Kalle Prorok/2020-04-27

Lab 3 – Q-learning Mountain car

Objective

In this lab you have access to some code (both as ipynb and py) that implement basic Q-learning for the Open AI gym Mountain car. It is supposed to work reasonable ok. You should understand the code and then play around and improve it. The mountain car is a control problem with a car with underpowered engine going uphills within less than 200 time steps. (it needs to learn to gain momentum by going in the opposite direction first)

Example output from a run:

```
Q shape: (19, 15, 3), first 2 is state and 3rd is action.

0.00016

Episode 100 Average Reward: -200.0 Epsilon: 0.7839999999999951

Episode 200 Average Reward: -200.0 Epsilon: 0.7679999999999902
```

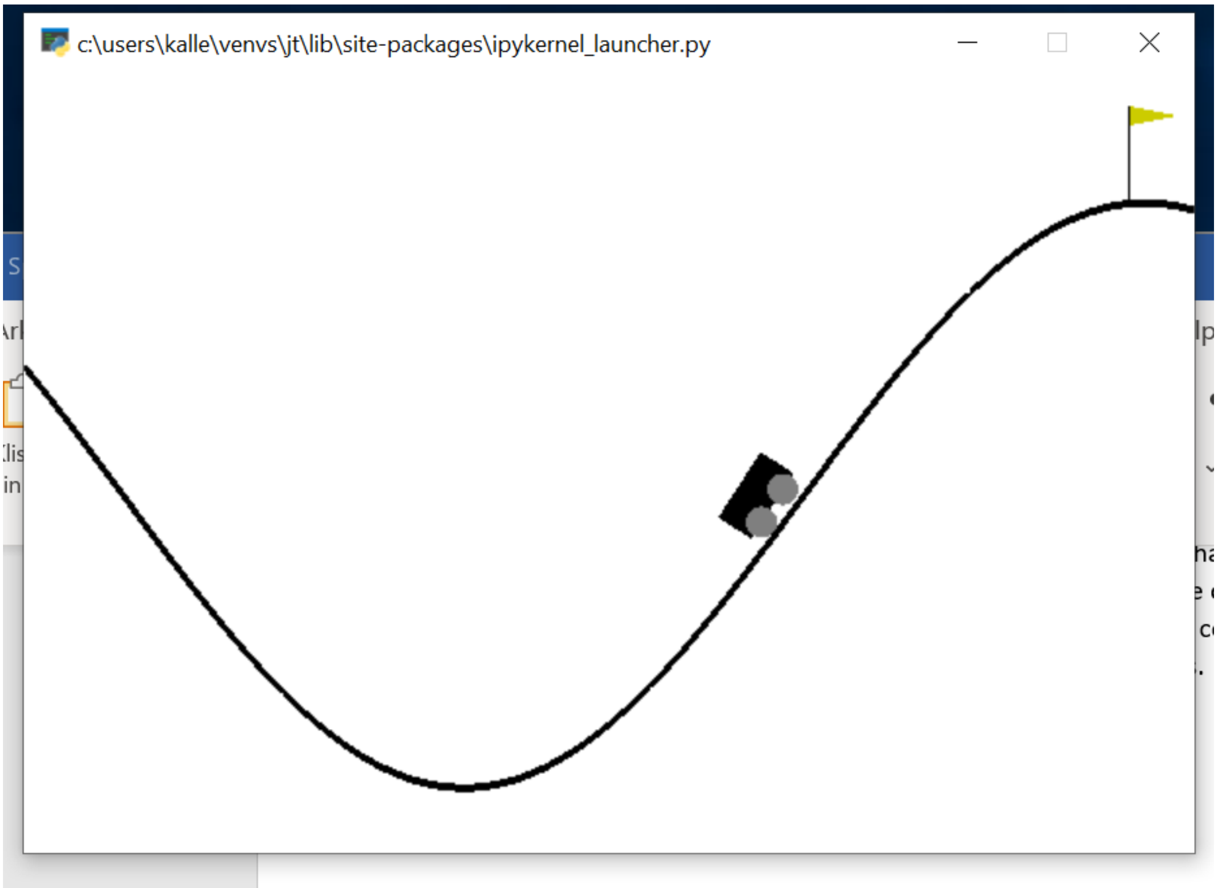
c:\users\kalle\venvs\jt\lib\site-packages\ipykernel_launcher.py

Fig 1. Xming window output.

```
Episode 4900 Average Reward: -181.06 Epsilon: 0.01599999999992594

Episode 5000 Average Reward: -194.89 Epsilon: -7.408966018820906e-14
```

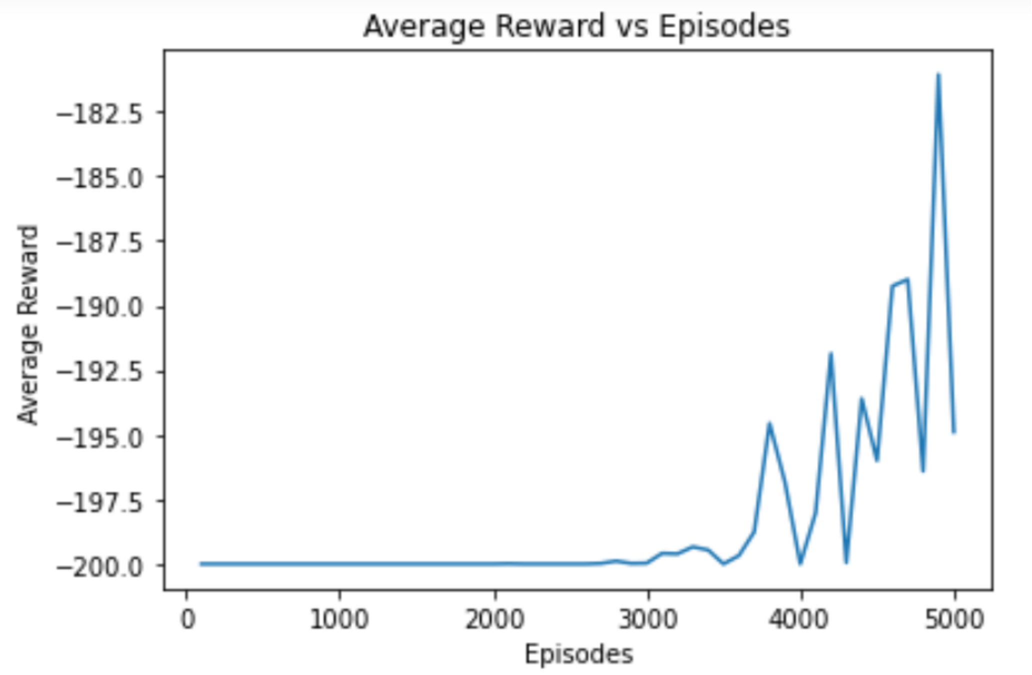

Fig 2. The result shown at end of run

Programming Environment

We have used Jupyter Notebook but you are free to use other Python environments. There are install instruction including for the graphical environment Xming on a Windows 10 computer separately. It is nice to have the graphics but not absolutely necessary.

Your task is to modify the code in some ways;

1. Study changes of the Hyperparameters Learning rate and Epsilon; what happens if you double the values or half the values, describe the behaviour and results. What is the 0.00016 in the example output?

2. Modify the code so it works with the CartPole environment (see video RL Intro 4 and Sutton-Barto book in the reference part). You need to move a comment to change the Environment and increase the state-part of Q from 2 dimensions to 4 dimensions and modify some code and hyper-parameters. (Standard Grid Q-learning is ok but if you want to be advanced and learn more you can try a Deep Reinforcement Learning solution like DQN but prepare yourself for spending much time then)

Report

Submit a short report in pdf together with your Python source files.

Good luck!

References:

Google Colaboratory

https://colab.research.google.com/notebooks/intro.ipynb#

Getting Started with Reinforcement Learning and Open AI Gym

https://towardsdatascience.com/getting-started-with-reinforcement-learning-and-open-ai-gym-c289aca874f

Open AI gym Mountain car

http://gym.openai.com/envs/MountainCar-v0/

https://github.com/openai/gym/blob/master/gym/envs/classic_control/mountain_car.py

Example 10.1: Mountain Car Task in Sutton-Barto book at page 244 (or 266 in the trimmed pdf)

http://incompleteideas.net/book/the-book.html

http://incompleteideas.net/book/RLbook2020trimmed.pdf

Cart-pole problem

Example 3.4 in the same Sutton-Barto book; (book-page 56 or 78 in the pdf)

https://gym.openai.com/envs/CartPole-v0/

https://github.com/openai/gym/wiki/CartPole-v0