Neural Networks Homework 3: Deep Reinforcement Learning

Manuel Guatto; email: manuel.guatto@studenti.unipd.it
Mat: 2022574

February 11, 2022

1 Introduction to the Homework

In this homework we have to implement and test neural network models for solving reinforcement learning problems. The point where we are starting to implement the homework is the code of the Lab 07. The first task requires to study the exploration profile and change the rewards to speed up the convergence of our model. In the second task we will solve another reinforcement learning problem. In this case we have chosen the MountainCar-v0.

2 Cart Pole-v1 Environment

2.1 Introduction to the problem

In this environment the main goal is to find a way of stabilizing a pole that is attached to a cart that could move along a frictionless track. The state space is composed by 4 features:

1. Cart Position (Min: -4.8, Max: 4.8)
2. Cart Velocity (Min: -Inf, Max: + Inf)
3. Pole Angle (Min: -0.418rad, Max: 0.418rad)
4. Pole Angular Velocity (Min: -Inf, Max: + Inf)

The action space instead is a 2 dimensional space:

1. Push Cart to the left (Value = 0)
2. Push Cart to the right (Value = 1)

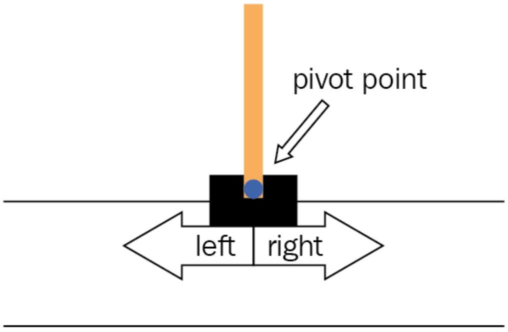


Figure 1: Image of the system

The standard reward for every step in the episode is 1 until the termination. An episode can terminate if one of the following condition becomes true:

- Pole angle $> 15\,\mathrm{deg}$
- Cart moves more than 2.4 units from the center
- We have reached 500 steps without encountering one of the previous conditions (in this case we have won the episode)

To deal with this problem we will use the network of the Lab07:

Network	Layer	In features	Out features	Activation Function
Homework 3	Linear	State Space dimension	128	Tanh
	Linear	128	128	Tanh
	Linear	128	Action Space dimension	/

Table 1: Neural Network Homework 3 Architecture

The parameters used for this task are:

- $\gamma = 0.97$
- replay memory capacity = 10000
- lr = 1e-2
- Number of episodes to wait before updating the target network = 10
- Batch size = 128
- Penalty to the reward when we are in a bad state = 0
- Minimum samples in the replay memory to enable the training = 1000
- Initial value for exponential decay = 5

2.2 Exploration Profile

First of all we consider two different exploration profiles, the first one is create to allow more exploration, instead the second one is the one used in the Lab07. The exploration profile is very important in the reinforcement learning landscape, since thanks to this we can deal with the exploration/exploitation dilemma. In particular the exploration profile we are taking in account is created to be used together with the softmax policy. In particular it works with the softmax temperature that present the following behaviour:

- if $\tau \longrightarrow \infty$ then we have a random choice of the actions
- if $\tau = 0$ then we have a greedy policy.

The exploration profile we are using in this homework is:

$$\tau = \tau_{initial} * expdecay^{i} \text{ for i = 1,2,...., num iterations}$$

So as the iteration goes on the uncertainty on the action to take will decrease. In our case we has chosen two exponential decay factors:

1. $expdecay = \frac{-log(initialvalue)}{(numiteration*2)}$
2. $expdecay = \frac{-log(initialvalue)}{(numiteration*6)}$

In the following image we can see the two exploration profiles and the scores comparison.

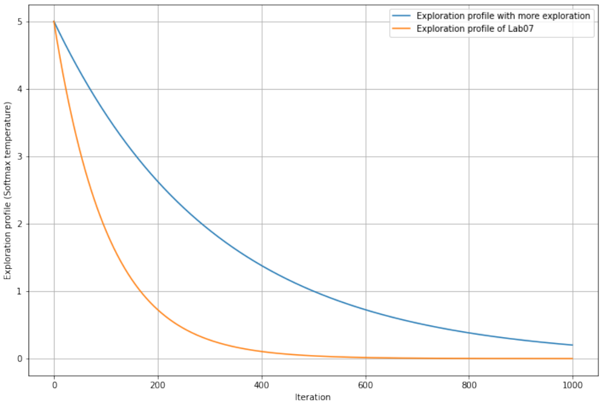


Figure 2: Exploration profiles

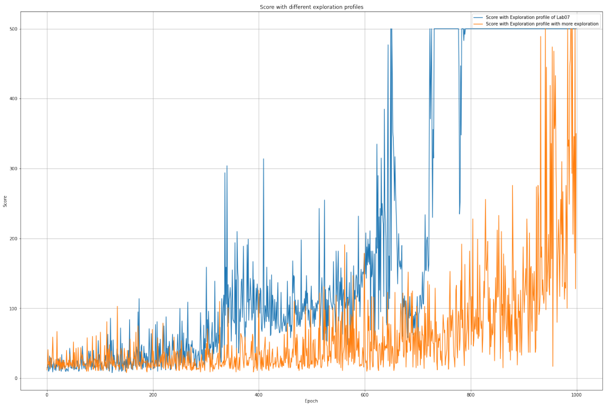


Figure 3: Exploration profiles score

As we can see the exploration profile used in the Lab07(orange in Figure 2 and blue in Figure 3) that has a faster decay of exploration, presents a faster increasing of the score w.r.t. the other exploration profile. This is not a general result since maybe in more complex problems the exploration can improve the knowledge and increase the long term rewards.

2.3 Change the rewards to speed up the learning convergence

Instead of trying to tune the hyperparameters that as said in the lab could be complicated, we try to change the way with which we compute the rewards. In particular we will test three different types of rewards:

1. Normed angle
   - With this reward we try to penalize not just the position of the cart but also the distance of the angle from the equilibrium position:
     $reward = reward - w_{position} * position - \alpha * |angle|$
2. Normalized position and angle
   - With this reward we try to penalize both the position of the cart from the center but also the distance of the angle from the equilibrium position. Both are normalized and then subtracted to 1:
     $reward = reward - (|\frac{position}{position_{maximum}}| + |\frac{angle}{angle_{maximum}}|)$
3. Angle displacement
   - With this reward we try to penalize the displacement between the angle at timestamp t and the one at timestamp t+1, if the difference is negative this means that the pole is returing near the equilibrium point and so we give a positive reward, in the opposite case we will give a negative reward.
     $reward = -\alpha * (|angle_{next}| - |angle|)$

The result of the convergence is available in the following image.

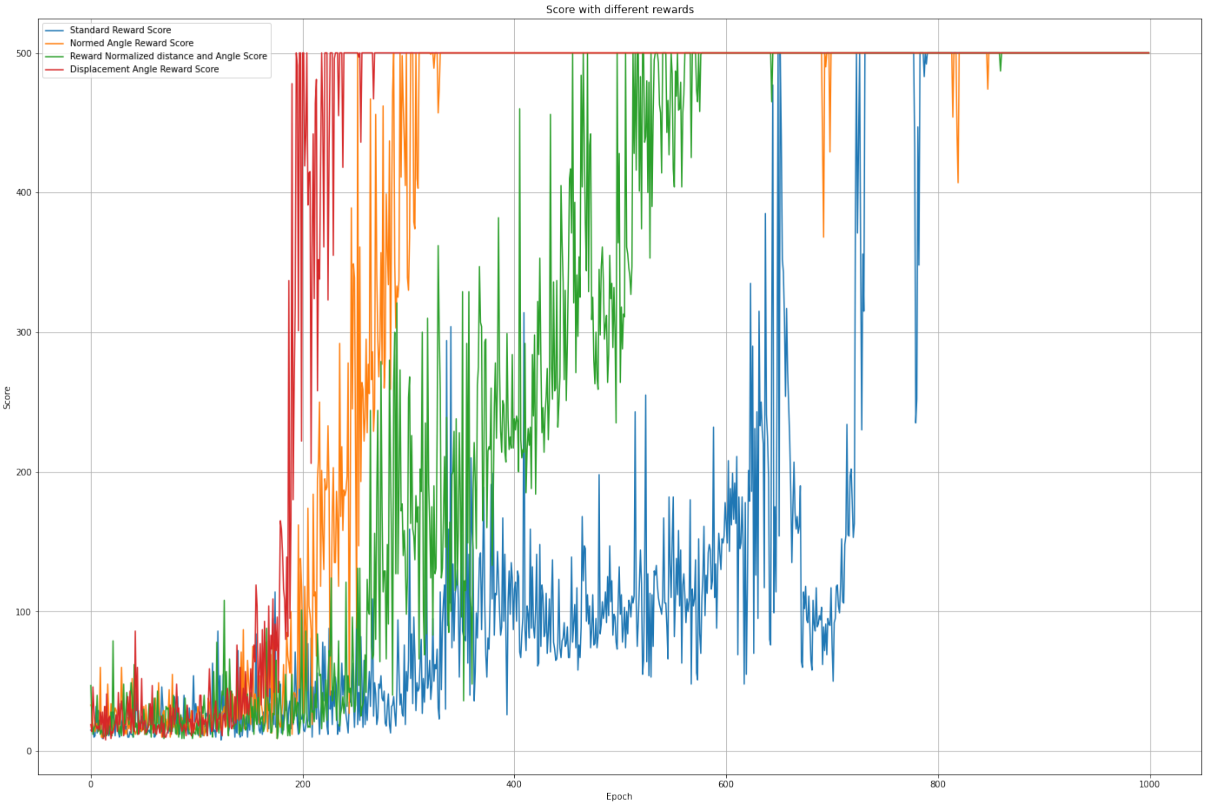
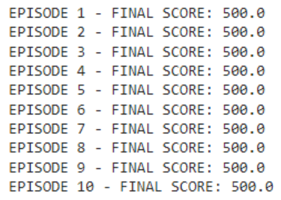


Figure 4: Comparison between the convergence of different rewards

As we can see from the above image, changing the rewards we have speed up the learning convergence w.r.t. the lab. Moreover we can see that the reward that has the faster convergence is the 3rd (Angle displacement):

```
reward = -alpha*(abs(next_state[2])-abs(state[2]))
```

Testing at the end our model we obtained for all the test 500 that is the maximum score that we can reach trying to control this type of environment (Figure 5).



```
EPISODE 1 - FINAL SCORE: 500.0
EPISODE 2 - FINAL SCORE: 500.0
EPISODE 3 - FINAL SCORE: 500.0
EPISODE 4 - FINAL SCORE: 500.0
EPISODE 5 - FINAL SCORE: 500.0
EPISODE 6 - FINAL SCORE: 500.0
EPISODE 7 - FINAL SCORE: 500.0
EPISODE 8 - FINAL SCORE: 500.0
EPISODE 9 - FINAL SCORE: 500.0
EPISODE 10 - FINAL SCORE: 500.0
```

Figure 5: Test CartPole results

3 Mountain Car v-0 Environment

3.1 Introduction to the problem

In this environment the goal is to drive up the mountain on the right; the main problem is that the car's engine is not strong enough to scale the mountain in a single pass. The state space of the environment is composed by 2 features:

1. Cart Position (Min: -1.2, Max: 0.6)
2. Cart Velocity (Min: -0.07, Max: +0.07)

The action space instead is a 3 dimensional space:

1. Push Car to the left (Value = 0)
2. No Push (Value = 1)
3. Push Car to the right (Value = 2)

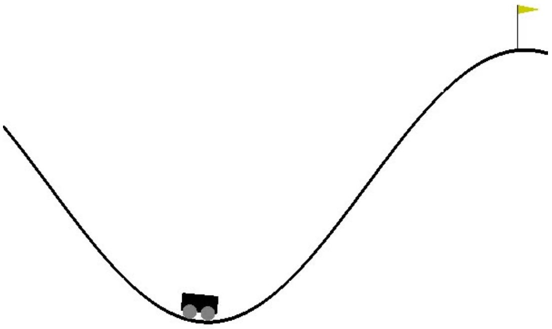


Figure 6: Image of the Mountain car system

The standard reward for every step in the episode is -1 until the termination. An episode can terminate if one of the following conditions occur:

- We reach 200 steps in the episode without reaching 0.5
- We reach the position 0.5

To deal with this problem we will use the network of the Lab07:

Network	Layer	In features	Out features	Activation Function
Homework 3	Linear	State Space dimension	128	Tanh
	Linear	128	128	Tanh
	Linear	128	Action Space dimension	/

Table 2: Neural Network Homework 3 Architecture

The parameters used for this task are:

- $\gamma = 0.97$
- replay memory capacity = 10000

- lr = 1e-2
- Number of episodes to wait before updating the target network = 10
- Batch size = 128
- Penalty to the reward when we are in a bad state = 0
- Minimum samples in the replay memory to enable the training = 1000
- Initial value for exponential decay = 5

Instead of using the score as -1 for every timestamp, to be more clear in plotting the results, we use the position of the car to determine the score. In this way we can see the improvement of the network, since in the training we have a temperature value different from zero, therefore the car does not reach 0.5 but a position near the best point. Due to this fact if we were using the original score, in the training we will see always -200, instead with this trick we can see the improvement in the position. Differently when we are in the test phase and therefore we act in a greedy way (temperature = 0) we reach every time the 0.5 position.

3.2 Training the agent

Up to now we have initialized both the neural network, the replay memory and all the hyperparameters. Now we want to train the agent. The number of epochs chosen is 500. First of all we set the exploration profile. We have decided to use the same as before, so with the exponential decay:

$$expdecay = \frac{-log(initialvalue)}{(numiteration*6)}$$

We have used this exponential decay to regulate the temperature of the softmax policy. Looking to the rewards we have chosen to give an higher reward to the actions that increase the momentum and to penalize the actions that act in contraposition of the improving of the mechanical energy. Moreover if the cart reach a good position the agent will get a bigger reward.

```
    if (action ==0 and state[1]<0) or (action==2 and state[1]>0):
        reward= reward + 6 * np.abs(state[0]+0.5)
    elif (next_state[0]<state[0] and action == 0 and state[0] > 0) or (next_state[0]>state[0] and action == 2
         and state[0] < 0):
        reward= reward + 3 * np.abs(state[0]+0.5)
    else:
        reward -= 2
    if(state[0]>0.485):
        reward += 100
```

Following the exploration profile and the reward reasoning we have obtained the below results.

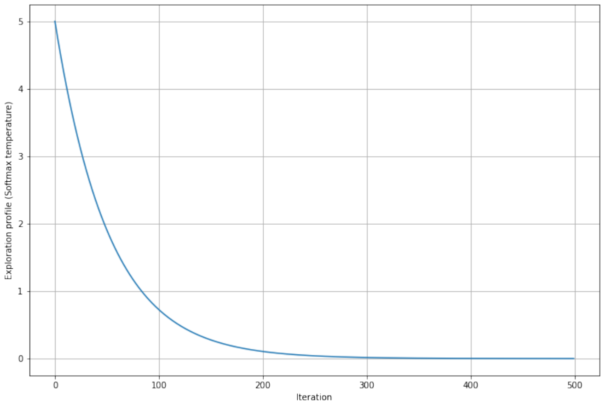


Figure 7: Exploration profile

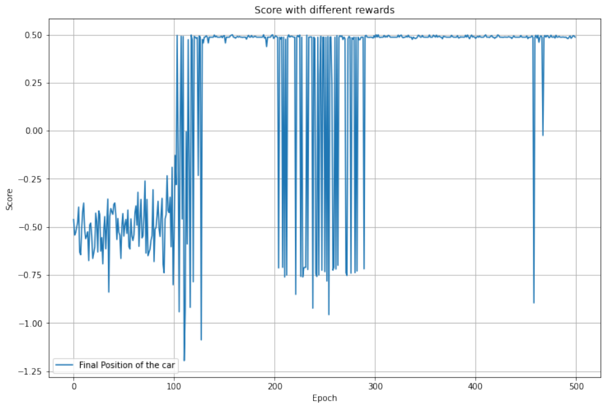
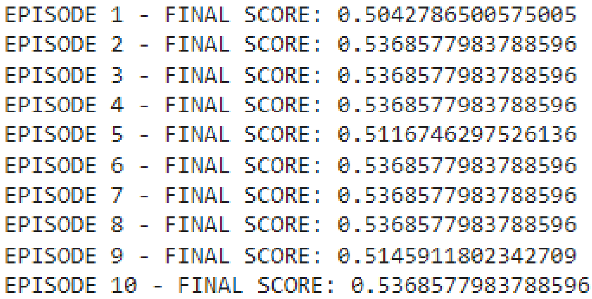


Figure 8: Mountain Car Environment score

3.3 Testing the agent

At the end after having tested the agent for 10 episodes we can conclude that we have "beaten" the environment since in every test we have reached the 0.5 position. In particular the results of the tests are available in the following image.



```
EPISODE 1 - FINAL SCORE: 0.5042786500575005
EPISODE 2 - FINAL SCORE: 0.5368577983788596
EPISODE 3 - FINAL SCORE: 0.5368577983788596
EPISODE 4 - FINAL SCORE: 0.5368577983788596
EPISODE 5 - FINAL SCORE: 0.5116746297526136
EPISODE 6 - FINAL SCORE: 0.5368577983788596
EPISODE 7 - FINAL SCORE: 0.5368577983788596
EPISODE 8 - FINAL SCORE: 0.5368577983788596
EPISODE 9 - FINAL SCORE: 0.5145911802342709
EPISODE 10 - FINAL SCORE: 0.5368577983788596
```

Figure 9: Results of testing the agent on the Mountain environment