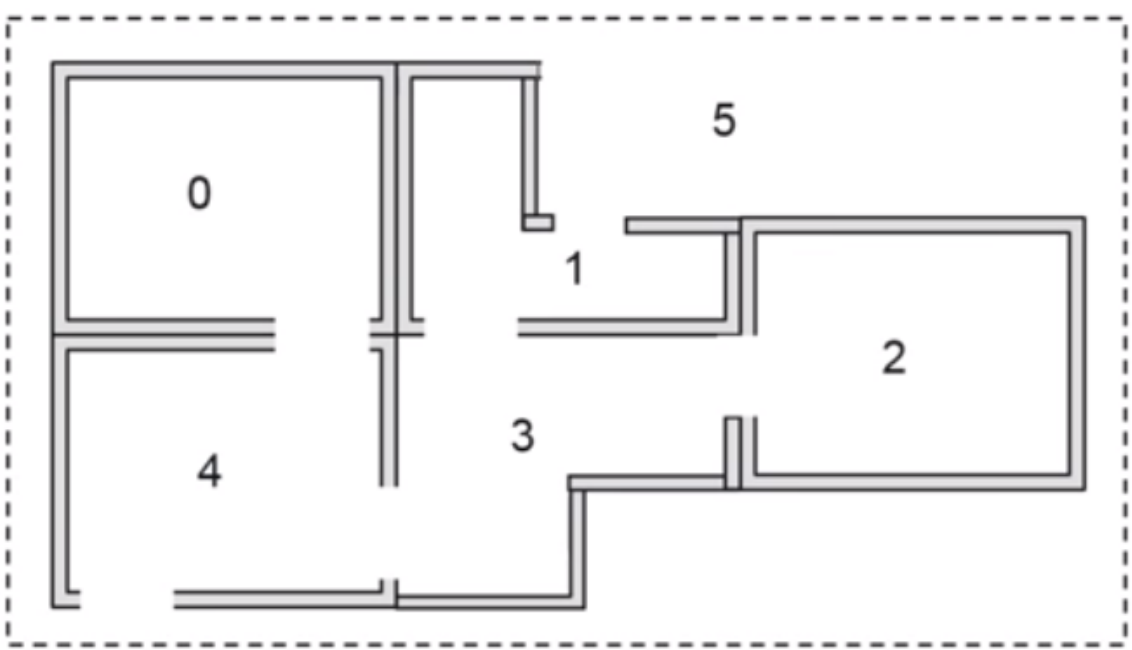
- Mathematical Approach for mapping a solution in Reinforcement Learning



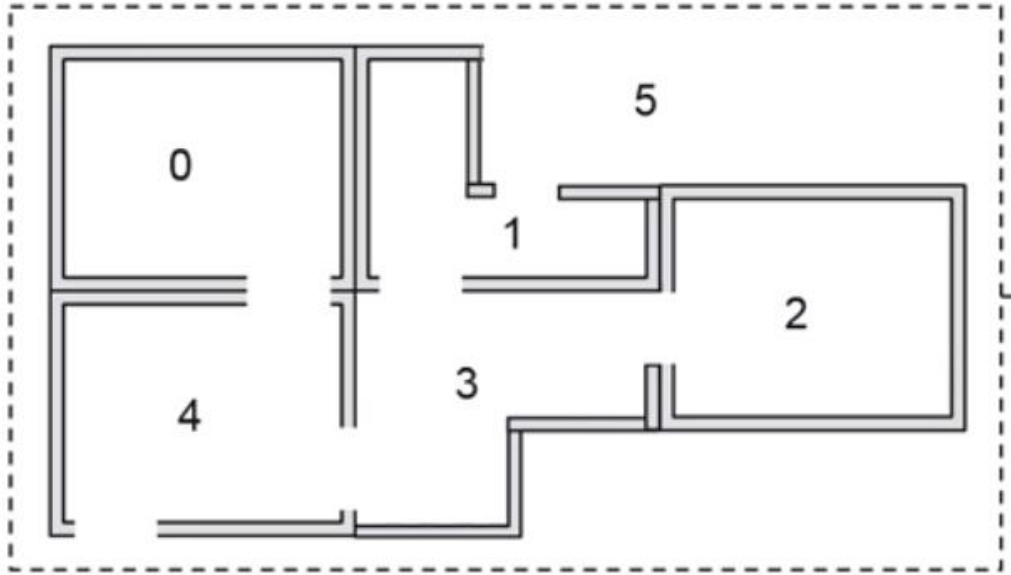
Introduction to Q-Learning

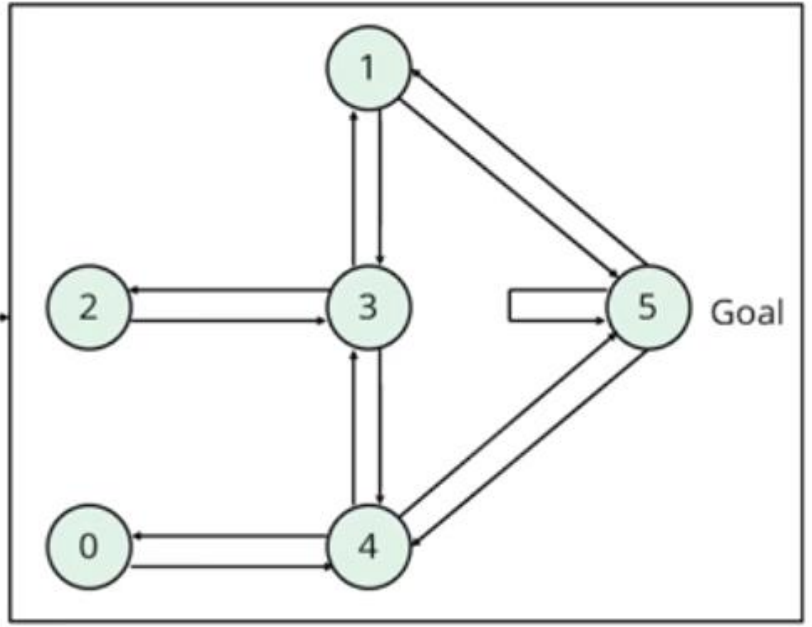


- Place an agent in any one of the rooms
- Goal is to reach outside of the building(state 5)

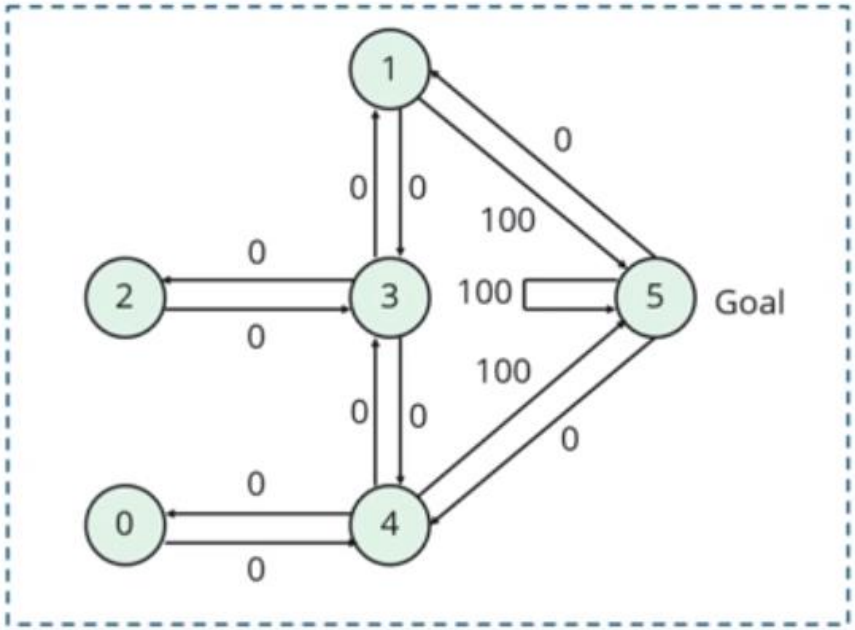


Introduction to Q-Learning





Let's initialize the Rewards



$R =$

State \ Action	0	1	2	3	4	5
0	-1	-1	-1	-1	0	-1
1	-1	-1	-1	0	-1	100
2	-1	-1	-1	0	-1	-1
3	-1	0	0	-1	0	-1
4	0	-1	-1	0	-1	100
5	-1	0	-1	-1	0	100

The -1's in the table represent null values

$$Q(state, action) \leftarrow (1-\alpha)Q(state, action) + \alpha\left(reward + \gamma \max_{a} Q(next\ state, all\ actions)\right)$$