Lab Report 2 Submission (2019)

Sylvain Lapeyrade, Reda Bourakkadi (sylla801, redbo196)

December 13, 2019

Answers

1) Exercise 1: Trade-off Between Exploration and Exploitation

Evaluate the effects of α, γ and ϵ, and plot your accumulated reward for your best set of values. To get a good result you will need to update ϵ from a large to a small value during training. Study the values of the Q table for your best solution. What are the major difficulties for learning in this environment?

By trying different values for α, γ and ϵ, we found the best results with a value of 0.9 for each. As suggested, with have decreased ϵ from the initial 0.9 by 0.001 after each episode, until 0.

While we manage to have total reward of more than 0.5 constantly as instructed, our accumulated reward for our best set of value is 0.6661 and the corresponding best values are as in Table 1:

0	3	3	3	0	0	2	0	3	1	0	0	0	2	1	0

Table 1: Best values of the Q-tables of our best cumulated reward.

They are corresponding to the best action as evaluated by the algorithm. *Left* is designated by 0, *Down* by 1, *Right* by 2 and *Up* by 3. The Q-Table values are on in Figure 1. Indeed, since we took the epsilon greedy approach, where we randomly generate value between 0 and 1 and then we see if they are smaller than /*epsilon*. If so, a random action between the four mentioned earlier is chosen. Otherwise, we choose the action with the maximum value in the Q-table.

The major difficulties for learning in this environment is to find adequate values for α, γ and ϵ and also figure out how much to decrease ϵ on each episode. This is very important, because it determined how long we want to **explore** (i.e. randomly select actions to try new possibilities) which correspond to a big ϵ value, and how long to **exploit** (i.e. take the best action already found to pursue the best set of actions) which is a small ϵ value.

Left	Down	Right	Up
2.24034053e-02	2.05161925e-04	2.24713243e-04	2.15449416e-04
9.50456411e-06	4.06757021e-05	1.08813765e-05	4.65207716e-02
2.25865266e-05	1.78173882e-05	1.82306304e-05	1.63852399e-02
1.78228153e-05	6.52572124e-06	4.44221807e-06	1.76157626e-02
4.32316778e-02	2.69057491e-05	9.44212465e-05	3.66241274e-05
0.00000000e+00	0.00000000e+00	0.00000000e+00	0.00000000e+00
2.71115287e-08	8.88247889e-09	5.20025762e-02	2.37693514e-08
0.00000000e+00	0.00000000e+00	0.00000000e+00	0.00000000e+00
8.80245540e-05	6.97456017e-05	5.46894576e-05	4.40332313e-02
3.06572962e-05	7.03347894e-02	2.23675336e-05	1.03796446e-05
5.17980659e-02	1.69806541e-06	3.65062105e-06	4.20391920e-06
0.00000000e+00	0.00000000e+00	0.00000000e+00	0.00000000e+00
0.00000000e+00	0.00000000e+00	0.00000000e+00	0.00000000e+00
3.69007597e-03	8.46520530e-04	2.50595263e-01	5.88882913e-03
1.65424746e-02	9.35429600e-01	1.87956221e-02	1.74856936e-02
0.00000000e+00	0.00000000e+00	0.00000000e+00	0.00000000e+00

Table 2: The direction and their respective rewards.

2) Exercise 2: Cooperative Multi-Agent Deep Reinforcement Learning

Select and investigate the impact of two learning parameters (lr, gamma, batch-size, num-units). Here batch-size and num-units affect the training and structure of the neural network policy. Run as many experiments as possible in parallel to save time. Plot the training progress, which is stored in the learning curves directory. How do the agents perform after training? What do you think are the major challenges for learning in this environment, and how is it different from the Frozen Lake environment? For this cooperative environment it would have been possible to use a single agent to control all three blue objects, or alternatively strictly decentralized learning using standard single-agent reinforcement learning algorithms. What is (in general) problematic with such approaches?

The learning rate parameter defines how big the current training will have impact on the next episode. For example, a low *lr* will not change the result too much from every episode while a big one will impact it more drastically.
The gamma quantifies the importance given for future rewards. The bigger it is the bigger the future rewards will have impact.
The batch-size is the number of sample trained at the same time before calculating the reward. The bigger it is the faster the training will be, however the learning will be slower.
The num-units is the number of units in the multilayer perceptron (i.e. neurons). Basically, the more units there are, the better the learning will be, the computation will also get bigger and therefore more time consuming.

After a successful training the agents perform much better since they have been well rewarded for doing what is expected of them, i.e. go to the landmarks without, if possible colliding with other agents.

Comparing to the Frozen Lake environment, the main difficulties is not that much to make the agents go to landmarks but rather the whole *multi-agent* perspective with avoiding collisions as much as possible. While in the Frozen Lake environment it was really about reaching the goal by dealing only with the environment. So in the first exercise, it was all about maximizing one agent utility while in the second one, it is more about maximizing the utility of the whole set of agents.

The usual problem of using a single agent or a strictly decentralized learning is that it is generally harder to make a good reward function. Also, the learning process will usually perform poorer than with multiple agents.

3) Exercise 3: Competitive Multi-Agent Deep Reinforcement Learning

How well do the agents perform? Are they equally good? Can you see any reason/explanation why they would not be? How do you think the length of episodes and size of the hockey rink would affect learning for your choice of reward system?

After the whole training, the agents perform very well, although we can see that the agents are not equally good. One explanation of this phenomenon can be that they don't learn the same thing at the same time since they do different actions. Therefore, it is only logical that they do not perform exactly the same. The length of episodes and size of hockey could improve the training but it would take much longer to have a good learning. So at the end it is a question of balance between the time and computation power you want to invest and the perfomance you want to have.