in104

Introduction to Reinforcement Learning

Florence Carton

April 10. 2018

ENSTA ParisTech

Florence Carton
ENSTA 2016,5

Thèse CEA / ENSTA
Sujet : Exploration of reinforcement learning algorithms for drone visual perception and control

mail : fcarton@ensta.fr
site : http://perso.ensta-paristech.fr/~fcarton/

U2IS , bureau R211 (jeudi)

table of contents

introduction

Intro to Reinforcement Learning

Sujet difficile : c'est normal de ne pas tout comprendre tout de suite

definitions

different types of machine learning algorithms

Unsupervised learning

We don't know anything

E.g. : classification

(Google news)

Supervised learning

Reinforcement learning

google news

Google News

Search

Headlines | Local | For You | U.S.

SECTIONS

- Top Stories
- World
- U.S.
- Business
- Technology
- Entertainment
- Sports
- Science
- Health

Manage sections

Top Stories

Trump tweets condemnation of Syria chemical attack, saying Putin shares the blame

Washington Post · 54m ago

RELATED COVERAGE

Jaish al-Islam to leave Douma in return for releasing prisoners

Local Source · Reuters · 1h ago

Trump threatens "Animal Assad," Putin over alleged chemical attack in Syria

CBS News · 1h ago

Trump Warns of 'Price to Pay' After Suspected Gas Attack in Syria

Wall Street Journal · 55m ago

Syria attack shows we now live in a world where using chemical weapons is considered OK -- and that should terrify us

Opinion · Fox News · 29m ago

View full coverage →

MORE ABOUT

- Syria
- Duma
- Chemical warfare

Blaze on 50th floor of Trump Tower in New York kills 1

Washington Post · 13h ago

In the News

- Syria
- Trump Tower
- Germany
- Duma
- Münster
- New York City
- Chemical warfare
- Donald Trump
- Bashar al-Assad
- Ghouta

Recent

One killed, dozens wounde

different types of machine learning algorithms

Unsupervised learning

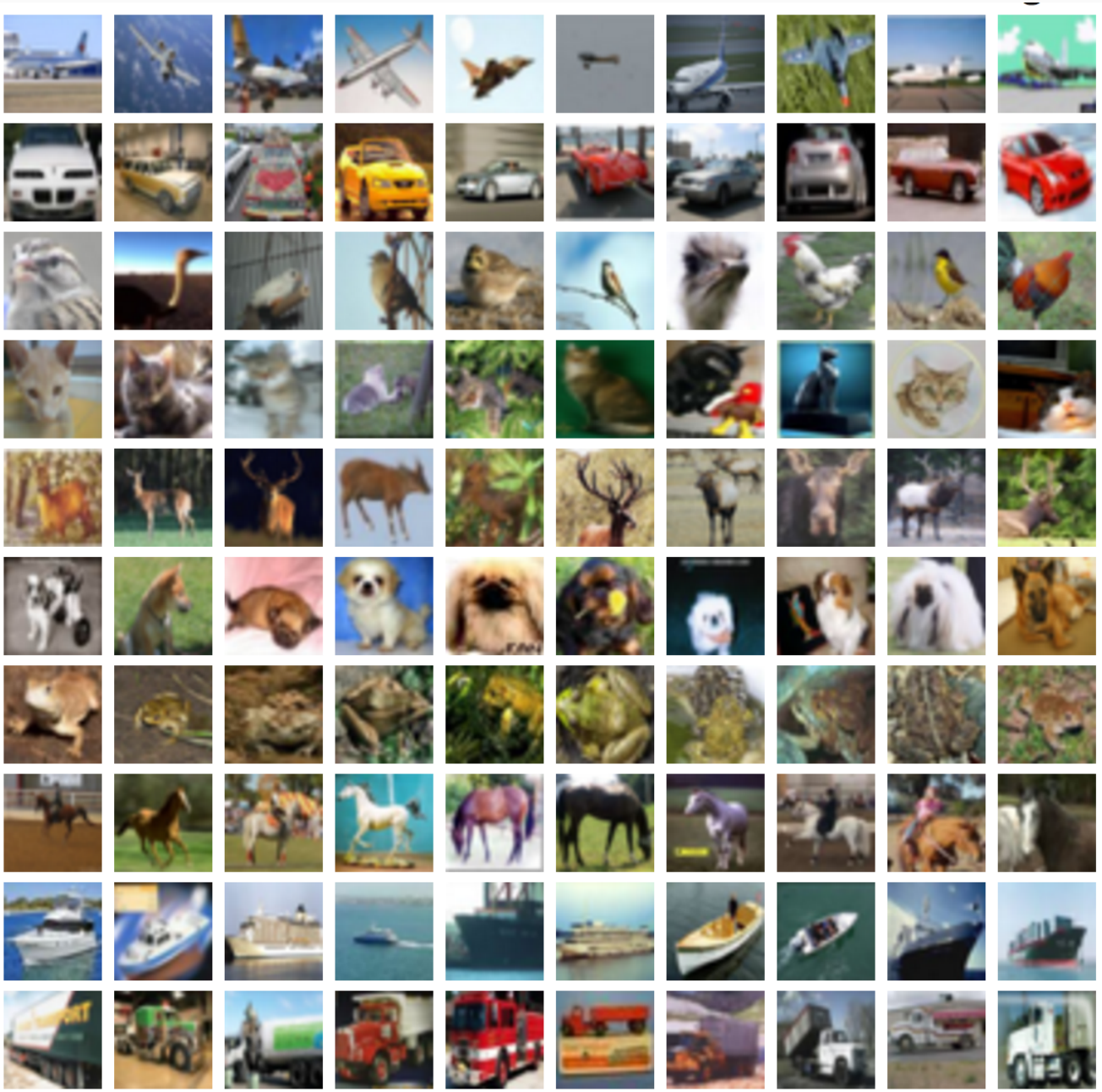
We don't know anything
E.g. : classification
(Google news)

Supervised learning

Groundtruth known
E.g : image recognition

Reinforcement learning

image recognition



different types of machine learning algorithms

Unsupervised learning

We don't know anything
E.g. : classification
(Google news)

Supervised learning

Groundtruth known
E.g : image recognition

Reinforcement learning

You know what you want but not HOW to get it $\rightarrow$ use of a reward function
E.g : board games

chess

supervised vs reinforcement

Supervised learning

- Requires datas
- Must adress distributional shift
- Simple and stable
- Only as good as the demo

Reinforcement Learning

- Requires reward function
- Must adress exploration
- Potentially non convergent
- Can become arbitrarily good

[0] slide inspired from Berkeley Lecture CS294 `http://rll.berkeley.edu/deeprlcourse/`

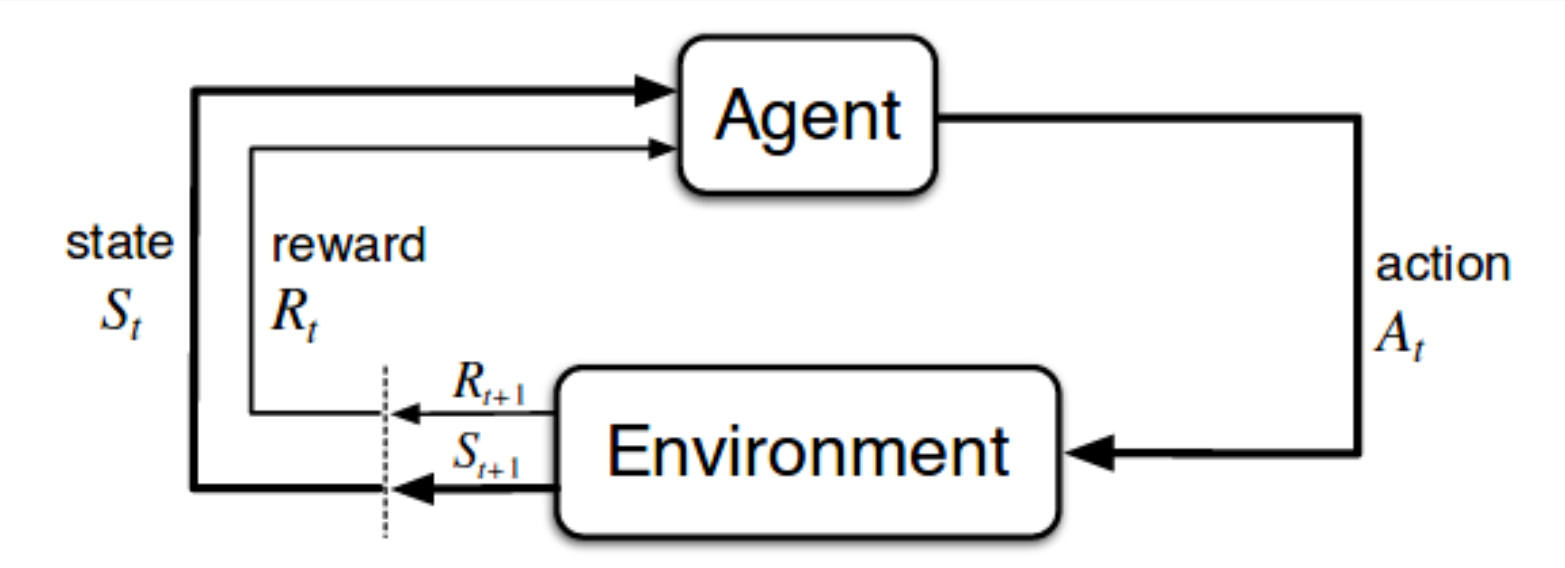


Figure 1: RL diagram

Algorithm from '*Reinforcement Learning : An Introduction*' by Richard Sutton

Policy

How the agent behaves : $\pi(a|s) = Pr(action = a|state = s))$

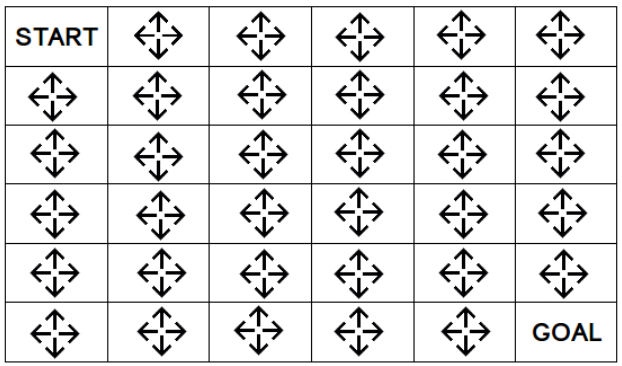


Figure 2: Random policy

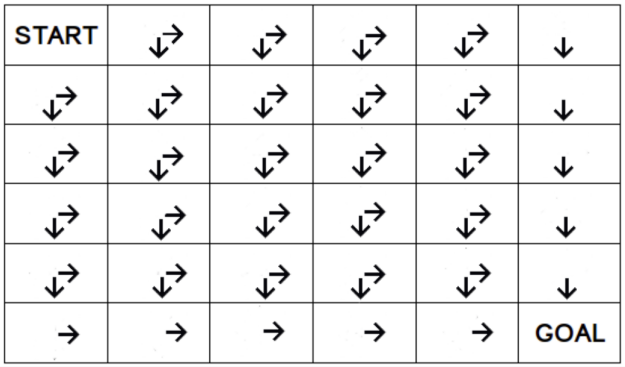


Figure 3: Optimal Policy

What we want = Optimal Policy !

definitions

- Reward : defines what is good or bad for the agent
- Return
  $R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ...$
  $= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$
- Trajectory (or episode):
  $\tau = (s_0, a_0, r_0, ..., a_{T-1}, r_{T-1}, s_T)$
- Value function
  $V_\pi(s) = \mathbb{E}_\pi[R_t | S_t = s]$
- State-Value (Quality) function
  $Q_\pi(s, a) = \mathbb{E}_\pi[R_t | S_t = s, A_t = a]$

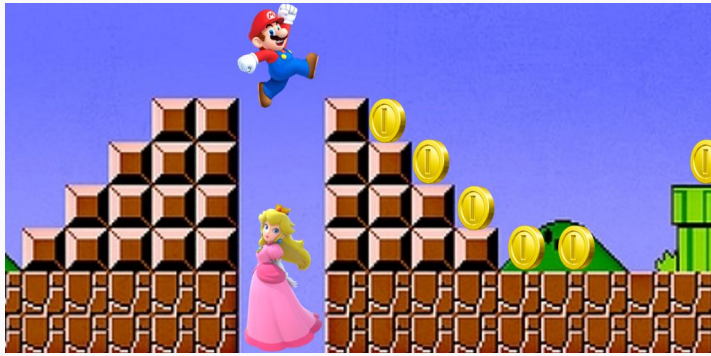


Figure 4: Return vs reward

START	1	2	3	4	5
1	2	3	4	6	6
2	3	4	5	6	7
3	4	5	6	7	8
4	5	6	7	8	9
5	6	7	8	9	GOAL

Figure 5: Value function

q learning

state\actions	up	down	right	left
state s_1	10	12	0	3
state s_2	3	20	4	1
...	...	...	...	...
state s_n	...	...	...	...

Figure 6: Q table

Optimal policy $\pi^*(s) = argmax_a Q(a, s)$

exploration vs exploitation

I want to go to a restaurant

Exploitation : I go to my favorite restaurant

Exploration : I go to a random restaurant

ϵ - **greedy policy**

Picking the best action ($=$ greedy action) with probability $1 - \epsilon$ and a random action with probability ϵ (ϵ from 0.01 to 0.1 are common choices).

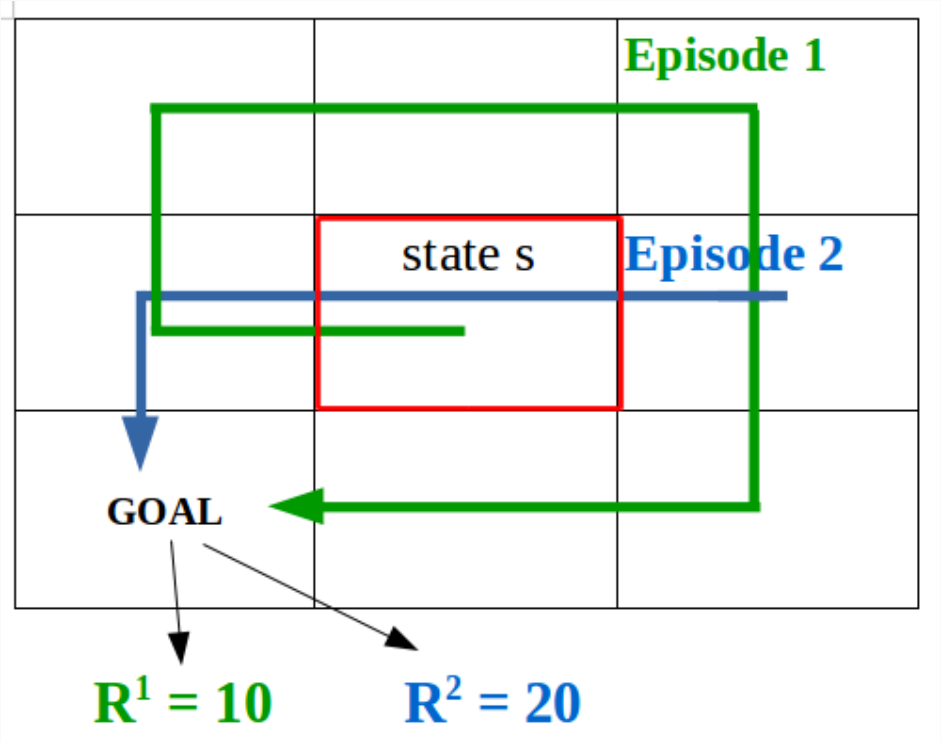
$$action = \begin{cases} random\_action & \text{with proba } \epsilon \\ best\_action & \text{with proba } 1 - \epsilon \end{cases}$$

algorithms

monte carlo algorithm

Idea : Q(s,a) = mean return starting from state s and performing action a.

$$Q(s, a) = \frac{R^1 + R^2 + ... + R^N}{N}$$



$$Q(s, action = left) = \frac{R^1 + R^2}{2} = 15$$

Q - learning

Idea : update at every step

$$Q(a, s) = Q(a, s) + \alpha \times (updating\_term)$$

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r + \gamma max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

SARSA

Idea : On-policy version of Q-learning

$$Q(a, s) = Q(a, s) + \alpha \times (updating\_term(\pi))$$

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

structure du projet

main.py : fonction principale : pour lancer les codes
params.py : fichier contenant les paramètres par défaut
environements
|– Environment.py : contient la classe de base Environment
|– EnvironmentGrid1D : environement simple 1D
|– load_env.py : contient une fonction pour charger l'environement demandé
agents
|– Agent.py : contient la classe de base Agent
|– AgentRandom.py : contient un agent aléatoire
|– load_agent.py : fonction pour charger un agent

work to do

Implémenter un ou plusieurs algorithmes d'apprentissage par renforcement pour résoudre le déplacement dans un labyrinthe.
Note : pour le rendu graphique il faut installer Openai Gym (`https://github.com/openai/gym`) (ou recoder avec les librairies python (tkinter))

work to do

Pistes pour aller plus loin :

- Rendu graphique
- Faire décroitre le taux d'exploration ϵ
- Résoudre l'environement Cartpole d'Openai Gym
- Faire un fichier de paramètres
- ...

evaluation

Projet et rapport à terminer pour le 5 Juin à minuit.

- Aboutissement du projet
- Clareté du code (commentaires, Readme, noms des variables...)
- Utilisation de Git
- Présentation
- Rapport

programme

- 10 Avril : Prise en main du code
- 17 Avril : Code Environement2D + 1 Agent
- 15 Mai : 2e Agent - Evaluation des commits git
- 22 Mai : Code Labyrinthe + 3e Agent
- 29 Mai : Améliorations

Questions?