Report 21/04 : Reinforcement learning for Cache-Friendly Recommendations

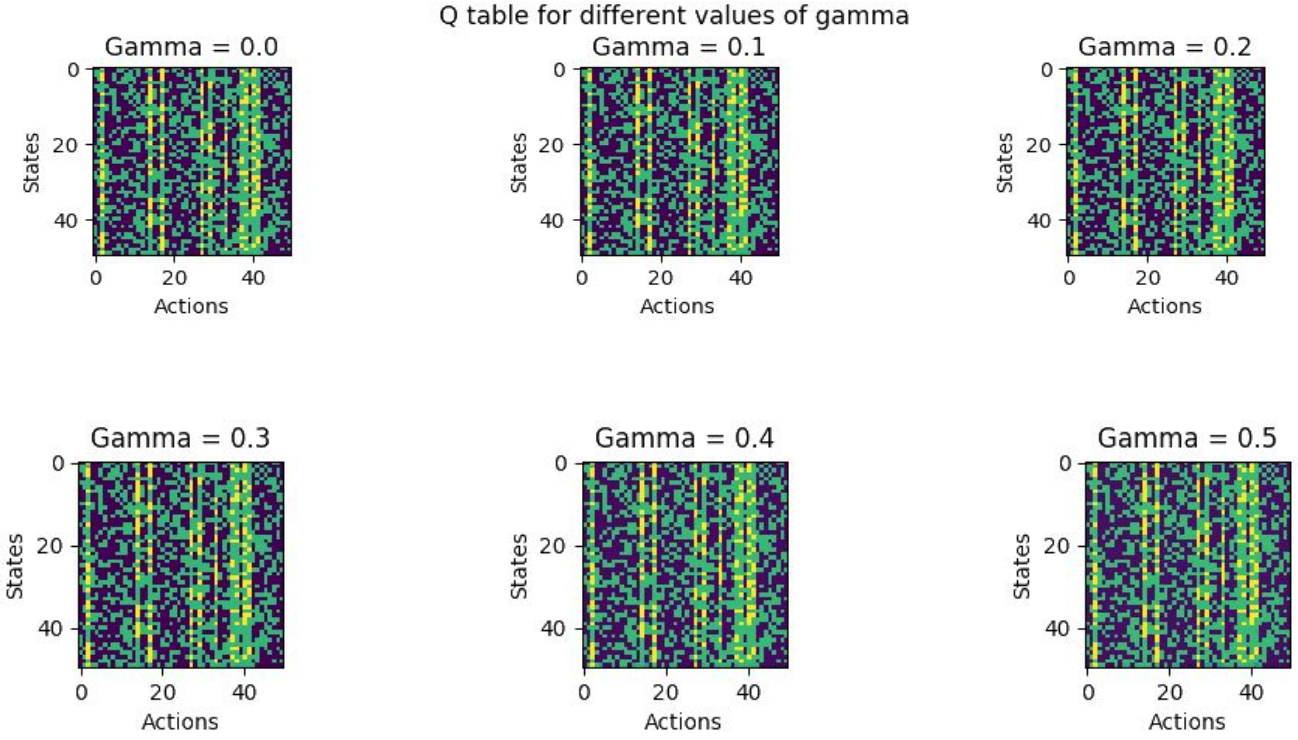
Jade Bonnet and Clément Bernard

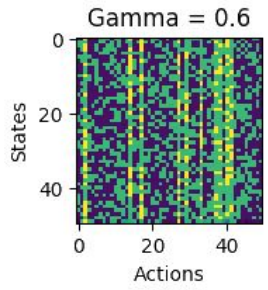
1) Try to change the value of gamma

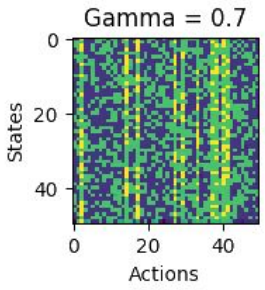
Here is our result when we implement 100 000 epochs for different values of gamma.
The initialisation values are :

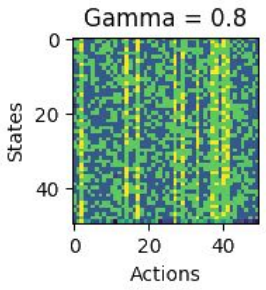
```
env = Environment(n_actions=50,n_states=50,alpha=0.6, to_leave=0.1, n_recommended=20,\
            n_cached=10,rewards=[10,5,5,-5],SEED=77)

q_table, all_penalties, all_rewards, all_q_table = q_learning(env,alpha = 0.2,gamma = gamma ,\
                           epsilon = 0.1,max_iter = max_iter_g)
```

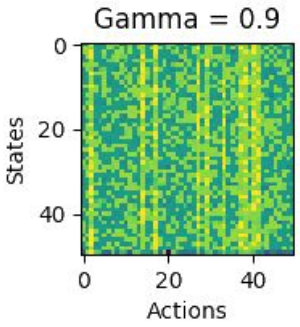
Initial values for the environment











Q_tables for different gamma values for 100 000 epochs with 50 spaces and 50 actions

(Previous mistake from the last report : we re-initialized the environment for each value of gamma. We removed this mistake here).

What we don't see is that, for instance with gamma = 0.9, it will converge but with a lot more epochs. The final values are also different.

We can predict the final values of the q_tables. Indeed, we have the following relation :

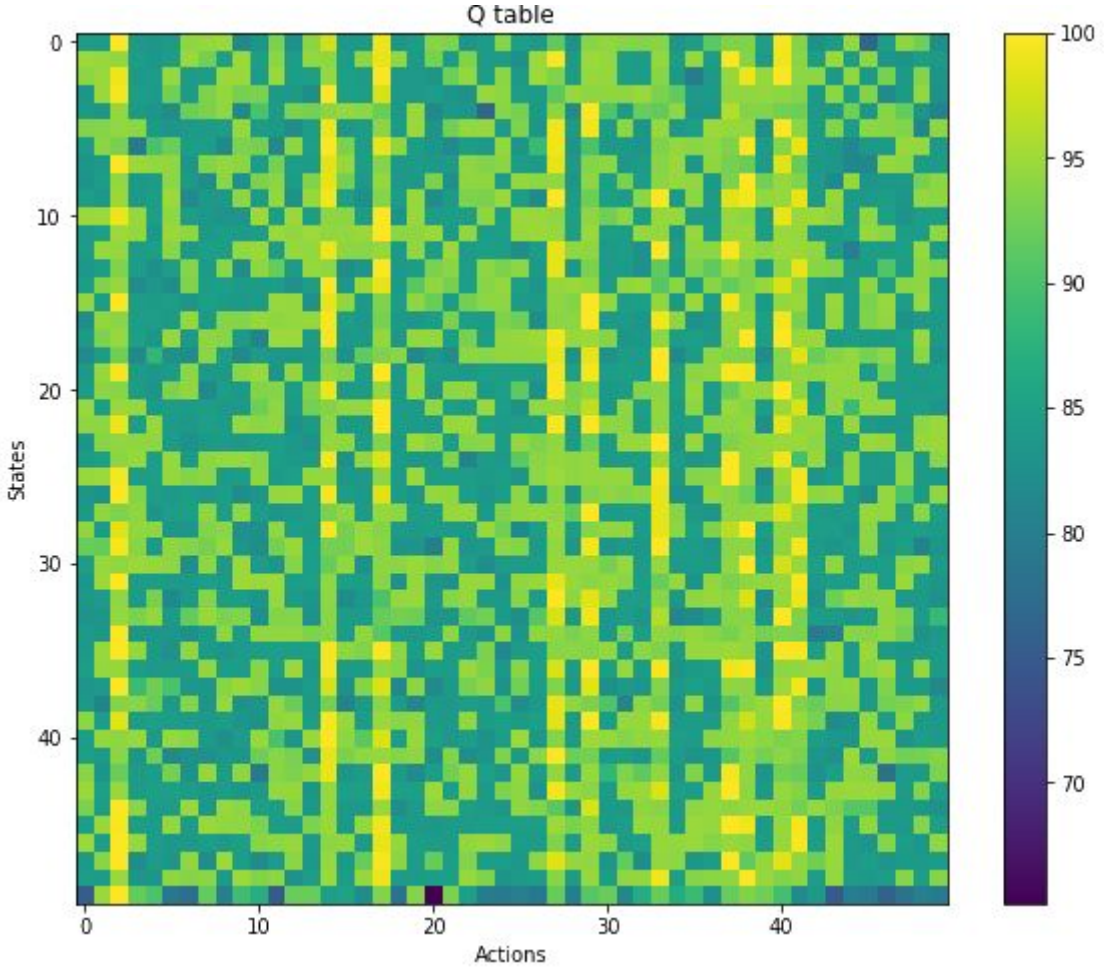
$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1},$$

Therefore, we have :

$$G_t \leq Max(R_t) \times \frac{1}{1-\gamma}$$

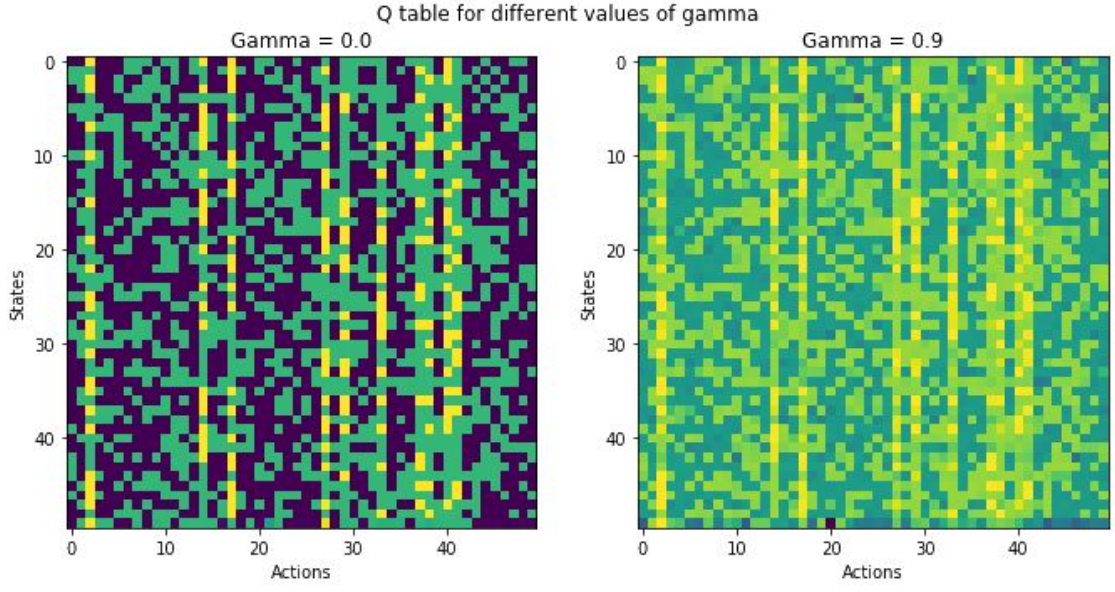
For $\gamma = 0.9$, we have : $G_t \leq \frac{10}{1-0.9} = 100$

It explains the time to converge to this value. We can see this scale with this plot :



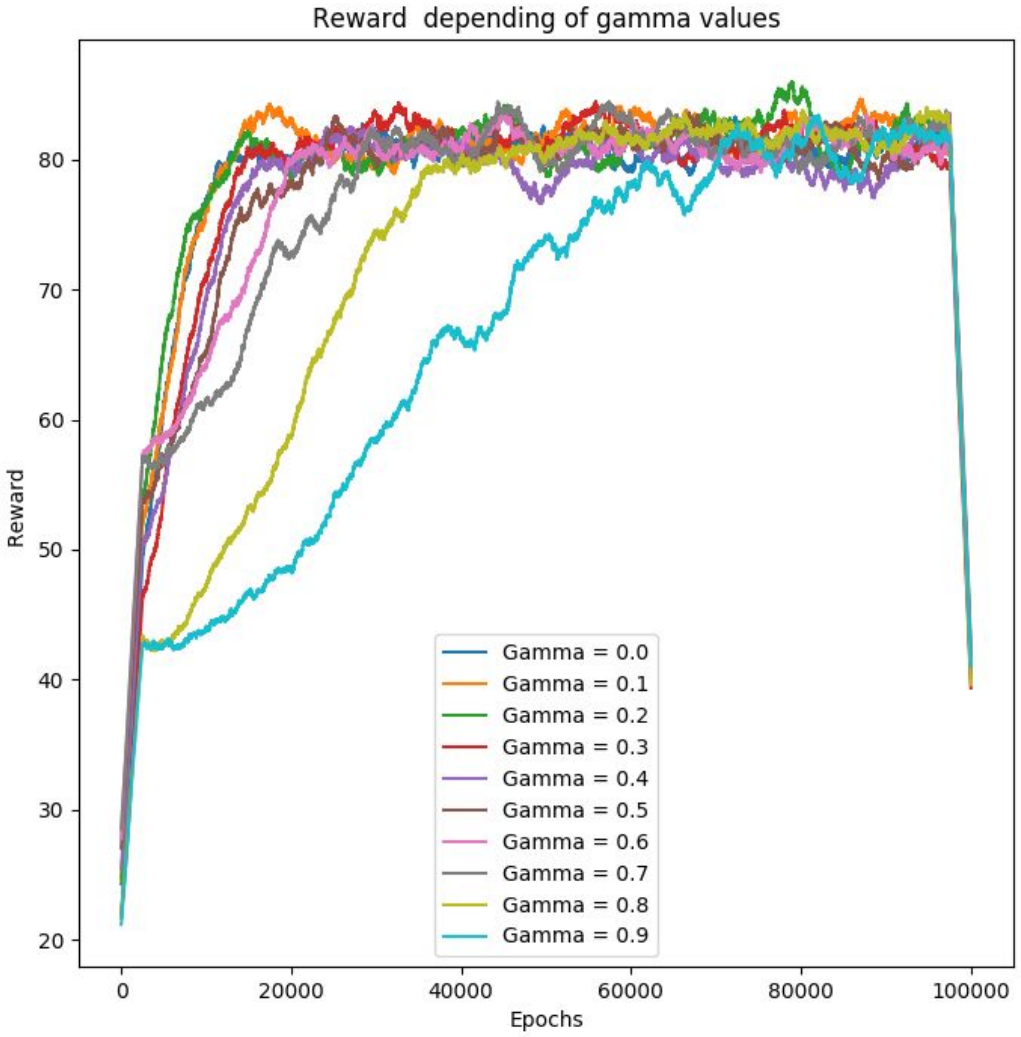
Q_table after 100 000 epochs for gamma = 0.9

We can therefore compare the q_table for gamma = 0 and gamma = 0.9.



Q_table for gamma 0 and 0.9

The difference is that for gamma = 0, it will only take the actions that lead to an immediate reward whereas for gamma = 0.9, it will take into account the future rewards. In this effect the rewards will make more time to converge, as we can see in this plot :



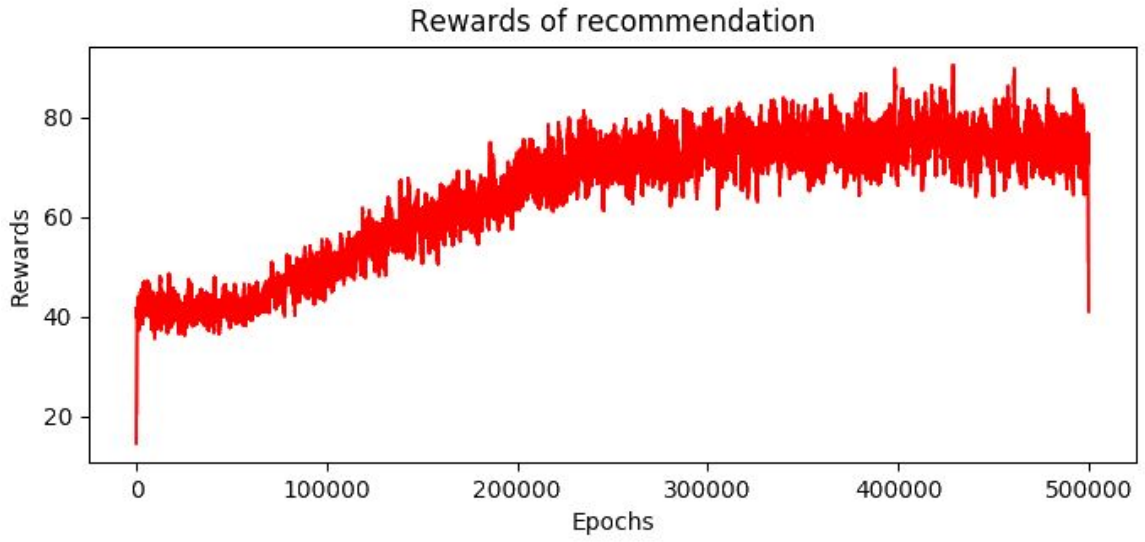
Averaged rewards for 100 000 epochs with different gamma values

We can see with this graph that the rewards converge slowly for high values of gamma.

2) Try to see why the penalties and rewards converge around 200 epochs

Rewards

We didn't compute the rewards well (we added the reward in the list that memorizes all the rewards only when there was a reward of -5 which explains why it had the same appearance as the penalty).
Here is the behavior of rewards :

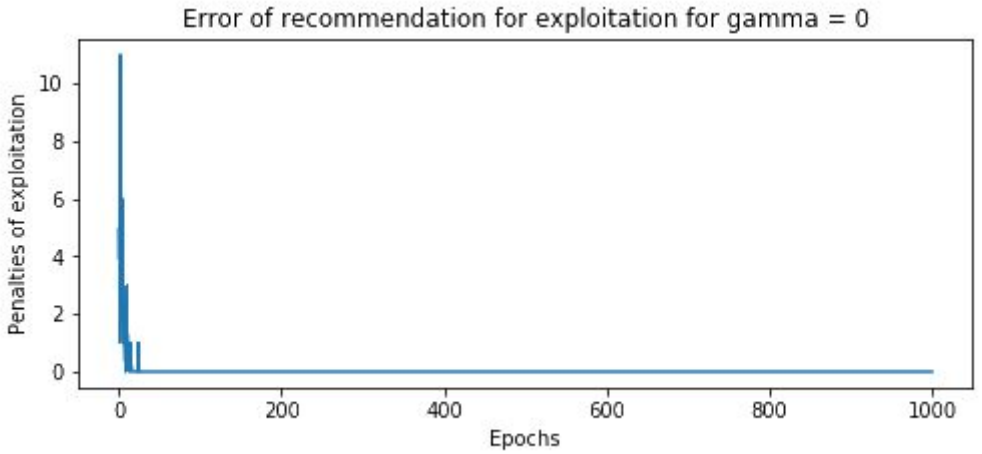


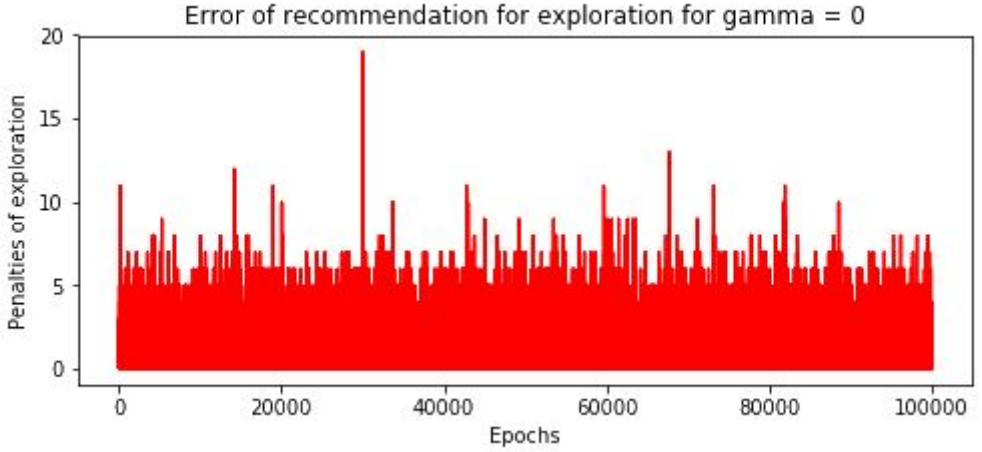
Reward through the epoch (gamma = 0.9)

We can see that the rewards increase through the epochs until the q_table has converged to its final values.

Penalties

We define a penalty as +1 whenever the algorithm recommended a content that has a reward of -5.
What we didn't see with our initial plot of the penalties is that it didn't differentiate the behavior of exploitation and exploration.





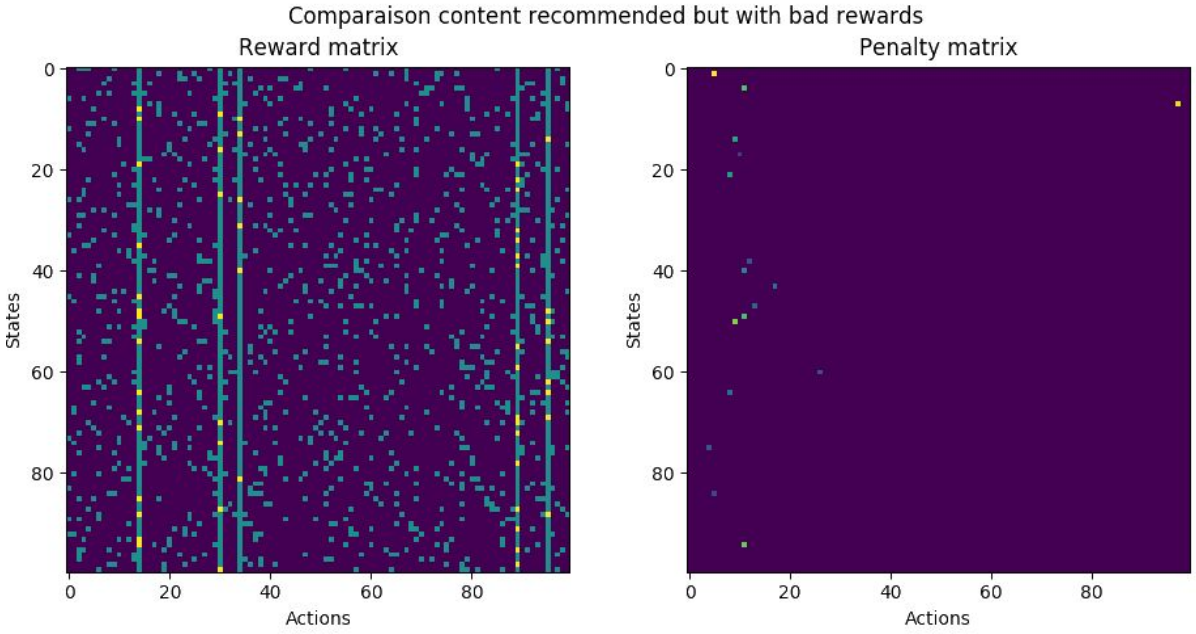
Penalties for exploitation and exploration (gamma = 0)

We can therefore see with these plots that when the algorithm is exploring, it recommends a lot of bad content (with a reward of -5).

Then, when the algorithm exploits (and take the maximum of the q_table), it starts to recommend bad contents and very quickly the q_table change the value for each state which leads to recommend content that isn't of reward -5. This is more the case here because we made gamma = 0. It leads to see only the immediate rewards, which makes sense to converge very quickly.

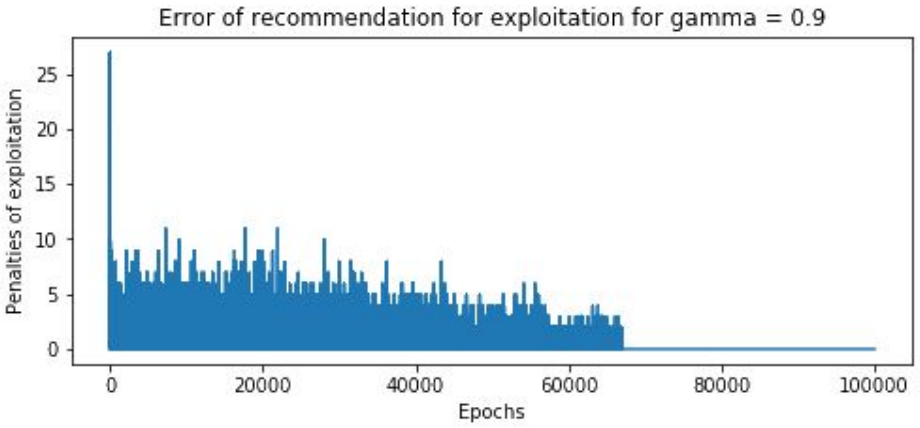
3) **Try to see if actions with currently bad rewards can be recommended if it will lead to good futur rewards**

We did the q_learning algorithm and we kept in memory whenever the algorithm recommends a content with an immediate reward of -5 (not by exploring, just by exploiting). We then plotted this with the reward matrix :



Comparaison reward matrix with matrix of current bad recommendations (100 000 epochs, gamma = 0.9)

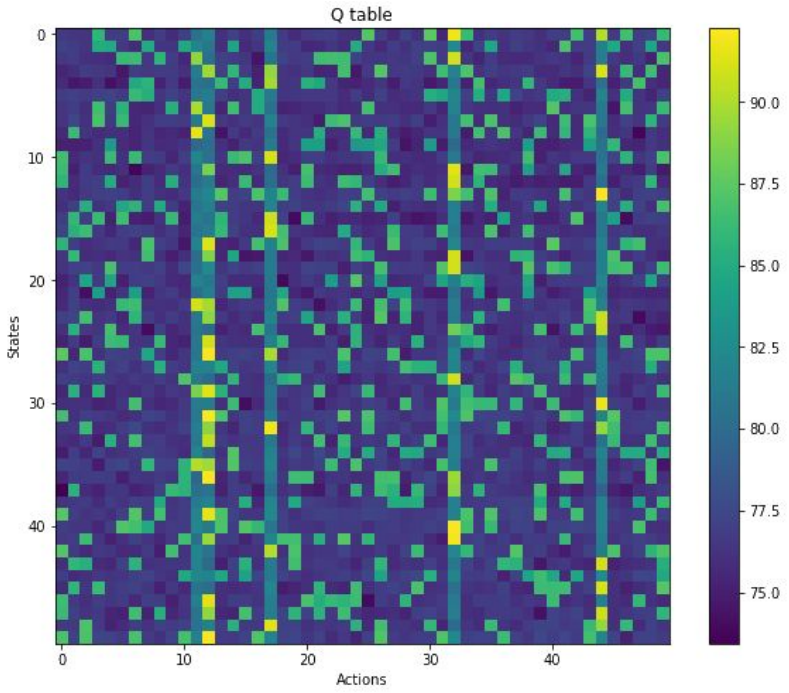
As gamma = 0.9, there are some actions that are recommended but have immediate reward of -5. We can see there on the plot of the right.
We then computed the exploited penalty with gamma = 0.9. Contrary to the case where gamma = 0 (the penalty converge before 100 epochs), we can see that as the algorithm takes into account the future benefits, current actions with bad rewards can be recommended. Here is the plot corresponding :



Exploited penalties for gamma = 0.99

4) Try to change the values of rewards

We tried to use the following rewards : [10,5,0,-5].
Our result is the following :



Q_table for reward of 10,5,0,-5

We can conclude that using different rewards leads to have more spreaded values in the q_table.