Reinforcement Learning

NANYANG THE INNOVATIVE POLYTECHNIC
School of Information Technology

Games

- http://learnml.eu/index.php

Learning Outcomes

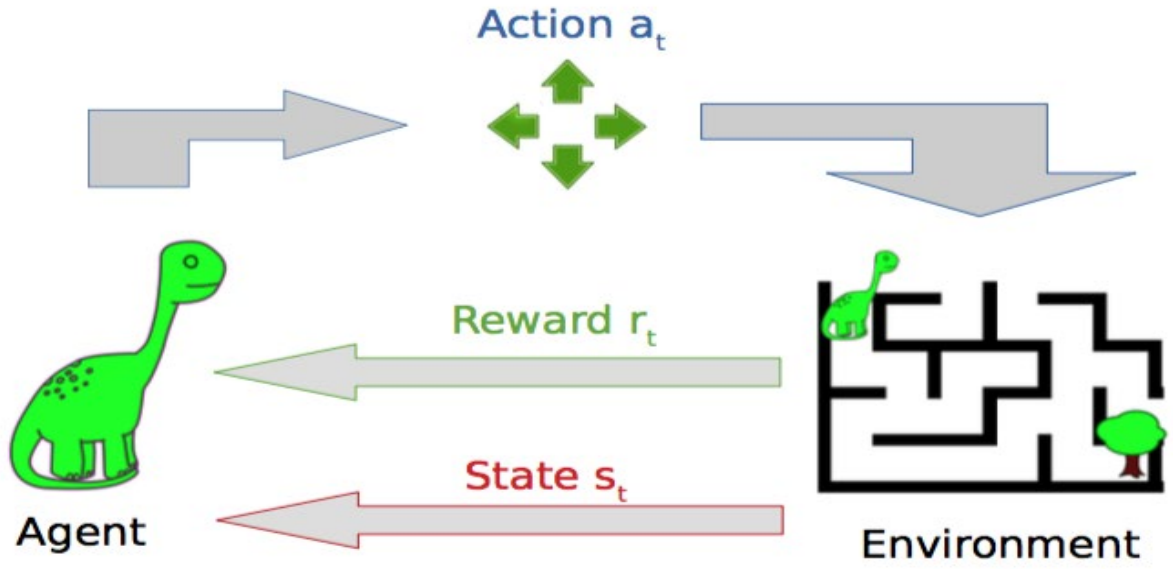
- Understand what is a reinforcement learning
- Understand how to apply reinforcement learning

Topics

- What is reinforcement learning
- How reinforcement learning work
- Reinforcement Learning Process
- Advantages and Disadvantages of reinforcement learning
- Challenge of reinforcement learning

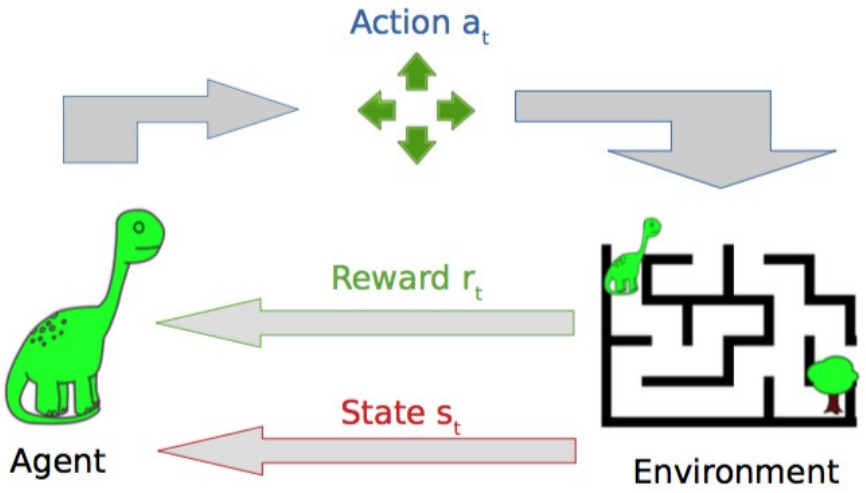
Reinforcement learning

- Reinforcement Learning is about taking suitable actions to maximize reward in a particular situation. It is employed by various software and machines to find the **best possible behaviour** or path to take in a specific situation.



Reinforcement learning

- Reinforcement learning differs from the supervised learning in a way that in supervised learning the training data has the answer key with it, so the model is trained with the correct answer itself whereas in reinforcement learning, **there is no answer** and the **reinforcement agent** decides what to do in order to perform the given task. In the absence of training data set, it is bound to **learn from its experience.**



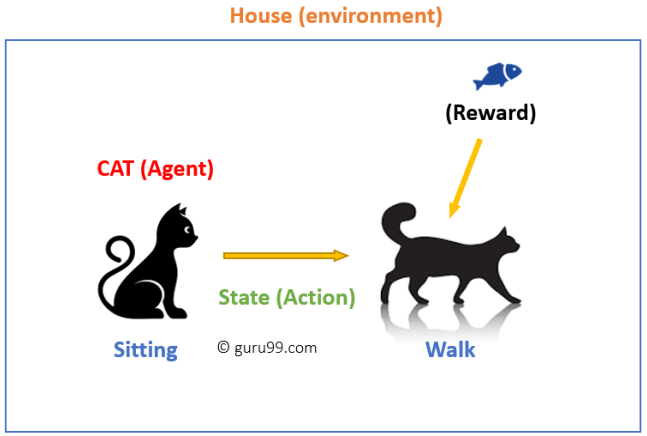
How Reinforcement Learning works?

Consider the scenario of teaching new tricks to your cat

- As cat doesn't understand English or any other human language, we can't tell her directly what to do. Instead, we follow a different strategy.
- We emulate a situation, and the cat tries to respond in many different ways. If the cat's response is the desired way, we will give her fish.
- Now whenever the cat is exposed to the same situation, the cat executes a similar action with even more eager in expectation of getting more reward(food).
- The cat learn from "what to do" from positive experiences.
- At the same time, the cat also learns what not do when faced with negative experiences.

How Reinforcement Learning works?

- Cat is an agent. House is the environment. In this case, it is your house. An example of a state could be your cat sitting, and you use a specific word to command the cat to walk.
- Our agent reacts by performing an action transition from one "state" to another "state."
- For example, your cat goes from sitting to walking.
- The reaction of an agent is an action, and the policy is a method of selecting an action given a state in expectation of better outcomes.
- After the transition, they may get a reward or penalty in return.



Characteristics of Reinforcement Learning

Here are important characteristics of reinforcement learning

- There is no supervisor, only a real number or reward signal
- Sequential decision making
- Time plays a crucial role in Reinforcement problems
- Feedback is always delayed, not instantaneous
- Agent's actions determine the subsequent data it receives

Reinforcement learning

Types of Reinforcement Learning

- Two kinds of reinforcement learning methods are:

- Positive:
- Something is added to increase the likelihood of a behavior.

- Negative:
- Something is removed to increase the likelihood of a behavior.

Reinforcement Learning Process

Learning Models of Reinforcement

There are two important concept in reinforcement learning model:

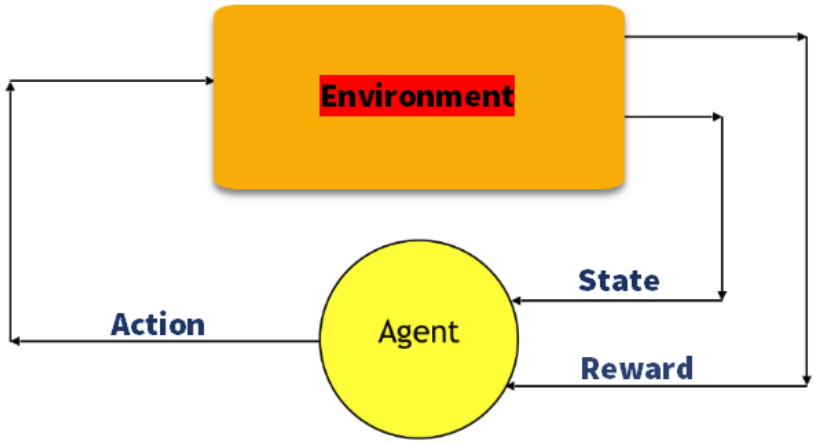
- Markov Decision Process
- Q learning

Reinforcement Learning Process

Markov Decision Process

The following parameters are used to get a solution:

- Set of actions- A
- Set of states -S
- Reward- R
- Policy- n
- Value- V



The mathematical approach for mapping a solution in reinforcement Learning is recon as a Markov Decision Process or (MDP).

Reinforcement Learning Process

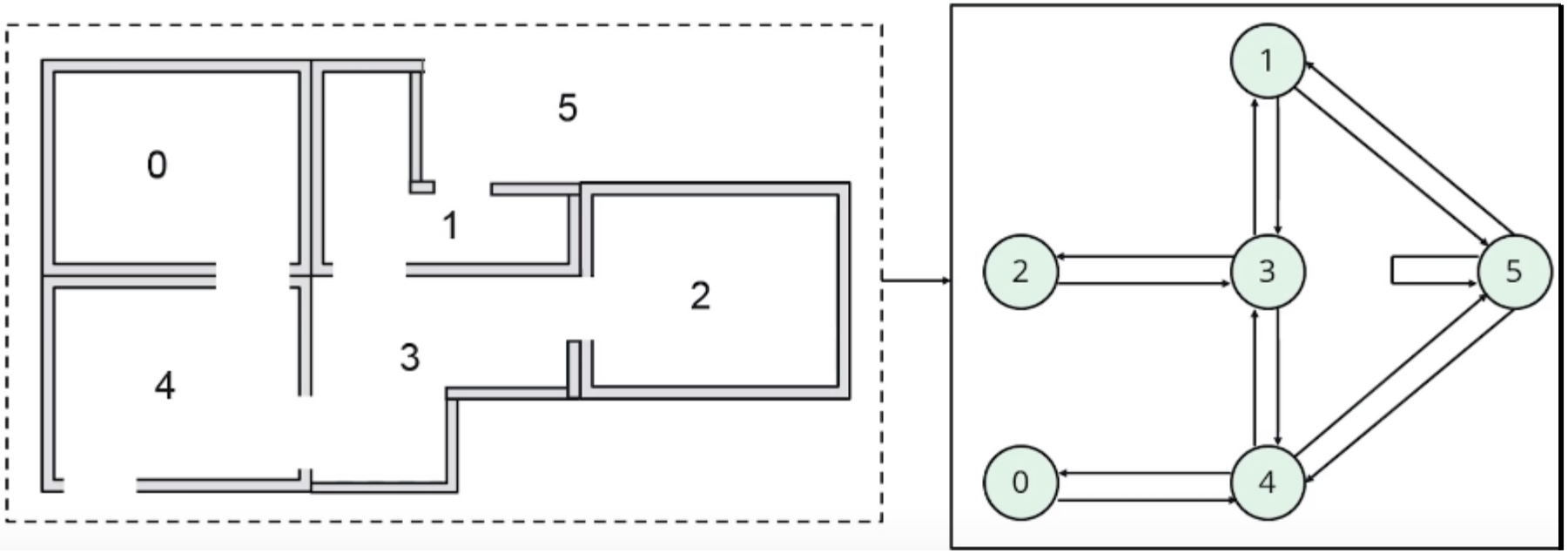
Q-Learning

- Q learning is a value-based method of supplying information to inform which action an agent should take.

Reinforcement Learning-Example

Environment

- Let's represent the rooms on a graph, each room as node and each door as link



5 is the goal

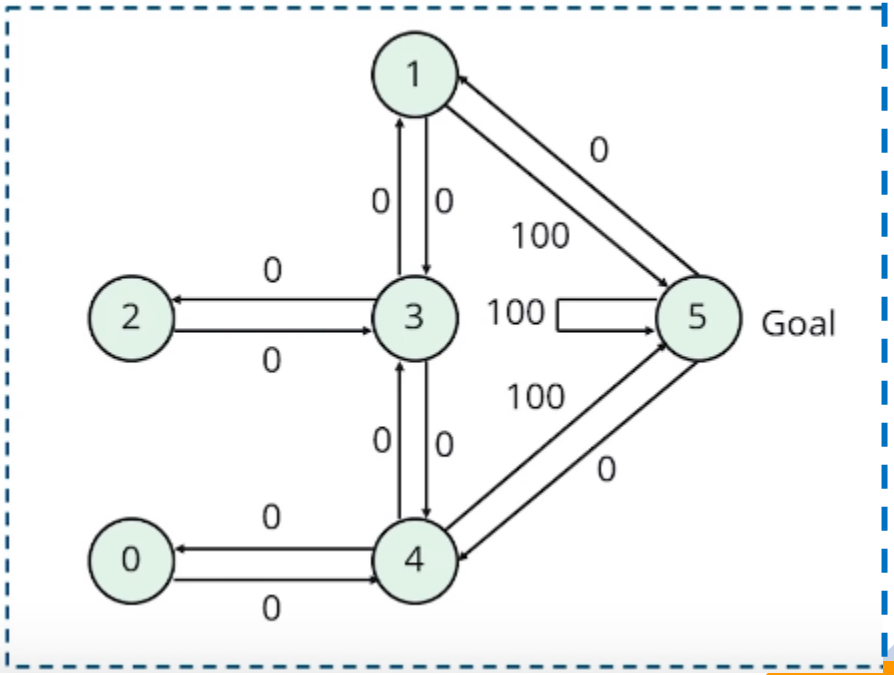
Adapted example from : http://firsttimeprogrammer.blogspot.com/

Reinforcement Learning-Example

Rewards

- Associate reward value for each door

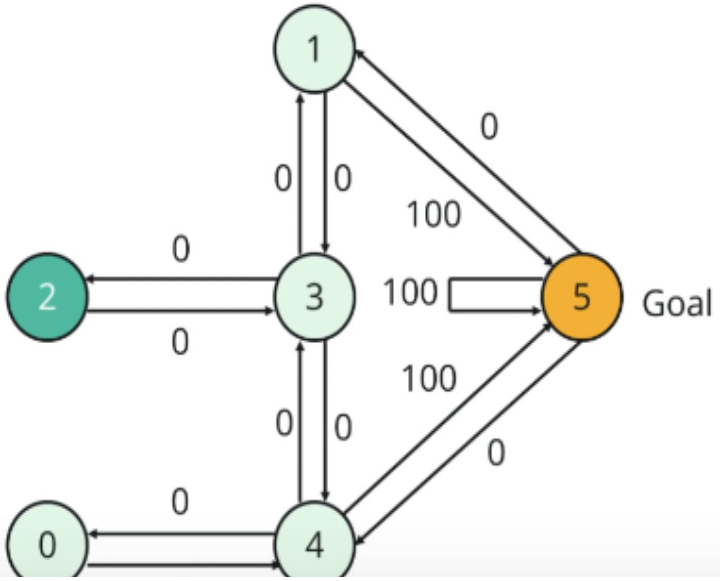
- doors that lead directly to the goal have a reward of 100
- Doors not directly connected to the target room have zero reward
- Because doors are two-way, two arrows are assigned to each room
- Each arrow contains an instant reward value



Reinforcement Learning-Example

Q learning States and Actions

- -Room(including room 5) represents a state
- -Agent's movement from one room to another represents an action
- -Here states are node number and action is the arrows direction



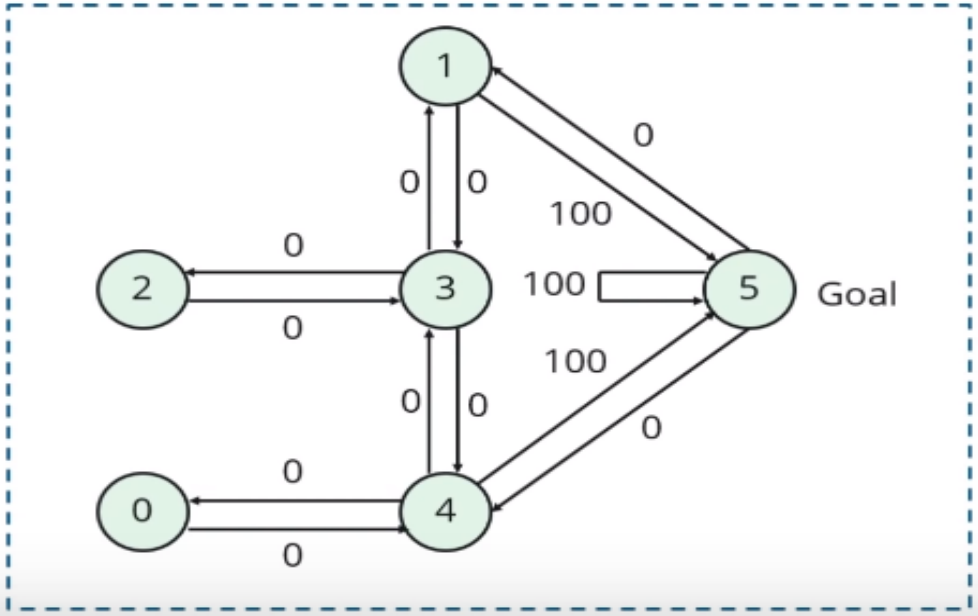
Example (Agent traverse from room 2 to room5):

1. Initial state = state 2
2. State 2 -> state 3
3. State 3 -> state (2, 1, 4)
4. State 4 -> state 5

Reinforcement Learning=Example

States and Actions

We can put the state diagram and the instant reward values into a reward table, matrix R.



$R =$

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

Reinforcement Learning-Example

- **Q-learning is an policy** reinforcement learning algorithm that seeks to find the best action to take given the current state.
- Q matrix(Q-Table) is memory to store agent learned experience.
- Q(state,action)=R(state,action)+ Gamma * Max[Q(next state, all actions]

- 0 <= Gamma parameter(discount parameter) > 1
- If Gamma is closer to 1, agent will consider future rewards with greater weight
- If Gamma is closer to zero, agent will tend to consider only immediate rewards

Action lead to next state

Current state	0	1	2	3	4	5
0						
1						
2			Q-values			
3						
4						
5						

Reinforcement Learning-Example

Q-Table and Q value

Set learning parameter Gamma=0.8, initial state Room 1, initialize Q-Table to zero

From room 1 you can either goto room 3 or 5, let's select room 5

From room 5, cal max Q value for next state based on all possible actions

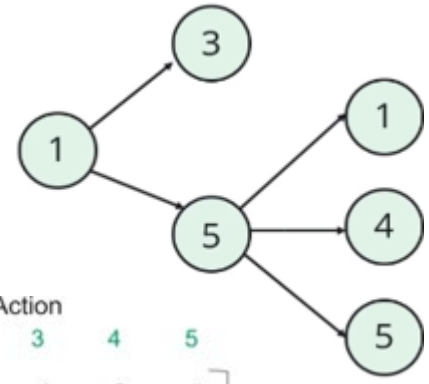
Q(state,action)=R(state,action)+ Gamma * Max[Q(next state, all actions]

Q(1,5)=R(1,5)+0.8*Max[Q(5,1),Q(5,4),Q(5,5)]=100+0.8*0=100

Update Q Table (1,5) with 100

Q=

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0



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

Reinforcement Learning-Example

Taking Action: Exploit

- Use the q-table as a reference and view all possible actions for a given state. The agent then selects the action based on the max value of those actions. This is known as ***exploiting*** since we use the information we have available to us to make a decision.

Reinforcement Learning-Example

Taking Action: Explore

- The way to take action is to act randomly. This is called ***exploring***. Instead of selecting actions based on the max future reward we select an action at random. Acting randomly is important because it allows the agent to explore and discover new states that otherwise may not be selected during the exploitation process.

- We can balance exploration/exploitation using epsilon (ε) and setting the value of how often you want to explore vs exploit.

Reinforcement Learning-Example

Taking Action: Explore or Exploit

For the next episode, we start with a randomly chosen initial state, i.e. state 3

- From room 3 you can either go to room 1,2 or 4, let's select room 1.
- From room 1, calculate maximum Q value for this next state based on all possible actions:

*Q(state, action) = R(state, action) + Gamma * Max[Q(next state, all actions)]*

Q(3,1) = R(3,1) + 0.8 * Max[Q(1,3), Q(1,5)]= 0 + 0.8 * [0, 100] = 80

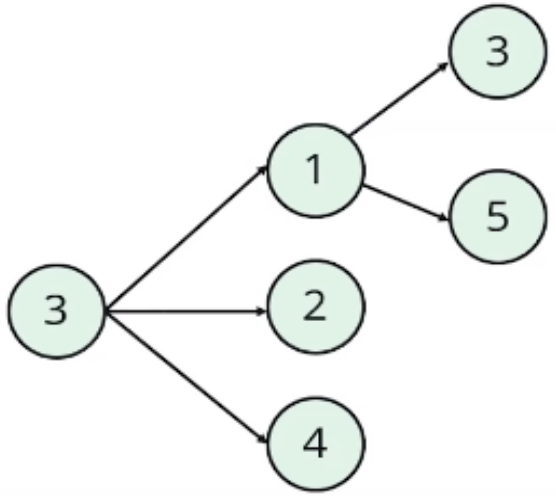
The matrix Q get's updated

$Q =$

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	0	0	0	100
2	0	0	0	0	0	0
3	0	80	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0

$R =$

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



Reinforcement Learning-Example

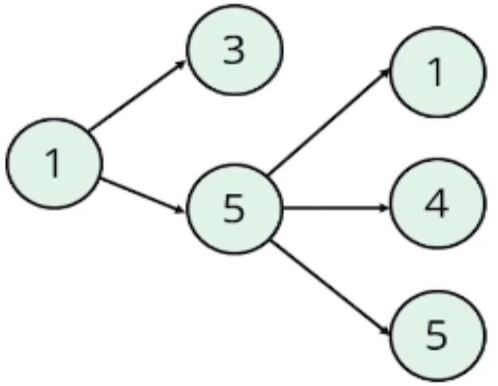
Taking Action: Explore or Exploit

For the next episode, the next state, 1, now becomes the current state. We repeat the inner loop of the Q learning algorithm because state 1 is not the goal state.

- From room 1 you can either go to room 3 or 5, let's select room 5.
- From room 5, calculate maximum Q value for this next state based on all possible actions:

*Q(state, action) = R(state, action) + Gamma * Max[Q(next state, all actions)]*

Q(1,5) = R(1,5) + 0.8 * Max[Q(5,1), Q(5,4), Q(5,5)] = 100 + 0.8 * 0 = 100
The matrix Q remains the same since, Q(1,5) is already fed to the agent



$Q =$

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	0	0	0	100
2	0	0	0	0	0	0
3	0	80	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0

$R =$

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

Reinforcement Learning-Example

basic update rule for q-learning:

Q[state, action] = Q[state, action] + lr * (reward + gamma * np.max(Q[new_state, :]) — Q[state, action])

Learning Rate: lr or learning rate, often referred to as *alpha* or α, can simply be defined as how much you accept the new value vs the old value.
Gamma: gamma or *γ* is a discount factor. It's used to balance immediate and future reward..
Reward: reward is the value received after completing a certain action at a given state.

Reinforcement Learning vs. Supervised Learning

Parameters	Reinforcement Learning	Supervised Learning
Decision style	reinforcement learning helps you to take your decisions sequentially.	In this method, a decision is made on the input given at the beginning.
Works on	Works on interacting with the environment.	Works on examples or given sample data.
Dependency on decision	In RL method learning decision is dependent. Therefore, you should give labels to all the dependent decisions.	Supervised learning the decisions which are independent of each other, so labels are given for every decision.
Best suited	Supports and work better in AI, where human interaction is prevalent.	It is mostly operated with an interactive software system or applications.
Example	Chess game	Object recognition

Reinforcement Learning

Why use Reinforcement Learning?

Here are prime reasons for using Reinforcement Learning:

- It helps you to find which situation needs an action
- Helps you to discover which action yields the highest reward over the longer period.
- Reinforcement Learning also provides the learning agent with a reward function.
- It also allows it to figure out the best method for obtaining large rewards.

Reinforcement Learning

When Not to Use Reinforcement Learning?

You can't apply reinforcement learning model is all the situation. Here are some conditions when you should not use reinforcement learning model.

- When you have enough data to solve the problem with a supervised learning method
- You need to remember that Reinforcement Learning is computing-heavy and time-consuming. in particular when the action space is large.

Reinforcement Learning

Applications of Reinforcement Learning

Here are applications of Reinforcement Learning:

- Robotics for industrial automation.
- Business strategy planning
- Machine learning and data processing
- Aircraft control and robot motion control

Reinforcement Learning

Challenges of Reinforcement Learning

Here are the major challenges you will face while doing Reinforcement earning:

- Reward design which should be very involved
- Parameters may affect the speed of learning.
- Realistic environments can have partial observability.
- Too much Reinforcement may lead to an overload of states which can diminish the results.
- Realistic environments can be non-stationary.

Summary

- What is reinforcement learning
- How reinforcement learning work
- Reinforcement Learning Process
- Advantages and Disadvantages of reinforcement learning
- Challenge of reinforcement learning