

UE Introduction à l'Intelligence Artificielle

Master 1, Semestre 1, 3 ECTS

Code UE : MU4RBI04

Chargé de Cours et resp. UE : Prof. Daniel Racocanu

Chargé de TP : Gabriel Jimenez

Version : oct. 2021

Travaux Pratiques (TPs) / *Laboratories (Labs)***TP1 - Processus markoviens et apprentissage par renforcement*****Markovian Decision Process and Reinforcement Learning***

L'objectif de ce TP est de découvrir la manière de déclarer et résoudre les processus markoviens (Markov Decision Process = MDP) ainsi que les différents aspects de l'apprentissage par renforcement (Reinforcement Learning = RL).

The objective of this lab is to discover how to declare and solve Markov Decision Process (MDP) as well as the different aspects of reinforcement learning (Reinforcement Learning = RL).

Tous les exemples et exercices de codage sont en Python. Suivez attentivement les instructions du fichier **README.md** car il contient les commandes nécessaires pour configurer votre environnement avec toutes les dépendances nécessaires.

*All the coding examples and exercises are in Python. Follow carefully the instruction of the **README.md** file as it has the necessary commands to set up your environment with all the dependencies needed.*

Exo. # 1 : Monde gelé / *Frozen world*

L'objectif de l'exercice est de mettre en œuvre l'algorithme Q-learning pour faire parcourir à un agent un chemin gelé en évitant les trous jusqu'à ce qu'il atteigne l'objectif. Les instructions se trouvent dans le dossier **exo-01** et dans le fichier **exo-01.ipynb**. Toutes les sections où vous devez commenter et coder sont explicitement signalées. Vous pouvez modifier d'autres parties du code, mais assurez-vous de les commenter.

*The objective of the exercise is to implement the Q-learning algorithm to make an agent walk through a frozen path avoiding the holes until it reaches the goal. The instructions are in the folder **exo-01** and file **exo-01.ipynb**. All the sections where you need to comment and code are explicitly remarked. You may modify other parts of the code, but make sure you comment about it.*

Exo. # 2 : Blackjack

L'objectif de l'exercice est de comparer deux algorithmes pour résoudre un problème d'apprentissage par renforcement. Les algorithmes sont : SARSA et Q-learning. Le problème d'apprentissage par renforcement est le jeu Blackjack. Suivez attentivement les instructions du fichier **exo-02.ipynb** dans le dossier **exo-02**.

*The objective of the exercise is to compare two algorithms to solve a reinforcement learning problem. The algorithms are: SARSA and Q-learning. The reinforcement learning problem is the game Blackjack. Follow carefully the instructions of the file **exo-02.ipynb** inside the folder **exo-02**.*

3 (optionnel / *optional*) : Grid world

Différents types de labyrinthes sont présentés et un agent doit les compléter le plus rapidement possible. Complétez la fonction **def update_Q (Q_table , alpha , gamma , state , action , reward , new_state , new_action)** dans le fichier **exo-03.py**. Commentez les algorithmes mis en œuvre pour résoudre le processus d'apprentissage par renforcement, ainsi que les composants du processus décisionnel de Markov présentés dans le problème. Téléchargez les graphiques que vous jugez pertinents pour votre analyse.

*Different types of maze are presented and an agent needs to complete them as fast as possible. Complete the function **def update_Q (Q_table , alpha , gamma , state , action , reward , new_state , new_action)** inside the file **exercise-03.py**. Comment about the algorithms implemented to solve the reinforcement learning process, as well as the components of the Markov Decision Process presented in the problem. Upload the plots you consider relevant for your analysis.*

NB :

Les CRs du TP vont consister en les codes-source Python ou Jupyter des Exos 1 et 2 (avec vos propres modifications, rajouts et commentaires, intégrés). Les 2 CRs est à déposer sur Moodle (dossier CR_TP1_votre_option (ISI, SAR, IPS, app)).

The reports of the Lab will consist of the Python or Jupyter source codes of the Exo 1 and Exo 2 (with your own adds, modifications, comments, integrated). The 2 reports are to be placed on Moodle (CR_TP1_your_option (ISI, SAR, IPS, app) folder).

Exemples en Matlab (optionnel : pour votre information et votre culture générale)

Markov Decision Process Toolbox

Functions related to the resolution of discrete-time Markov Decision Processes.

<https://fr.mathworks.com/matlabcentral/fileexchange/25786-markov-decision-processes-mdp-toolbox>

createMDP

Create Markov decision process model

https://fr.mathworks.com/help/reinforcement-learning/ref/createmp.html?s_tid=srchtitle

createGridWorld

Create a two-dimensional grid world for reinforcement learning

<https://fr.mathworks.com/help/reinforcement-learning/ref/creategridworld.html>

rlMDPEnv

Create Markov decision process environment for reinforcement learning

<https://fr.mathworks.com/help/reinforcement-learning/ref/rl.env.rlmdpenv.html>

Reinforcement Learning Toolbox

Design and train policies using reinforcement learning

<https://fr.mathworks.com/products/reinforcement-learning.html>

Get Started

Learn the basics of Reinforcement Learning Toolbox

MATLAB Environments

Model reinforcement learning environment dynamics using MATLAB

Simulink Environments

Model reinforcement learning environment dynamics using Simulink models

Agents

Create and configure reinforcement learning agents using common algorithms, such as SARSA, DQN, DDPG, and A2C

Policies and Value Functions

Define policy and value function representations, such as deep neural networks and Q tables

Training and Validation

Train and simulate reinforcement learning agents

Policy Deployment

Code generation and deployment of trained policies

Exemples présents dans Matlab :

https://fr.mathworks.com/help/reinforcement-learning/examples.html?category=getting-started-with-reinforcement-learning-toolbox&s_tid=CRUX_topnav

Train Reinforcement Learning Agent in Basic Grid World

Train Reinforcement Learning Agent in MDP Environment

Create Simulink Environment and Train Agent

Exemple d'apprentissage d'un robot en marche bipède :

<https://fr.mathworks.com/help/reinforcement-learning/ug/train-biped-robot-to-walk-using-reinforcement-learning-agents.html>