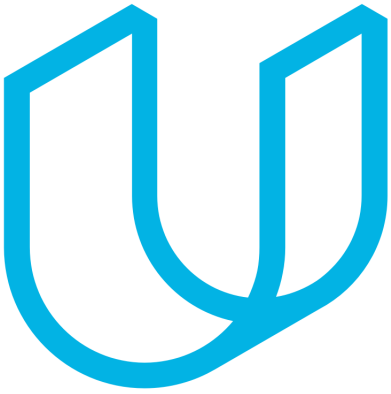
Navigation Project
Eat the yellow bananas (and avoid the blue ones)

Jean-Baptiste Gheeraert



UDACITY

DEEP REINFORCEMENT LEARNING NANODEGREE
UDACITY

July 16, 2019

Table des matières

Chapitre 1

Project description

1.1 Environment

In this project an agent is trained to navigate and collect bananas. The state space has 37 dimensions (containing the agent's velocity, along with ray-based perception of objects around the agent's forward direction). The action space has 4 dimensions : move forward, move backward, turn right and turn left. The environment is considered solved when an average score of +13 over 100 consecutive episodes is obtained.

1.2 Learning algorithm

The algorithm used to solve this environment is a Deep Q-Network [3]. Briefly the Deep Q-Learning algorithm consists in a reinforcement routine where the action value is estimated using a Deep Neural Network.
In order to train the Network we would like to compare the estimated action value with the true action value but it is unknown. Hence we compare the estimated action value with the sum of the current reward and the future estimated maximum action value.
In a mathematical form we get :

$$J(w) = \mathbb{E}\left[\left(q_\pi(S,A) - \hat{q}(S,A)\right)^2\right] \tag{1.1}$$

If we derive this equation, using a learning rate of value $\alpha\frac{1}{2}$ and replacing the unknown q_π by its estimate $R + \gamma \max_a \hat{q}(S', a, \boldsymbol{w}^-)$ we get :

$$\Delta \boldsymbol{w} = \alpha\left(R + \gamma \max_a \hat{q}\left(S', a, \boldsymbol{w}^-\right) - \hat{q}(S, A, \boldsymbol{w})\right)\nabla_{\boldsymbol{w}}\hat{q}(S, A, \boldsymbol{w}) \tag{1.2}$$

The w^- is a set of parameters for a parallel DQN that are kept fixed during the backpropagation. This is for mathematical reason (if it was not fixed, the derivative would be different).

The first structure tried was the one from the first DQN task of the nanodegree (with few modifications) and it gave very good results. I did not have to change the structure of the hyperparameters in order to obtain these results.

The neural network is composed of 3 FC layers with Relu activation after the first and second layer.
- First layer : input size = 37 and output size = 64
- Second layer : input size = 64 and output size = 64
- Third layer : input size = 64 and output size = 4

The training hyperparameters are as follow :
- Buffer size : 100,000
- Batch size : 64
- γ : 0.99
- τ : 0.001
- learning rate : 0.0005
- update every : 4
- ϵ_{start} : 1 ; ϵ_{end} : 0.01 ; ϵ_{decay} : 0.995

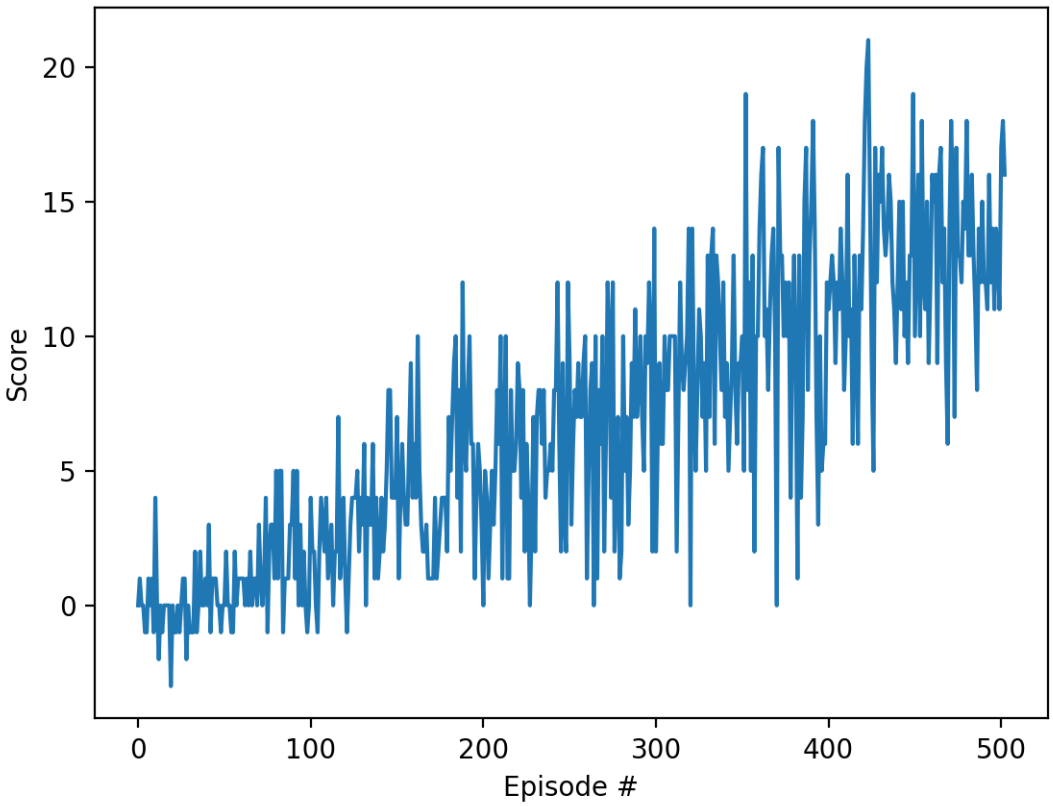
Chapitre 2

Plot of rewards

The environment has been solved in 403 episodes.

```
Episode 100     Average Score: 0.672
Episode 200     Average Score: 3.82
Episode 300     Average Score: 6.17
Episode 400     Average Score: 9.48
Episode 500     Average Score: 12.85
Episode 503     Average Score: 13.00
Environment solved in 403 episodes!     Average Score: 13.00
```

Here is the graph of the score evolution :



Chapitre 3

Ideas of future works

To improve the convergency speed, the evolution mentioned in the course may be used. The Double DQN [1] may help to reduce the overestimates of the action values ; the dueling DQN [5] could be useful in order to better generalize accross action space and proritized experience replay [4] could speed convergence using a more intelligent experience sampling.
A Rainbow DQN [2] that mix all the recent ameliorations of the original DQN could also be used.

Bibliographie

[1] Hado van Hasselt, Arthur Guez et David Silver. "Deep Reinforcement Learning with Double Q-learning". In : *AAAI*. 2015.

[2] Matteo Hessel et al. "Rainbow : Combining Improvements in Deep Reinforcement Learning". In : *AAAI*. 2017.

[3] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In : *Nature* 518 (fév. 2015), p. 529-33. doi : 10.1038/nature14236.

[4] Tom Schaul et al. "Prioritized Experience Replay". In : *CoRR* abs/1511.05952 (2016).

[5] Ziyu Wang et al. "Dueling Network Architectures for Deep Reinforcement Learning". In : *ICML*. 2016.