Introduction to Reinforcement Learning

Ens'IA

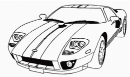
github.com/YannSia

12 avril 2022

Classes of Learning Problems

Supervised Learning

- **Data :** *(x,y)*
- x : data, y : labels
- **Goal** : Learn function $x \longrightarrow y$



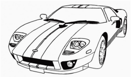
↓

this is a car

Classes of Learning Problems

Supervised Learning

- **Data :** *(x,y)*
- x : data, y : labels
- **Goal** : Learn function $x \longrightarrow y$

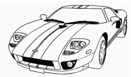


↓

this is a car

Unsupervised Learning

- **Data :** x
- x : data, no label
- **Goal** : Learn underlying structure

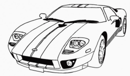


these things are similar

Classes of Learning Problems

Supervised Learning

- **Data :** (x,y)
- x : data, y : labels
- **Goal** : Learn function $x \longrightarrow y$

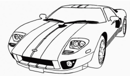


↓

this is a car

Unsupervised Learning

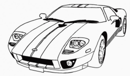
- **Data :** x
- x : data, no label
- **Goal** : Learn underlying structure



these things are similar

Reinforcement Learning

- **Data :** state-action pairs
- **Goal** : Maximize reward



use this to move fast

Examples

Figure 1 – https://www.youtube.com/watch?v=kopoLzvh5jY

Examples

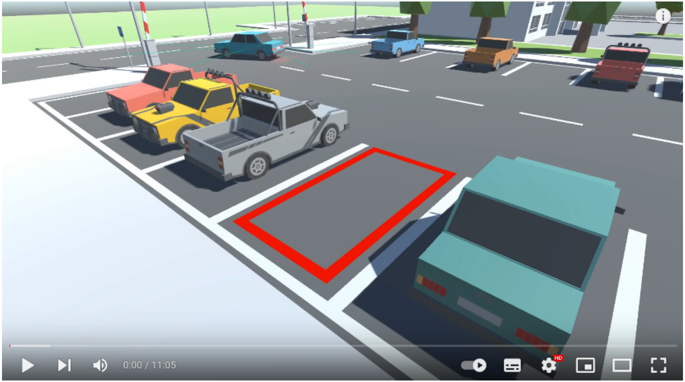


Figure 2 – https://www.youtube.com/watch?v=VMp6pq6_QjI&t=220s

Key Concepts

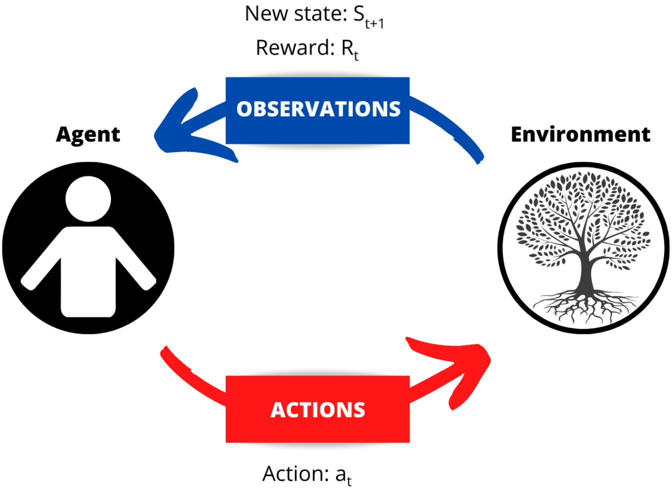


Figure 3 – Learning process

Hypothesis and notations

- **Markov Decision Process (MDP)**
  States : S
  Model : $T(S, a, S') = Prob(S'|S, a)$
  Actions : $A(S)$
  Rewards : $R(S)$ **or** $R(S, a)$ **or** $R(S, a, S')$
- **Infinite horizon**
- **We sum rewards**

Reward

How to define the total reward = what the agent will get ?

Reward

How to define the total reward = what the agent will get ?

$$R_t = r_t + r_{t+1} + r_{t+2} + ...$$

Reward

How to define the total reward = what the agent will get ?

$$R_t = r_t + r_{t+1} + r_{t+2} + ...$$

How to encourage quick high rewards ?

Reward

How to define the total reward = what the agent will get ?

$$R_t = r_t + r_{t+1} + r_{t+2} + ...$$

How to encourage quick high rewards ?

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + ...$$

Reward

How to define the total reward = what the agent will get ?

$$R_t = r_t + r_{t+1} + r_{t+2} + ...$$

How to encourage quick high rewards ?

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + ...$$

This is called **discounted sum of reward**

Example 1

+2	+2	🟦 +2	🟩 +1
+2	▨	🟦 +2	🟥 -1
+2	+2	🟦 +2	🟦 +2

green and red : final states

gray : forbidden

blue : **Where should we go ?**

Example 1

+2	+2	🟦 +2	🟩 +1
+2	▨	🟦 +2	🟥 -1
+2	+2	🟦 +2	🟦 +2

green and red : final states
gray : forbidden
blue : **Where should we go ?**

$$\gamma = 1$$

Example 1

+2	+2	←	+1
+2	▨	↕	-1
+2	+2	X	←

green and red : final states

gray : forbidden

blue : Where should we go ?

It's better to play for ∞

Example 2

-2	-2	-2	+1
-2		-2	-1
-2	-2	-2	-2

green and red : final states

gray : forbidden

blue : **Where should we go ?**

Example 2

-2	-2	→	+1
-2	▨	→	-1
-2	-2	→ or ↑	↑

green and red : final states

gray : forbidden

blue : Where should we go ?

We should end the game

Example 3

-0.01	-0.01	-0.01	+1
-0.01	▨	-0.01	-1
-0.01	-0.01	-0.01	-0.01

green and red : final states

gray : forbidden

blue : start

Example 3

-0.01	-0.01	-0.01	+1
-0.01	▨	-0.01	-1
-0.01	-0.01	-0.01	-0.01

difficulty : when doing an action :
10% change to go on both perpendicular directions
What are the best actions to take to maximize the score ?

Example 3

→	→	→	+1
↑	▨	↑	-1
↑	←	←	←

Conclusion

Hard to find the best action to perform !

How to learn the *quality* of a state-action pair ?

Q-Learning

$$Q(s_t, a_t) = \mathrm{E}[R_t | s_t, a_t]$$

This is the **Quality of a state-action pair**

Q-Learning

$$Q(s_t, a_t) = \mathrm{E}[R_t | s_t, a_t]$$

This is the **Quality of a state-action pair**

$$\pi^*(s) = \arg\max_a Q(s, a)$$

This is the **optimal policy**

Q-Learning - Bellman Equations

$$Q(S,a) = R(S,a) + \gamma \sum_{S'} T(S,a,S') \max_{a'} Q(S',a')$$

Q-Learning - Bellman Equations

$$Q(S,a) = R(S,a) + \gamma \sum_{S'} T(S,a,S') \max_{a'} Q(S',a')$$

How to **learn** the Q-value ?

Q-Learning - Bellman Equations

$$Q(S, a) = R(S, a) + \gamma \sum_{S'} T(S, a, S') \max_{a'} Q(S', a')$$

How to **learn** the Q-value ?

$$Q(S, a) \leftarrow Q(S, a) + \alpha[r + \gamma \max_{a'} Q(S', a') - Q(S, a)$$

α : learning rate

How to make the agent learn ?

The agent doesn't know what are all the states. It just know where it is and what it can do

How to make the agent learn ?

The agent doesn't know what are all the states. It just know where it is and what it can do

We can only update the Q-value of the state the agent has seen.

Exploration VS Exploitation

Solution :

$$\epsilon \in]0, 1[$$

$$a_t = \left\{ \begin{array}{ll} \arg\max(Q_t) & \text{if } random() < \epsilon \\ \text{random action} & otherwise \end{array} \right.$$

Want to learn more ?

- Deep Q-Learning
- Double Q-Learning

Sources

- © Alexander Amini and Ava Soleimany
  MIT 6.S191 : Introduction to Deep Learning
  `IntroToDeepLearning.com`
- Udacity : Reinforcement Learning course ud600 (Georgia Tech CS 8803)
  `https://classroom.udacity.com/courses/ud600`
- `https://towardsdatascience.com/introduction-to-reinforcement-learning-c99c8c0720ef`