LECTURER: Nghia Duong-Trung

ARTIFICIAL INTELLIGENCE

TOPIC OUTLINE

History of AI

Modern AI Systems

Reinforcement Learning

Natural Language Processing – Part 1

Natural Language Processing – Part 2

Computer Vision

UNIT 3

REINFORCEMENT LEARNING

STUDY GOALS

- Understand the basic principles of reinforcement learning.
- Utilize Markov decision processes.
- Apply the Q-learning algorithm.

EXPLAIN SIMPLY

1. How does reinforcement learning work?

2. What is temporal difference learning?

3. How does the Q-learning algorithm work?

THE PROCESS OF REINFORCEMENT LEARNING

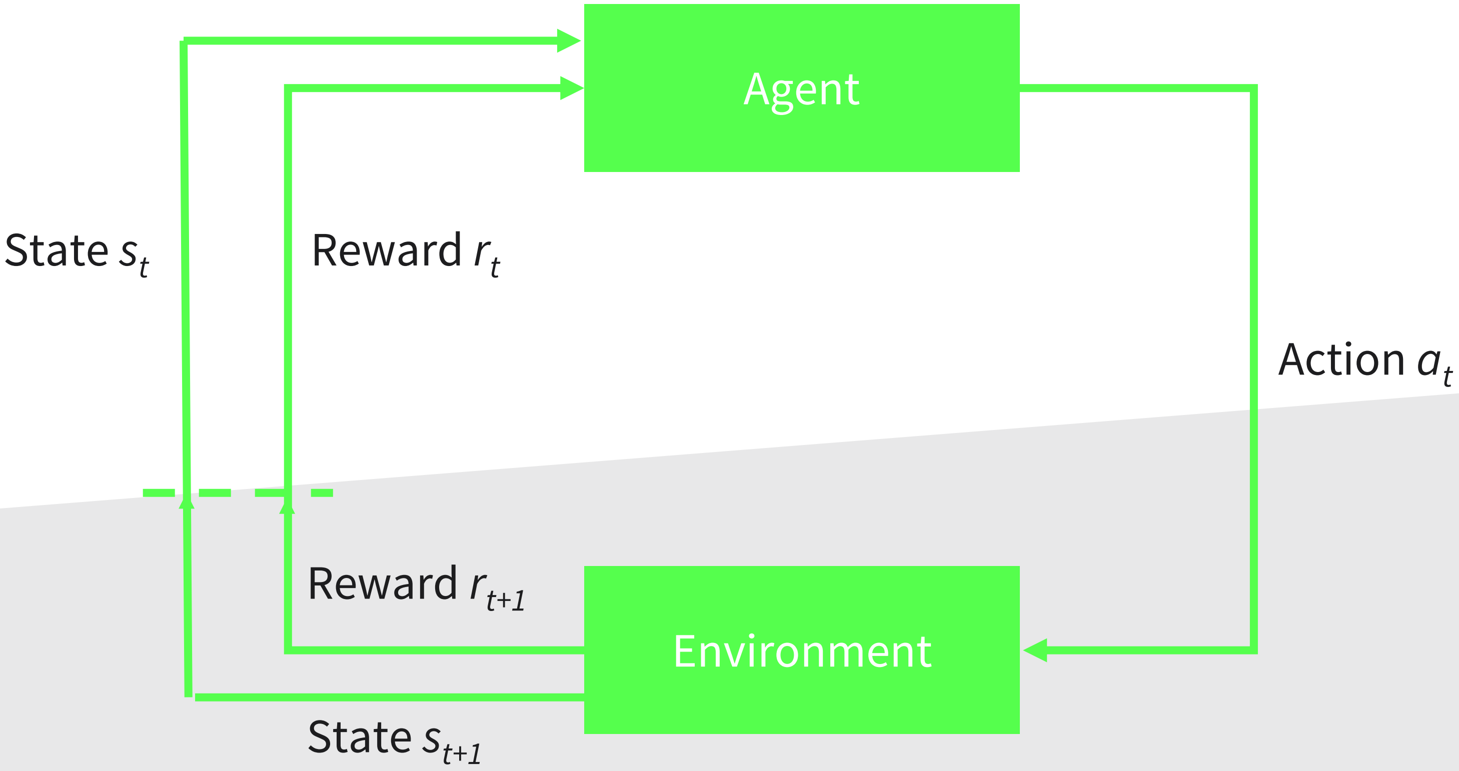
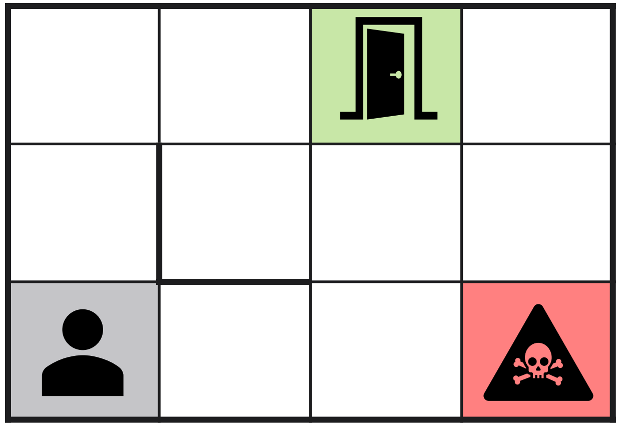
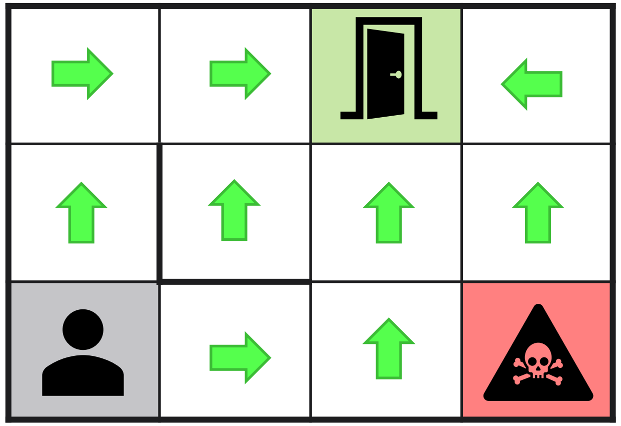


Image Source: Custom Depiction

REINFORCEMENT LEARNING EXAMPLE





- State: the location of the person at a specific time
- Action: go left, go right, go up, go down
- Reward could be simply be defined as 1 of the person manages to get out of the door and zero if, for example, go to a toxic room

- The idea is the person “remembers” which way he/she should go to get the reward
- -> maximize the rewards over time -> maximize **the (expected) return**.
- The person/agent is called a policy and a policy is nothing else than a function taking as input the state and returning as output the action.

Image Source: Custom Depiction

COMPONENTS OF THE MARKOV DECISION PROCESS

- States: S
- Actions: A
- Rewards for an action a at a state s: $r_a = R(s, a, s')$
- Transition probabilities for the actions to move from one state to the next state: $T_a(s, s')$
- Transition function: $T_{a_t}(s, s') = P(s_{t+1} | s_t, a_t)$
- Policy: $\pi(s, a) = p(a_t = a | s_t = s)$

TEMPORAL DIFFERENCE LEARNING

- Model-free approach → learning from experience

$$\underbrace{Q(s,a)}_{\text{Original state (t-1)}} = \underbrace{r}_{\text{Reward}} + \gamma max_{a'} \underbrace{Q(s',a')}_{\text{New state (t)}}$$

$$TD(s,a) = r + \gamma max_{a'} Q(s',a') - Q(s,a)$$

⤷

$$Q_t(s,a) = Q_{t-1}(s,a) + \underbrace{\alpha}_{\text{Learning rate}} TD_{\mathrm{t}}\,(s,a)$$

Based on temporal difference learning

Q-Learning Algorithm:

1. Choose an action for the current state. Possible strategies:
   - **Exploration:** perform random actions in order to find out more about the environment
   - **Exploitation:** perform actions based on known information
2. Perform chosen action
3. Evaluate the outcome and get the value of the reward, update Q-table

REVIEW STUDY GOALS

- Understand the basic principles of reinforcement learning.
- Utilize Markov decision processes.
- Apply the Q-learning algorithm.

THE BELLMAN EQUATION

Concepts:

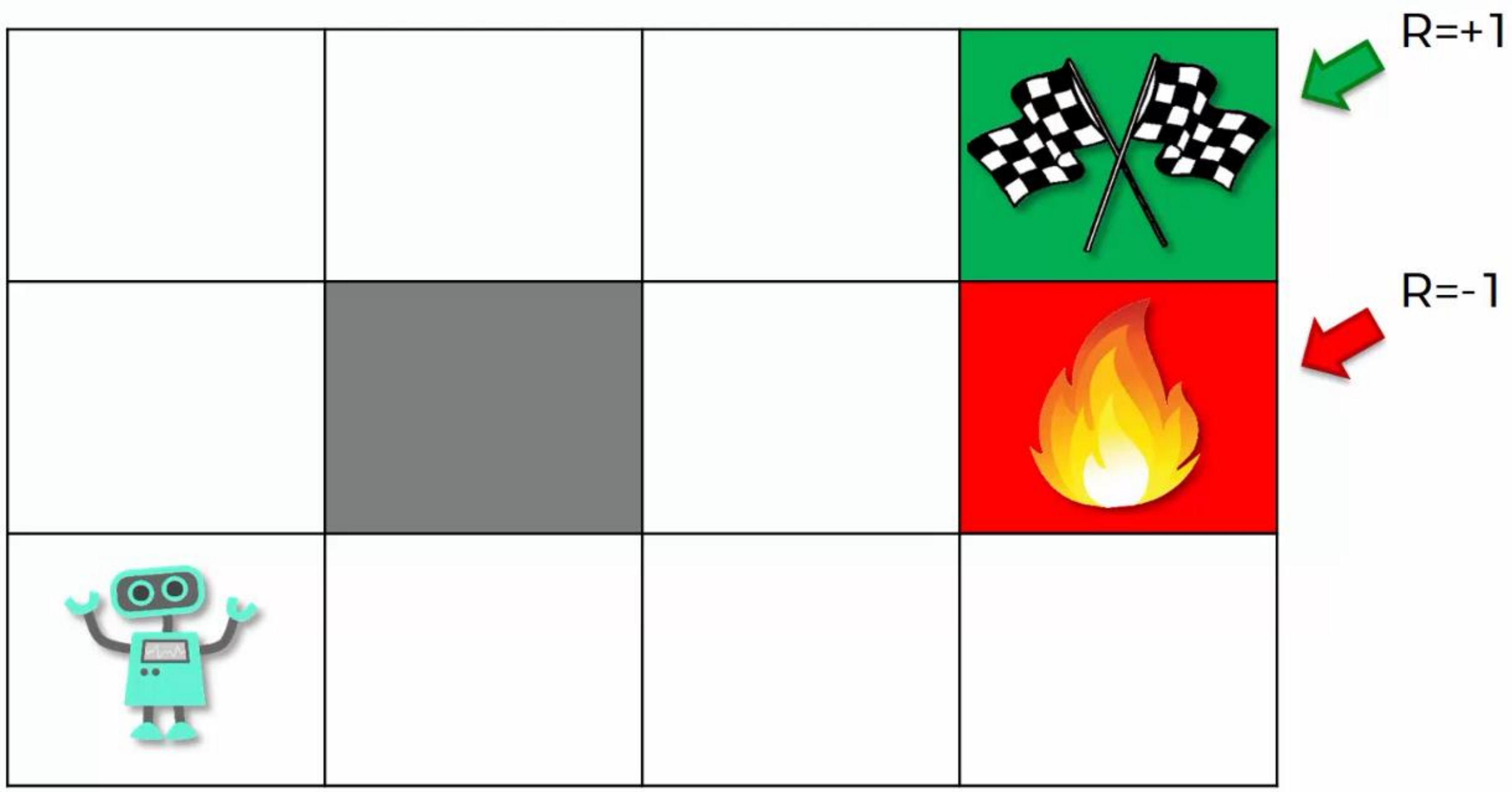
- s – State
- a – Action
- R – Reward
- γ - Discount

Richard Ernest Bellman[3] (August 26, 1920 – March 19, 1984) was an American applied mathematician, who introduced dynamic programming in 1953, and made important contributions in other fields of mathematics.

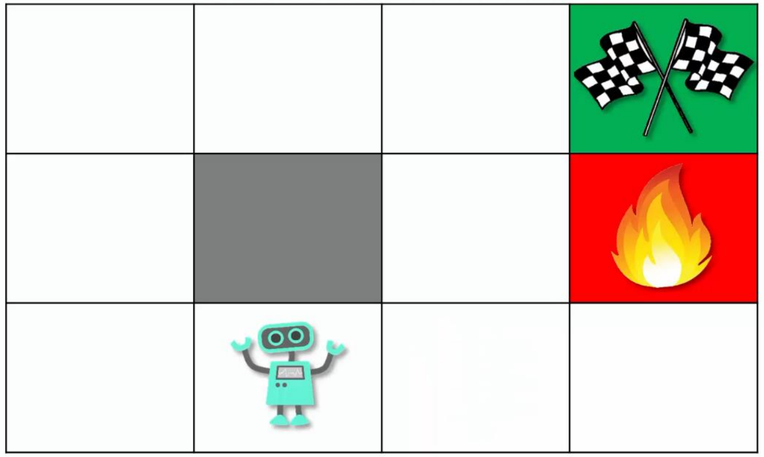
THE BELLMAN EQUATION

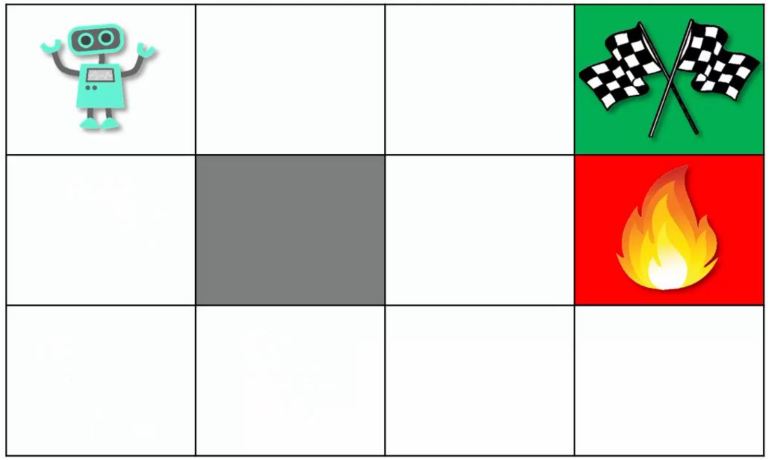
- A classical maze where you have some blocks.
  - White blocks: an agent can step into
  - Gray blocks are the one that are just not accessible
  - The green is where the agent is supposed to end up in. The agent wins the game.
  - The red is the firepit. If the agent goes into the firepit, it loses the game.

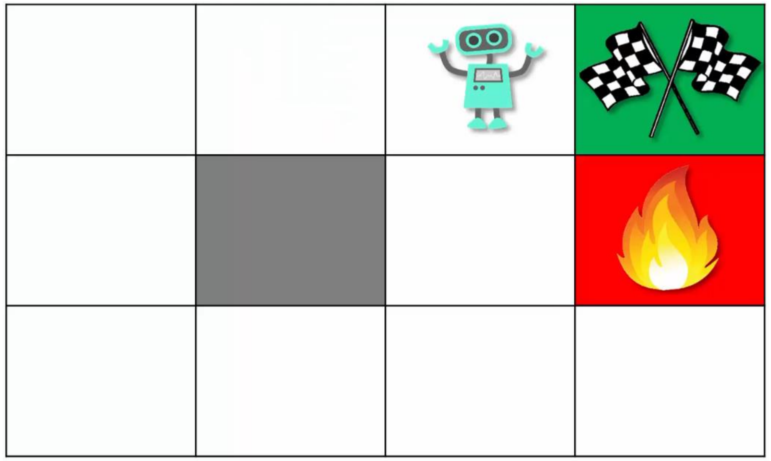
  - An agent has 4 possible actions: go up, down, left, right.

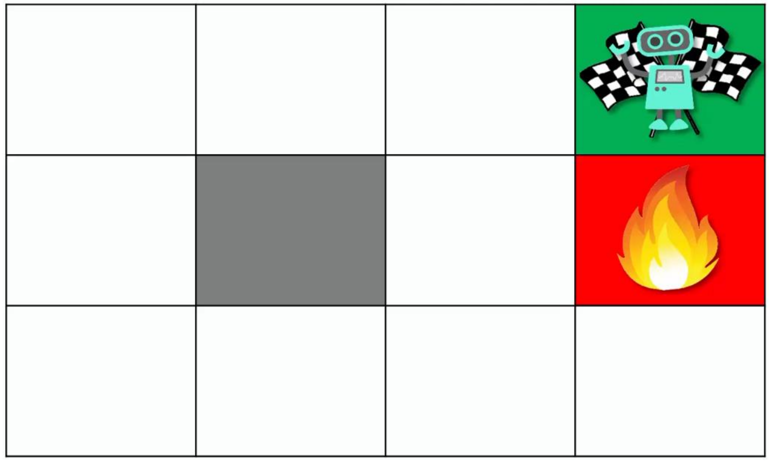


THE BELLMAN EQUATION



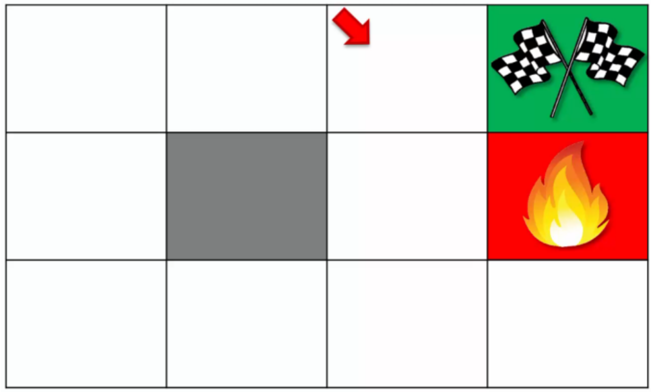




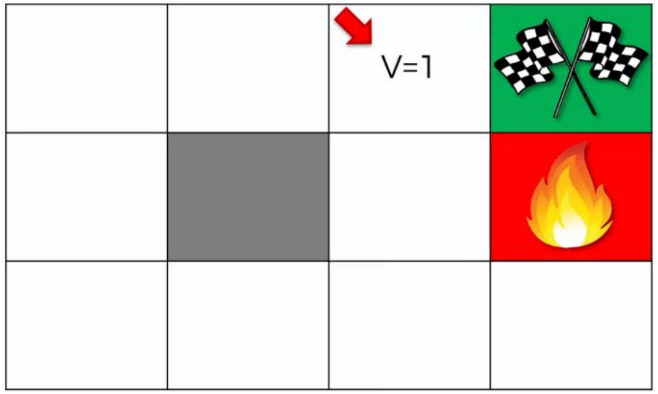


The agent gets in the Green square and gets a Reward.
It asks: how can I get to this Green square? What are the preceding states I was in and what actions should I take to get here?

THE BELLMAN EQUATION

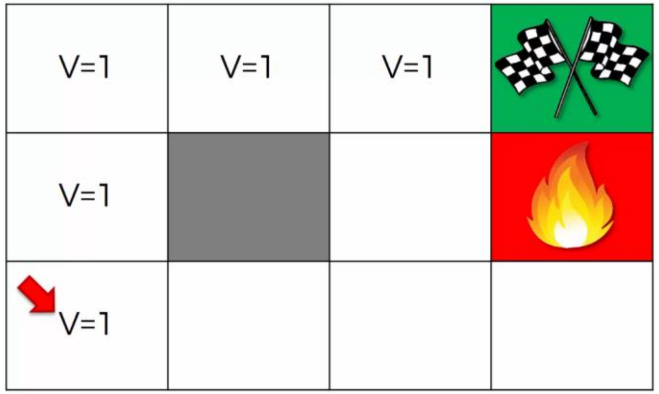


As soon as I’m in the White square, I know I’ll just take one more action, I’ll be in the Green square. So I’ll get a reward of 1.



I’ll mark the White square with 1 because it leads to the reward of 1. So the perceived value of being in this state.

Next, how can I get into this square?

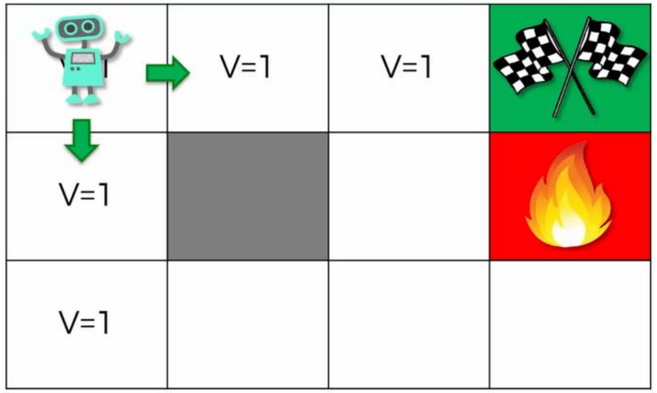


We could possibly think about designing an equation that helps an agent go through the maze.

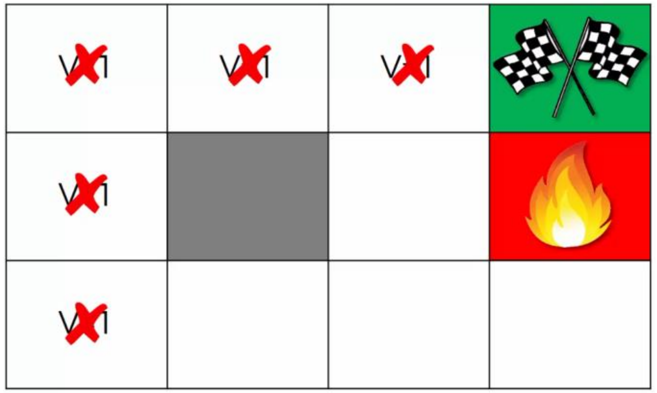
-> Look at the reward, then the preceding states give it a value of equal to reward

-> create a pathway

THE BELLMAN EQUATION



If the agent starts at the corner, which way should it go? If the values are all equal to 1.



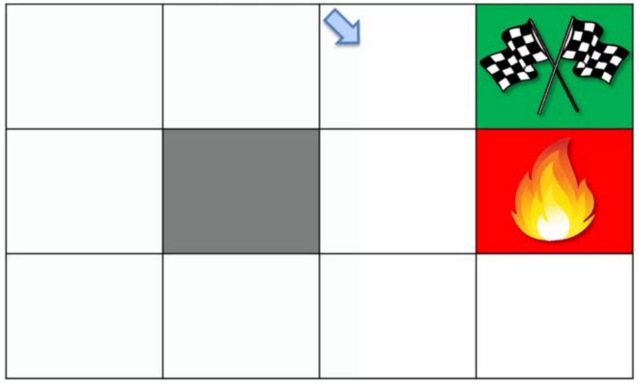
So, the Bellman equation

$$V(s) = \max_{a}(R(s,a) + \gamma V(s'))$$

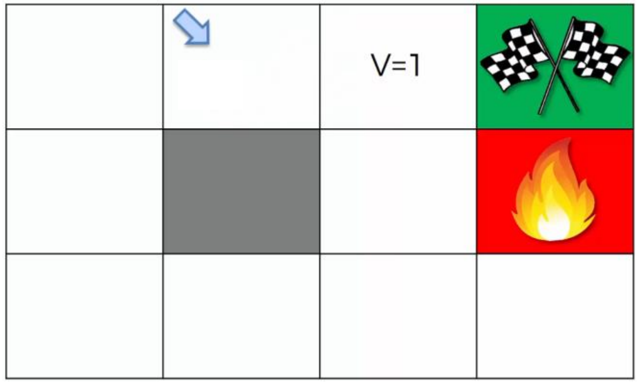
V(s): the value in a certain state.
s' : s prime is the state, the following state, the state that you will end up in after this state by taking a certain action *a*.
There are many actions that the agent can take, and that's why we take the *max*.

THE BELLMAN EQUATION

$$V(s) = \max_{a}(R(s,a) + \gamma V(s'))$$

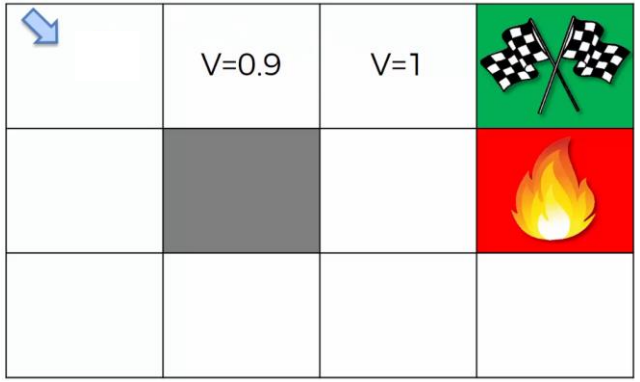


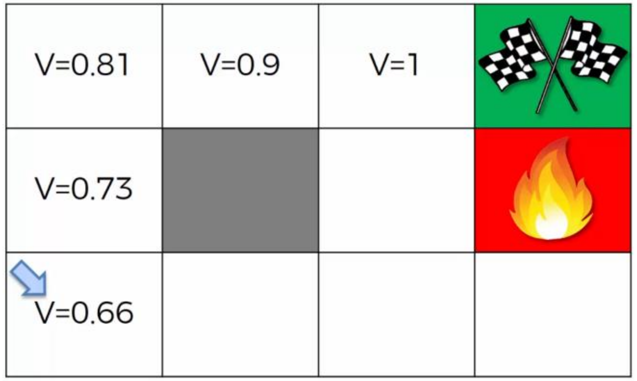
Max(the reward + the value of the next state)
The maximum will be moving to the right. The reward of moving the right is 1.
We don’t have the value in the Green because we are in the best state possible. The final state. It won’t have a value.



γ=0.9
The maximum reward will be moving to the right.
V(s) = 0 + 0.9*1 = 0.9

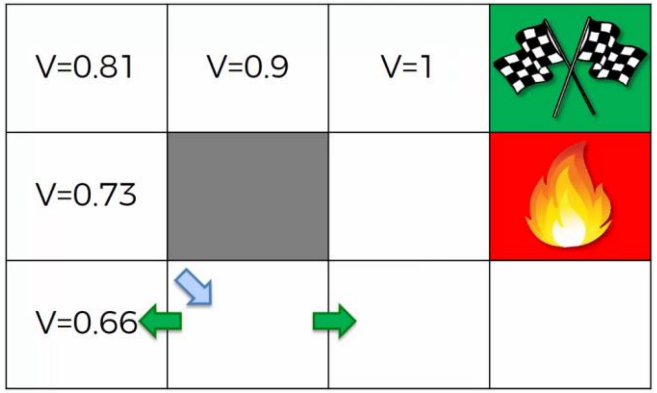
The discount factor discounts the value of the state as you are further away.





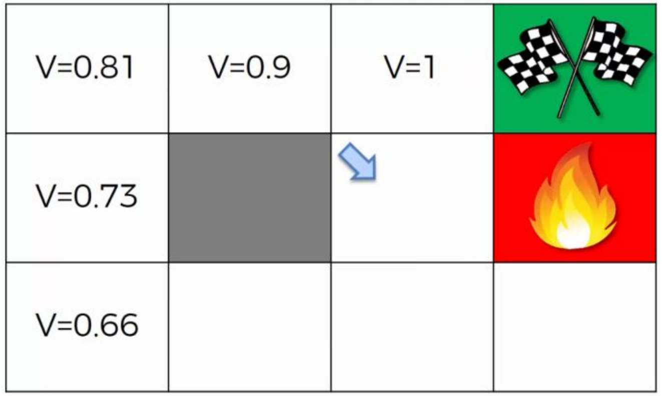
THE BELLMAN EQUATION

$$V(s) = \max_{a}(R(s,a) + \gamma V(s'))$$



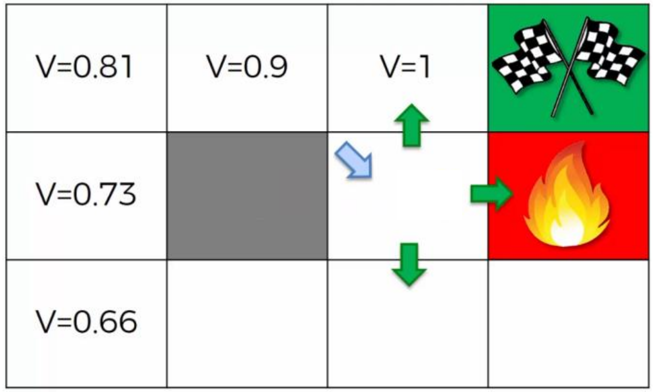
V=0.81	V=0.9	V=1	🏁
V=0.73	▨		🔥
V=0.66 ←		→	

How do we calculate the value in this square? You might not actually go left, right?



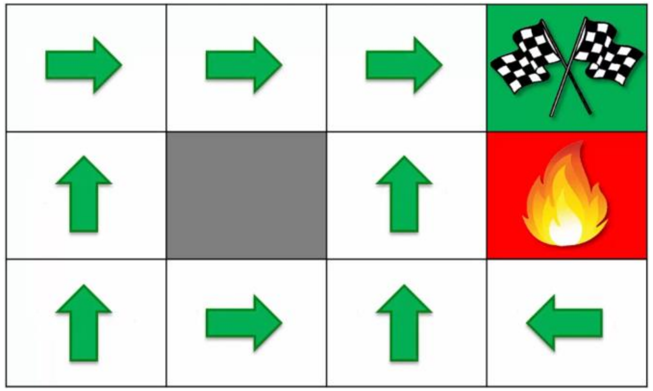
V=0.81	V=0.9	V=1	🏁
V=0.73	▨		🔥
V=0.66			

Calculate this square first. Obviously from here, the best way is to go up.

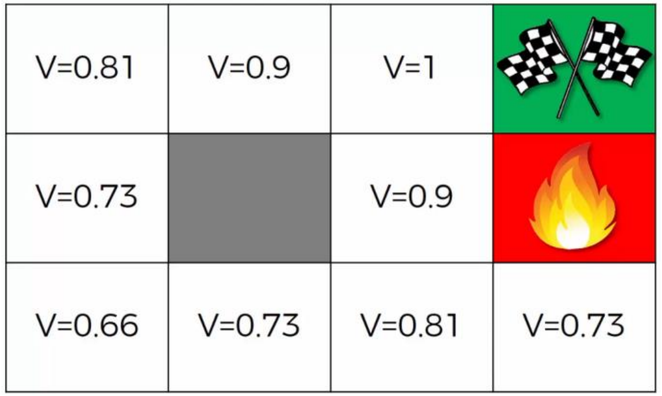


V=0.81	V=0.9	V=1	🏁
V=0.73	▨	↑ → ↓	🔥
V=0.66			

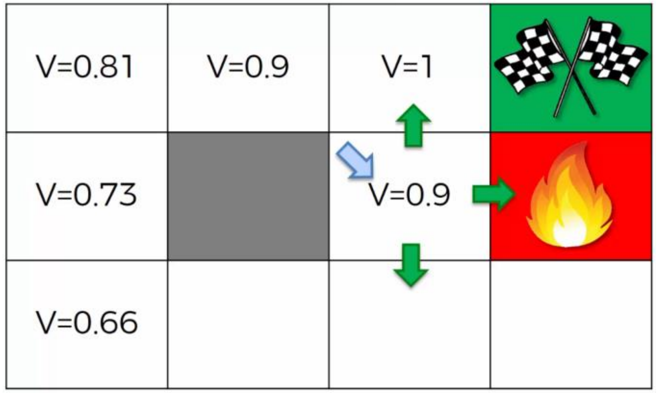
There are 4 possible actions.



→	→	→	🏁
↑	▨	↑	🔥
↑	→	↑	←

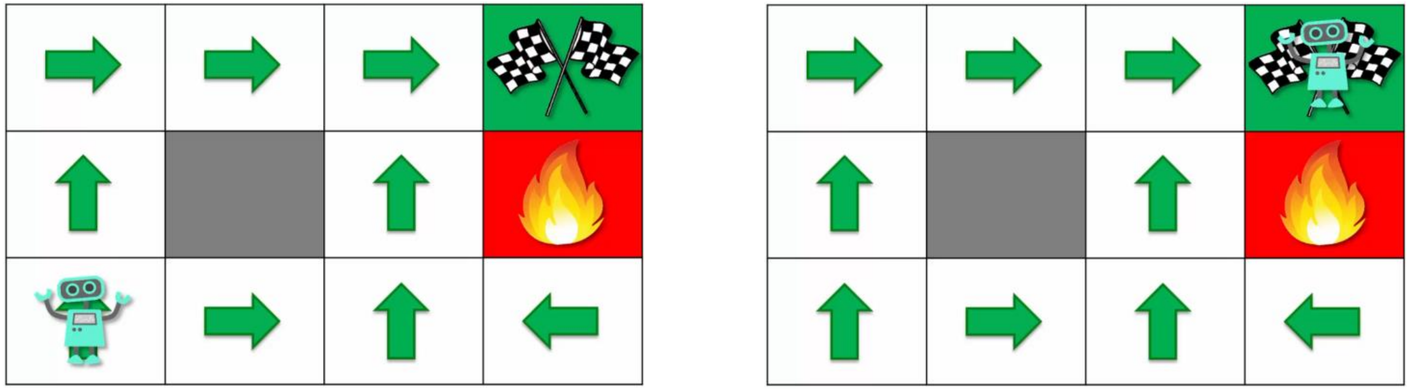


V=0.81	V=0.9	V=1	🏁
V=0.73	▨	V=0.9	🔥
V=0.66	V=0.73	V=0.81	V=0.73



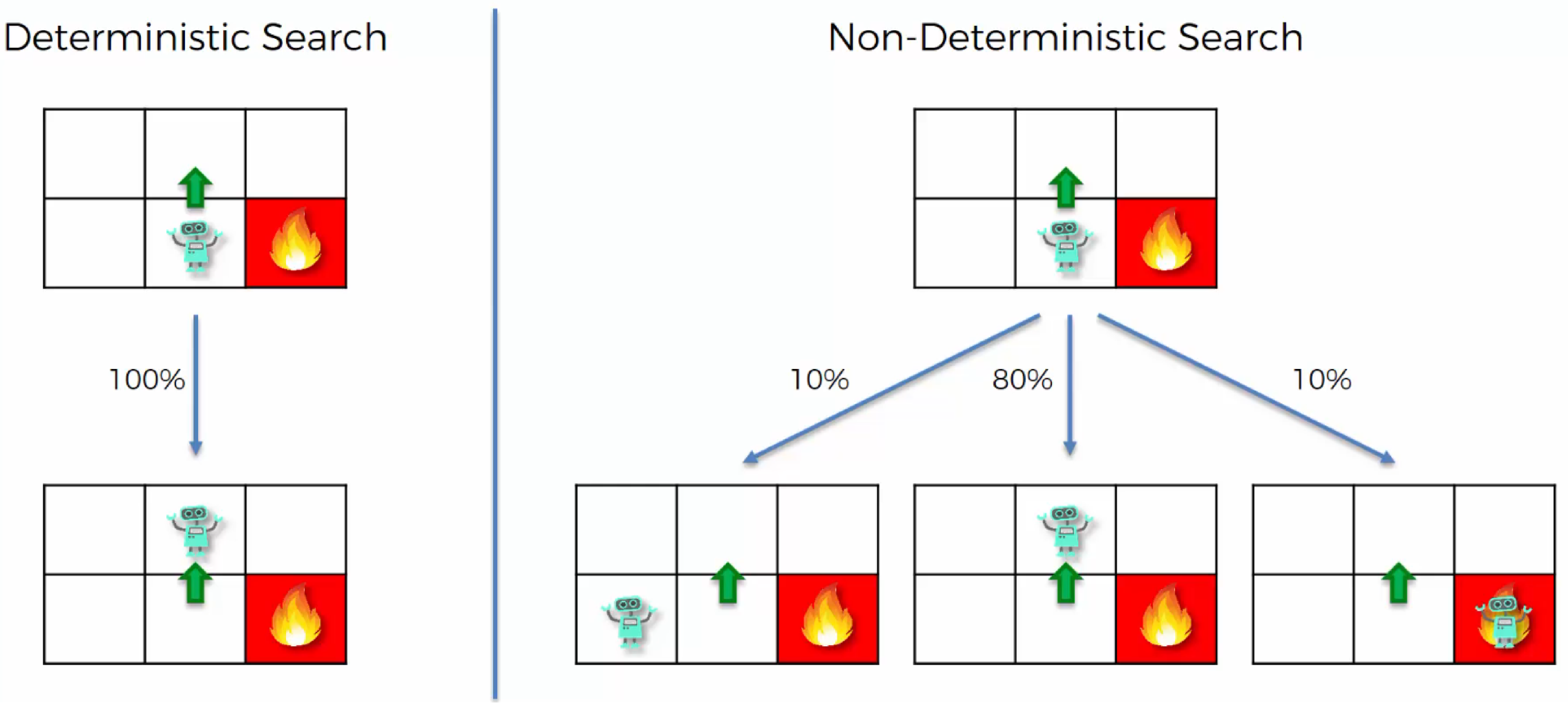
V=0.81	V=0.9	V=1	🏁
V=0.73	▨	V=0.9	🔥
V=0.66			

MARKOV DECISION PROCESS (MDP)



In reality… it is not the case.

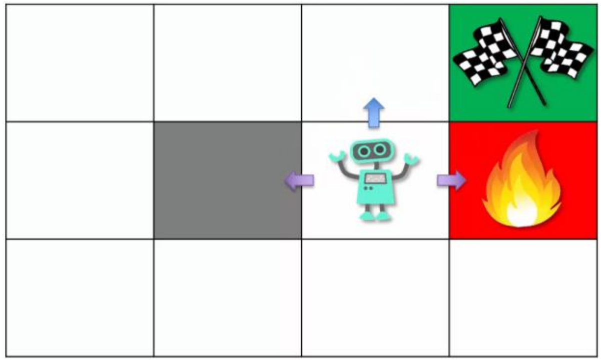
MARKOV DECISION PROCESS (MDP)



MARKOV DECISION PROCESS (MDP)

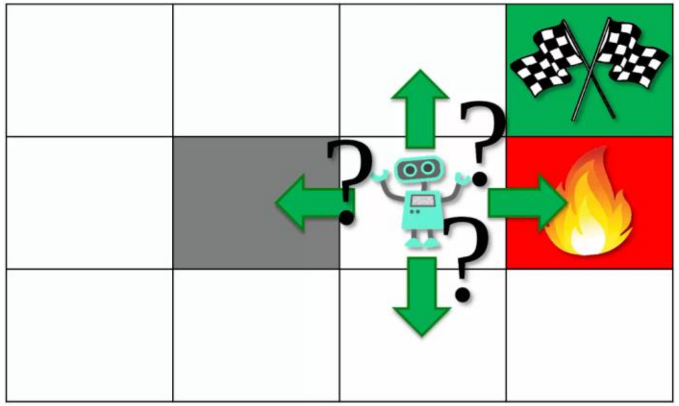
A stochastic process has the **Markov property** if the conditional probability distribution of future states of the process (conditional on both past and present states) depends only upon the present state, not on the sequence of events preceded it. A process with this property is called a **Markov process**.

The state that the agent is in now does not depend on how it gets there. The future only depends on where the agent is and the actions it will take.



MARKOV DECISION PROCESS (MDP)

Markov Decision Processes (MDPs) provide a mathematical **framework** for modeling decision making in situations where outcomes are partly random and partly under the control of a decision maker.



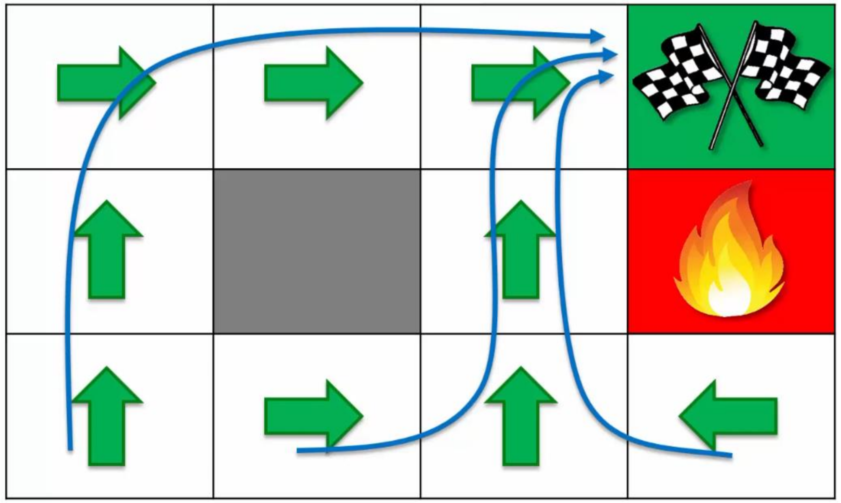
MARKOV DECISION PROCESS (MDP)

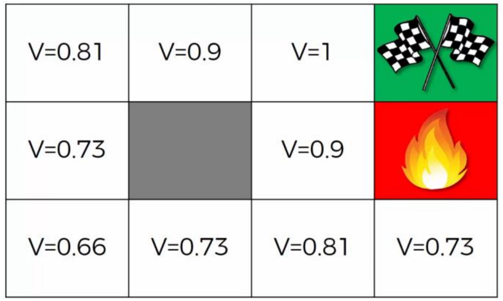
$$V(s) = \max_a(R(s,a) + \gamma \overbrace{V(s')}^{0.8 * V(s'_1) + 0.1 * V(s'_2) + 0.1 * V(s'_3)})$$

$$V(s) = \max_a\left(R(s,a) + \gamma \sum_{s'} P(s,a,s')V(s')\right)$$

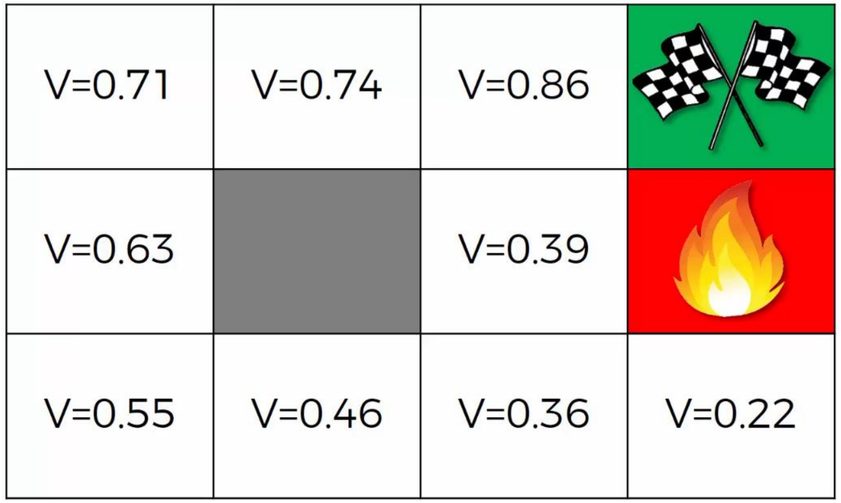
POLICY VS PLAN

$$V(s) = \max_{a}\left(R(s,a) + \gamma \sum_{s'} P(s,a,s')V(s') \right)$$





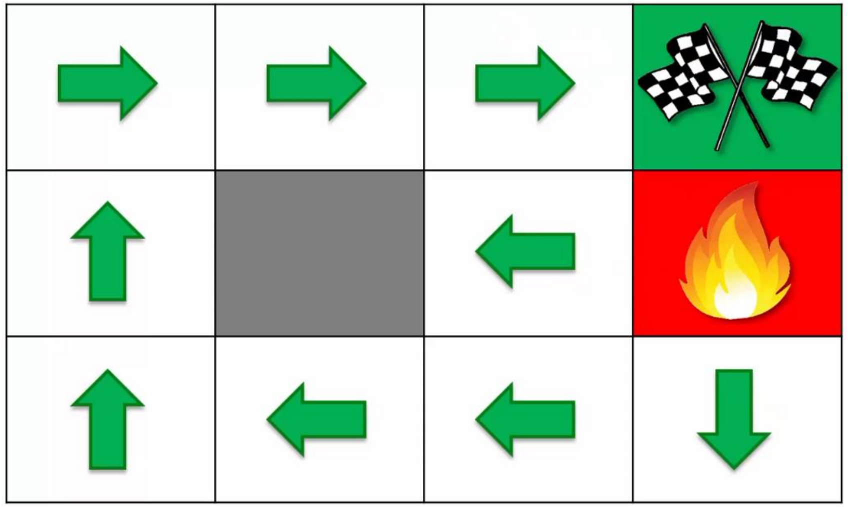
V=0.81	V=0.9	V=1	🏁
V=0.73	▨	V=0.9	🔥
V=0.66	V=0.73	V=0.81	V=0.73

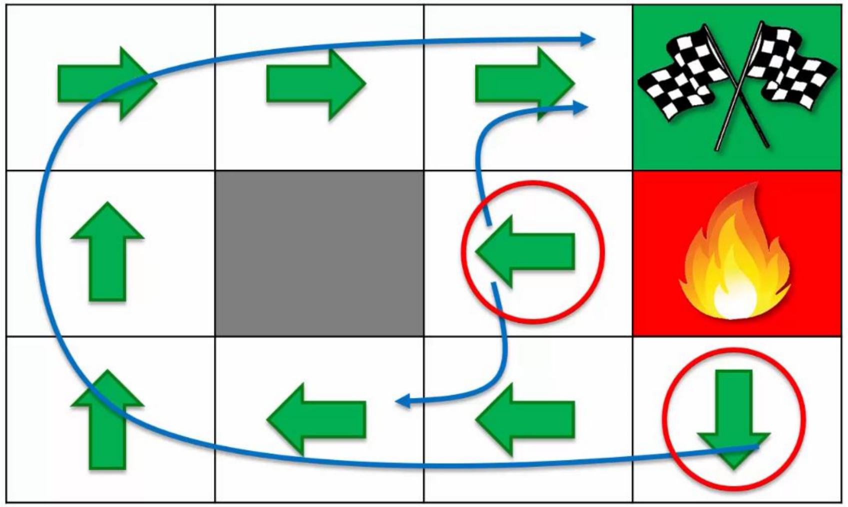


V=0.71	V=0.74	V=0.86	🏁
V=0.63	▨	V=0.39	🔥
V=0.55	V=0.46	V=0.36	V=0.22

A plan is where deterministic search exists. The agent knows the steps from a starting point to a goal.

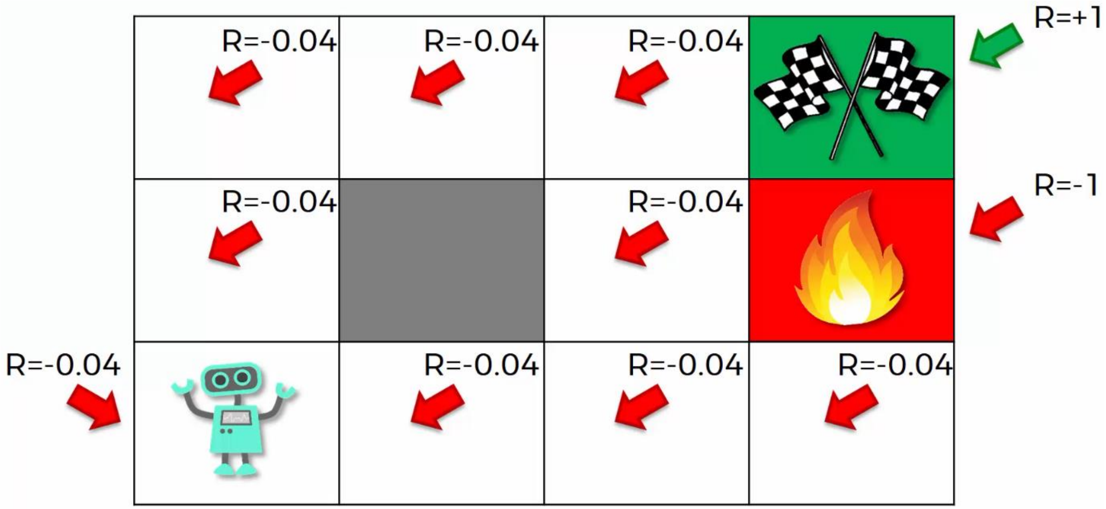
POLICY VS PLAN





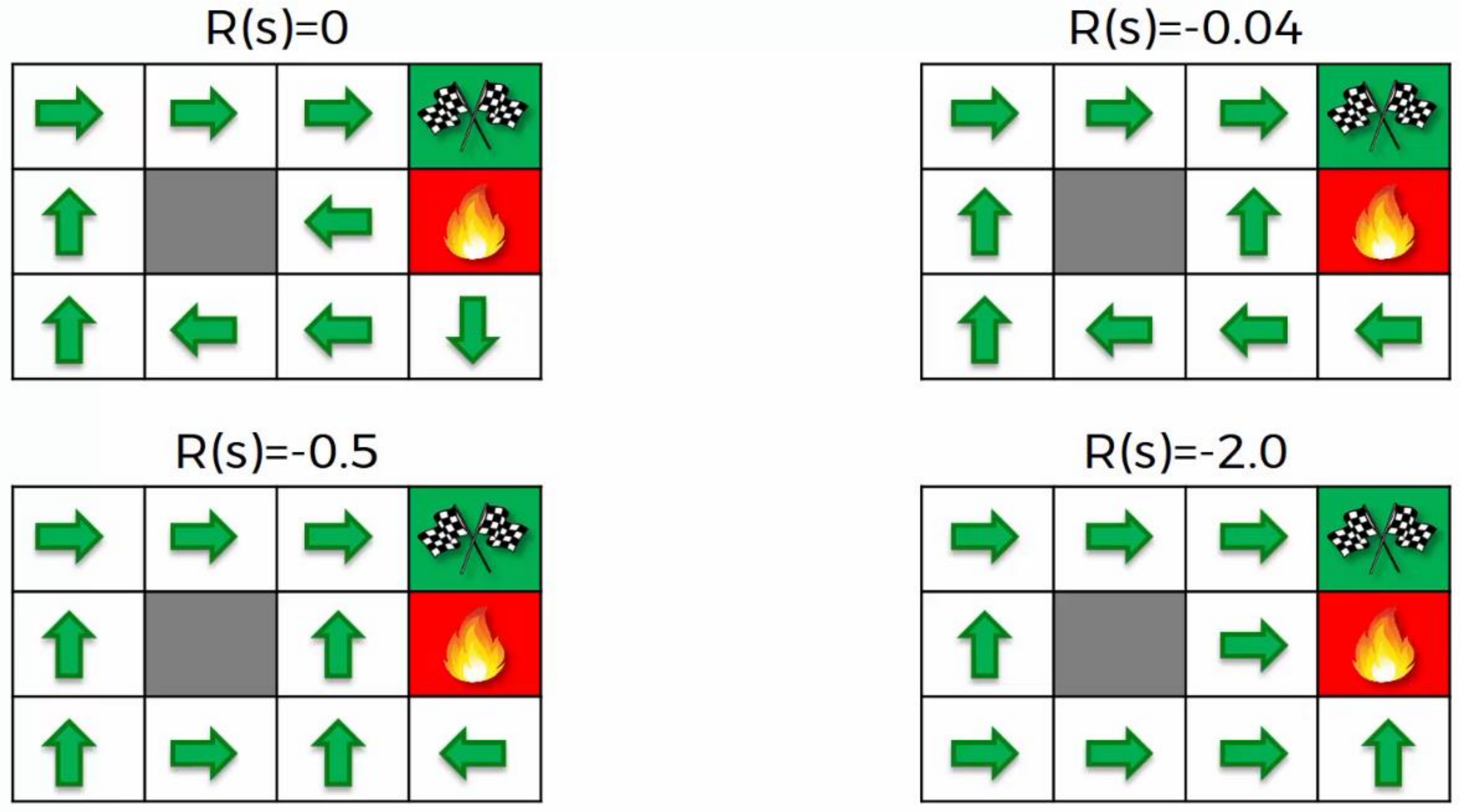
Does the agent want to travel longer or shorter even it knows the high change getting to a firepit?

LIVING PENALTY



Every time the agent moves, it'll get a negative reward. It is called the living penalty because no matter where it goes, it will always get this negative reward, except for those final squares.
So the longer it walks around, the more it accumulates the negative reward.
Therefore, it is an incentive for it to finish the game earlier as quickly as possible.

LIVING PENALTY



R(s)=0

→	→	→	🏁
↑	■	←	🔥
↑	←	←	↓

R(s)=-0.04

→	→	→	🏁
↑	■	↑	🔥
↑	←	←	←

R(s)=-0.5

→	→	→	🏁
↑	■	↑	🔥
↑	→	↑	←

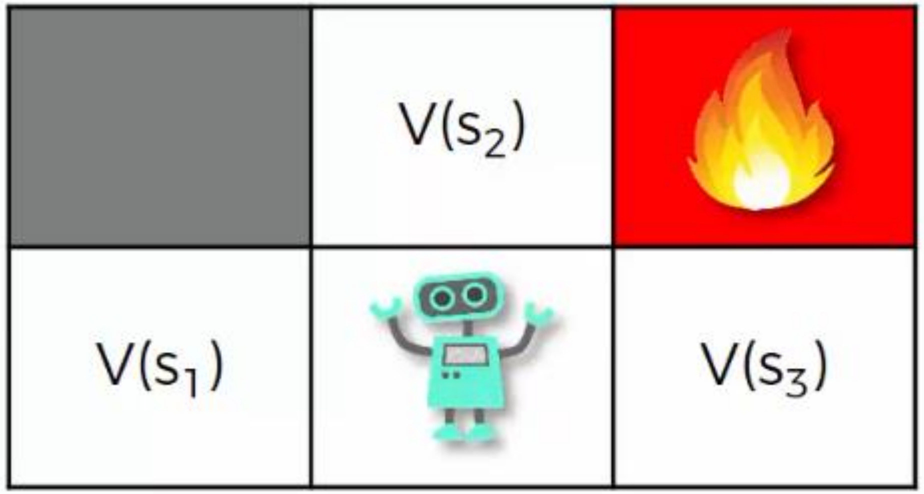
R(s)=-2.0

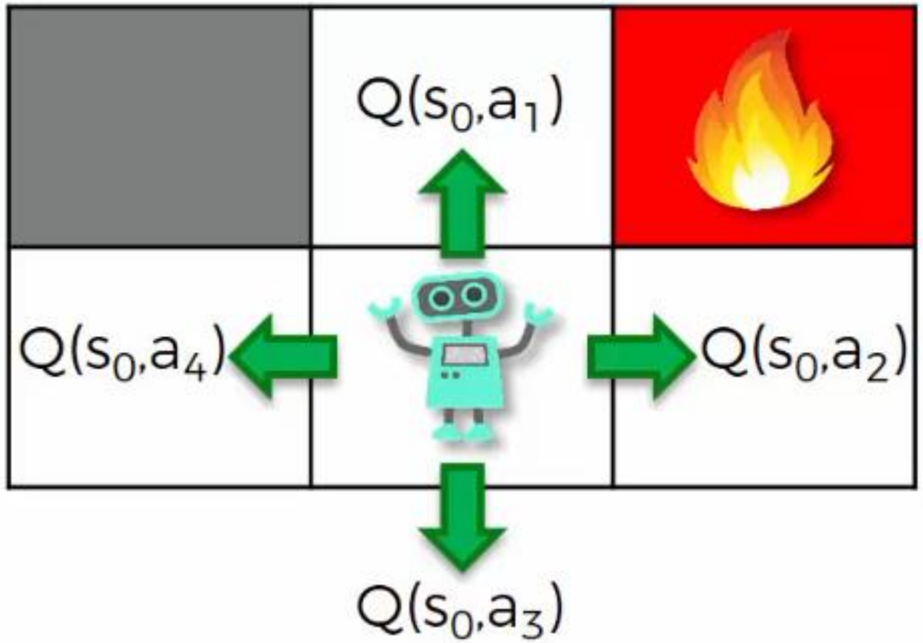
→	→	→	🏁
↑	■	→	🔥
→	→	→	↑

Q-LEARNING INTUITION

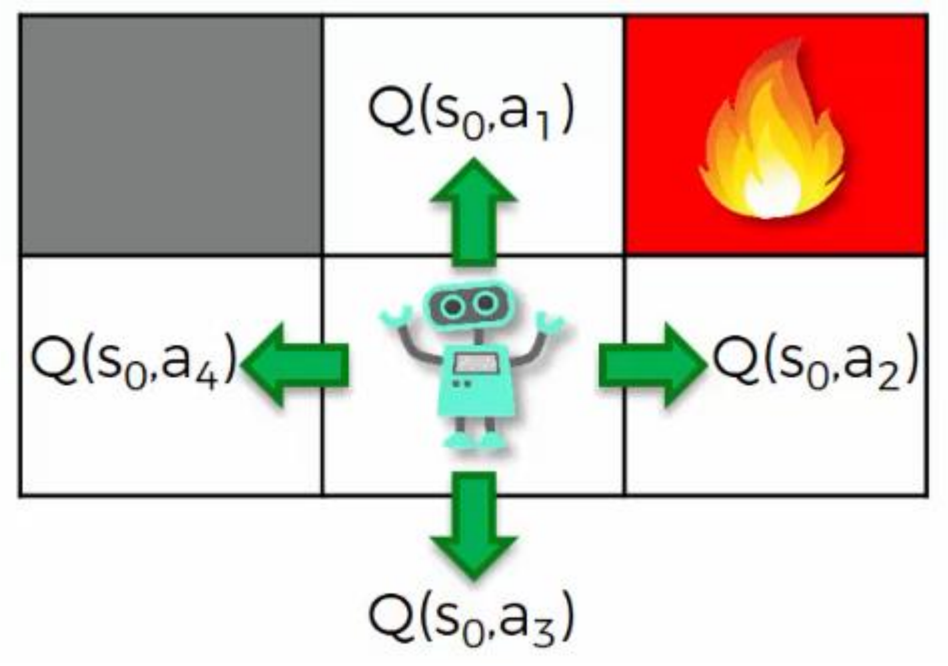
$$V(s) = \max_a \left(R(s,a) + \gamma \sum_{s'} P(s,a,s')V(s') \right)$$

So far, we've been dealing with values, the value of being in a certain state, and now we're going to look at how **Q** fits into all of that.





Q-LEARNING INTUITION

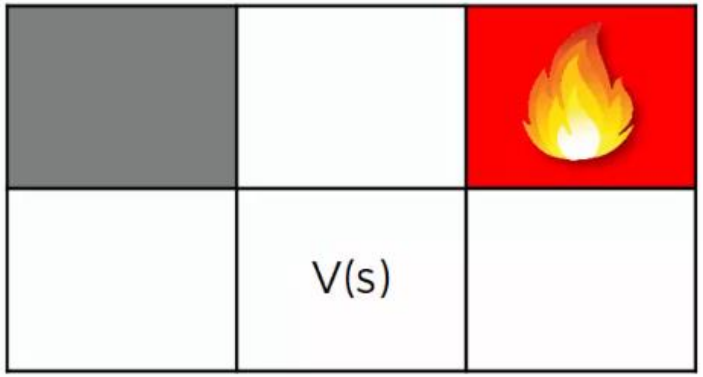


Q represents the quality of the action.

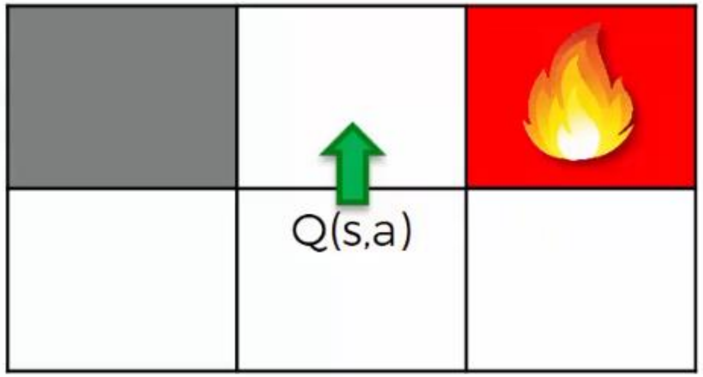
Here, we get 4 actions. What are the different qualities of these actions.

So we need a metric telling how we quantify this action?

Q-LEARNING INTUITION

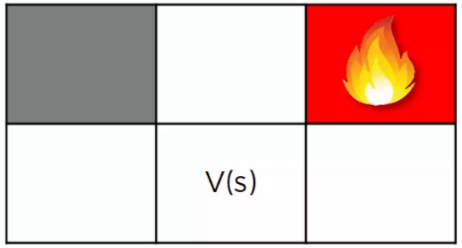


$$V(s) = \max_a \left(R(s,a) + \gamma \sum_{s'} P(s,a,s')V(s') \right)$$

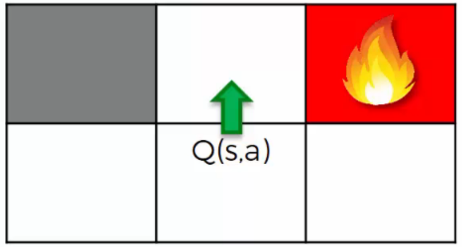


$$Q(s,a) = R(s,a) + \gamma \sum_{s'} \left(P(s,a,s')V(s') \right)$$

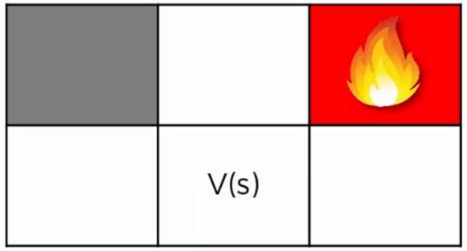
Q-LEARNING INTUITION



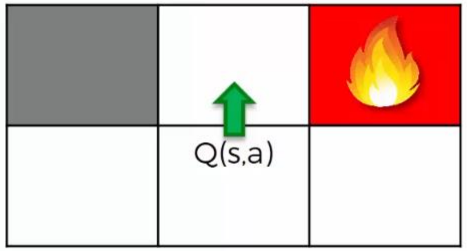
$$V(s) = \max_a \Big(\underbrace{R(s,a) + \gamma \sum_{s'} P(s,a,s')V(s')} \Big)$$



$$Q(s,a) = R(s,a) + \gamma \sum_{s'} \big(P(s,a,s')V(s')\big)$$



$$V(s) = \max_a \Big(R(s,a) + \gamma \sum_{s'} P(s,a,s')V(s') \Big)$$



$$Q(s,a) = R(s,a) + \gamma \sum_{s'} \big(P(s,a,s')\overbrace{V(s')}\big)$$

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} \Big(P(s,a,s') \max_{a'} Q(s',a')\Big)$$

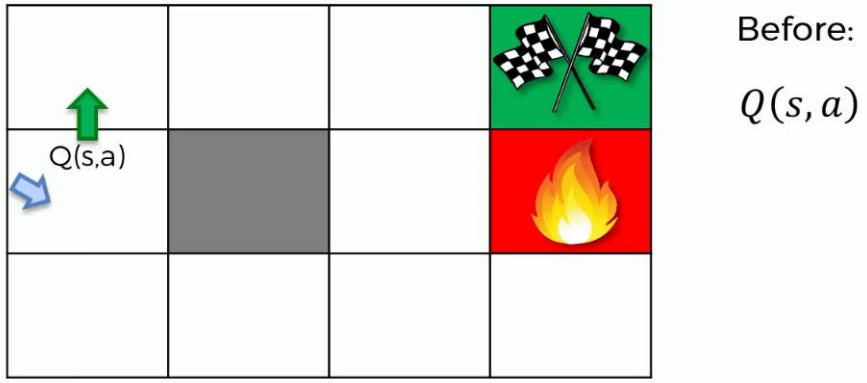
TEMPORAL DIFFERENCE

V=0.71	V=0.74	V=0.86	🏁
V=0.63	▨	V=0.39	🔥
V=0.55	V=0.46	V=0.36	V=0.22

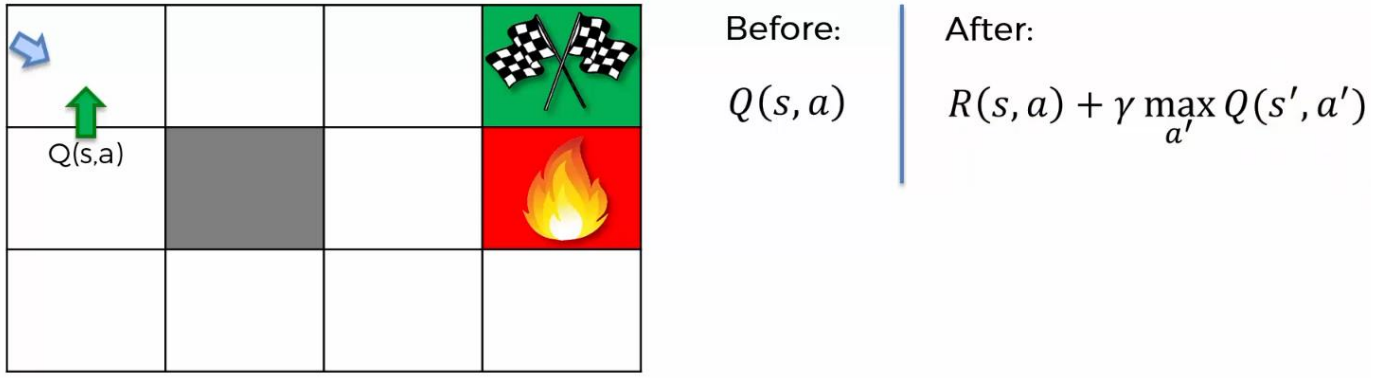
$$Q(s,a) = R(s,a) + \gamma \sum_{s'} \left(P(s,a,s') \max_{a'} Q(s',a') \right)$$

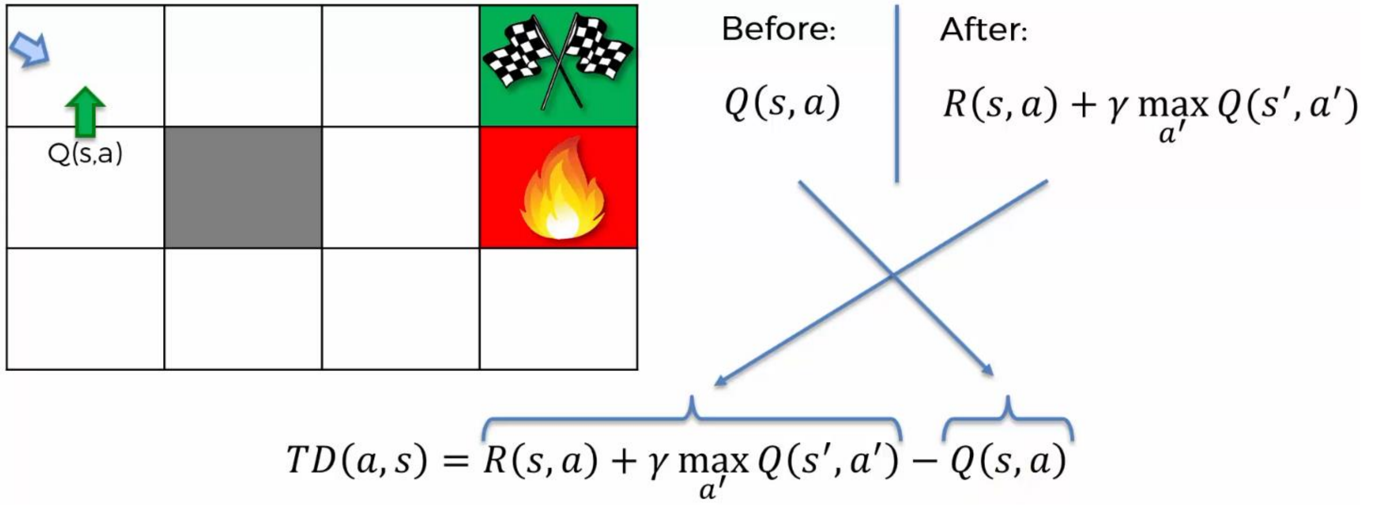
$$Q(s,a) = R(s,a) + \gamma \max_{a'} Q(s',a')$$

TEMPORAL DIFFERENCE



We have Q values calculated by the agent by walking around.
The agent is sitting in the blue-arrow cell and he needs to make a choice.





$$TD(a, s) = \underbrace{R(s, a) + \gamma \max_{a'} Q(s', a')} - \underbrace{Q(s, a)}$$

TEMPORAL DIFFERENCE

$$TD(a,s) = R(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a)$$

The reason it is called the temporal difference is because we basically assumably calculate in different time. In the easiest manner, we assume that they are equal.

$$Q(s,a) = Q(s,a) + \alpha TD(a,s)$$

α is the learning rate.

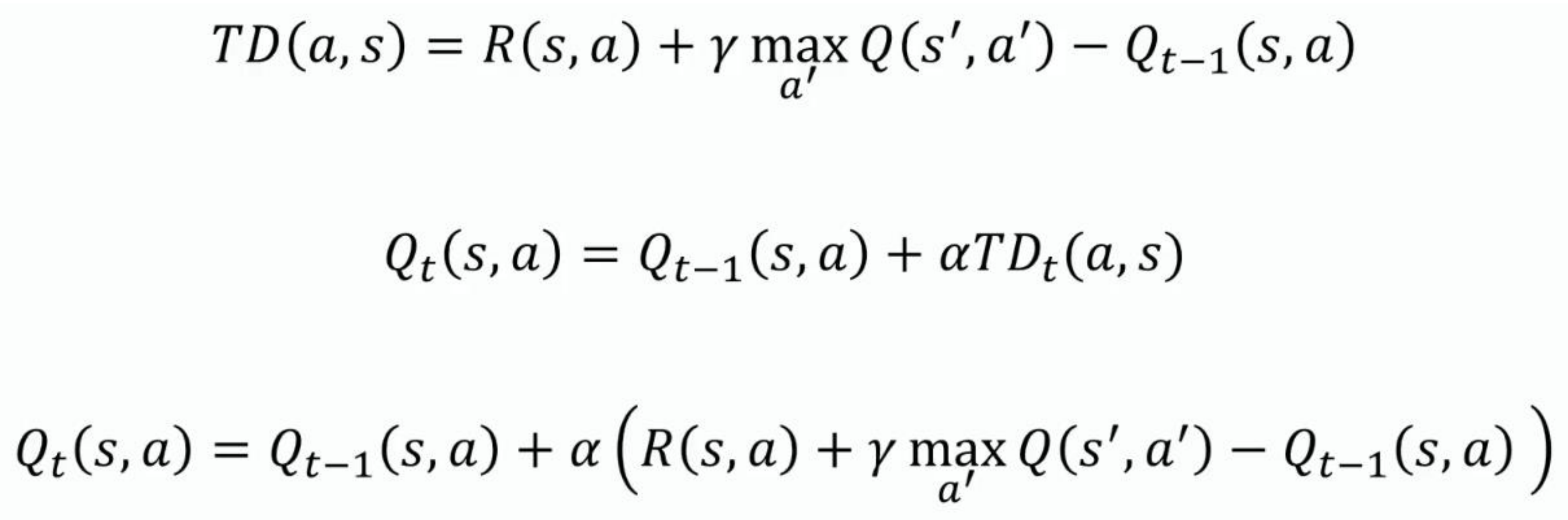
However, in reality, there is a shift in time.
Q(s,a) is “80%”.
R(s,a) + γmaxQ(s’,a’) happens in randomness.

$$TD(a,s) = R(s,a) + \gamma \max_{a'} Q(s',a') - Q_{\boxed{t-1}}(s,a)$$

The key idea is to update Q over time.

$$Q_t(s,a) = Q_{\boxed{t-1}}(s,a) + \alpha TD_{\boxed{t}}(a,s)$$

TEMPORAL DIFFERENCE



$$TD(a,s) = R(s,a) + \gamma \max_{a'} Q(s',a') - Q_{t-1}(s,a)$$

$$Q_t(s,a) = Q_{t-1}(s,a) + \alpha TD_t(a,s)$$

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

α should not be 0 or 1, but a value in between.
When the temporal difference is zero, meaning that the algorithm has converged.
Remember that the environment is changing over time.

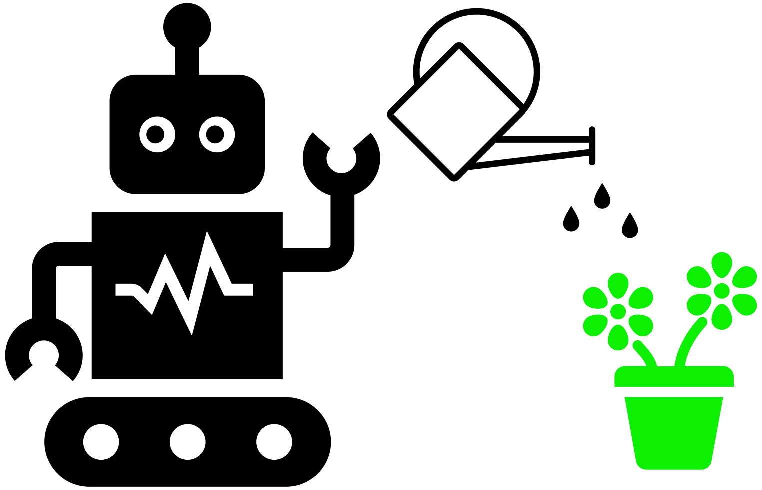
REFERENCE

http://ai.berkeley.edu/reinforcement.html

SESSION 3

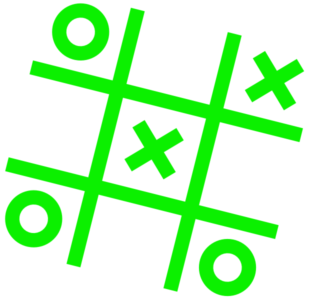
TRANSFER TASK

1) Imagine you have a gardening bot to water your plants. How could the bot apply reinforcement learning to learn how to perfectly water the plants?



TRANSFER TASK

2) You have a computer that learns a game by playing against a random opponent. How would the learning process change if the computer played against another computer using the same algorithm?



TRANSFER TASKS

3) In Q-Learning exploration and exploitation play an important role. What is more important at what stage of the learning process and why?

Refs:

- https://www.youtube.com/clip/Ugkx8aZpwZVuoNNM_ByAJevUIM1AwCJyK_AM
- https://medium.com/@angelina.yang/what-is-exploration-vs-exploitation-in-reinforcement-learning-a3b96dcc9503
- https://ai-ml-analytics.com/reinforcement-learning-exploration-vs-exploitation-tradeoff/

TRANSFER TASK
PRESENTATION OF THE RESULTS

Please present your results.

The results will be discussed in plenary.

TRANSFER TASK
PRESENTATION OF THE RESULTS

Please present your results.

The results will be discussed in plenary.

LEARNING CONTROL QUESTIONS

1. Who performs the actions in reinforcement learning?

a. The agent

b. The policy

c. The present state

d. A model-free approach

LEARNING CONTROL QUESTIONS

2. What component describes which action is picked in a certain state?

a. The policy
b. The environment
c. The value function
d. The agent

3. What do the decisions in MDPs depend on?

a. The history of states

b. The future state

c. The final state

d. The present state

LEARNING CONTROL QUESTIONS

4. What kind of approach is used in temporal difference learning?

a. A model-based approach

b. A model-free approach

c. A data-driven approach

d. A supervised approach

https://www.analyticsinsight.net/model-free-vs-model-based-reinforcement-learning-know-now/