Advanced Topics in Artificial Intelligence

T-720-ATAI-2016

EXERCISE 2 (Omniscient Q-Learning)

In this exercise you will be using reinforcement learning to solve two tasks with varying grid sizes.

Update TEAL

Pull the latest version of TEAL if you're using Git, or download it from https://github.com/ThrosturX/rl-mouse/tree/exercises. Do this before every exercise.

Reinforcement learning

Rather than manually constructing the policy yourself like last week, you are now going to make the mouse learn to find the cheese and avoid traps on her own. The task is considered *solved* if the reward is consistently higher than 390 (more than ~90% of the time). The ratio determines how well the task was solved (i.e. a ratio of 0.2 but a reward of 400 means the task is 'solved' because the mouse always finds the cheese, but the solution is quite sub-optimal because it took on average 5 times longer to get to the cheese than needed).

Make a solution that creates an `OmniscientMouseAgent` with the actions `['left','forward','right']` in the "solutions" directory (you can use `ex2_template.py` as a template). Let the agent perform 1,000 actions each training epoch.

Call `teal.py <solution-file> --grid_size <size> --seed <your-kennitala>` to run your experiments.

Quick note: You can get immediate results with CTRL+C (`KeyboardInterrupt`) if you notice that the task is solved but many evaluation epochs remain.

A. Answer the following questions. Support your answers with appropriate arguments and data.

1. Approximately how many actions does it take for the agent to solve the task

with grid size 3? (*Hint: Use 100 actions per training session instead of 1000.)*
2. Approximately how many actions does it take for the agent to solve the task with a grid size of 5?
3. The agent might be unable to solve the task within the allotted amount of sessions on an 8x8 grid, how does it compare at the end to smaller ones?
4. How many actions do you need for a 12x12 grid? *(Hint: if you want to give it enough time to 'solve' the task, use the* `--max_epochs` *parameter and set to a value around 1000.)*
5. Estimate how long it would take to solve the task on a 15x15 grid. What about 20x20?
6. What do you think would happen if we tried this on a more interesting, realistic task? (E.g. controlling a RoboCup football robot, a chess program, playing video games, or perhaps eventually controlling an intelligent, humanoid robot.)
7. Identify two fundamental problems with our current approach, and describe some high-level solutions (you don't need to go into great detail here).

Exploration and exploitation

One of the most important topics in RL is the tradeoff between exploration and exploitation. Exploitation means taking the actions that you think will lead to the greatest rewards now. With exploration you might do something different in order to learn about alternative strategies.

B. Answer the following questions. Support your answers with appropriate arguments and data.

1. What do you think would happen if an agent never explored?

2. Make a copy of your solution file, and set the exploration rate in the copy to 0 (it should be 0.25 in the original). Now compare them with `teal.py <solution-file> --compare_to <other-solution> --grid_size 8 --seed <your-kennitala>`. What do you see? Is the result as you expected? Why (not)? Do you think this result is specific to the current task or do you think this is a common result?

3. What do you think would happen if an agent always explored during its training phase? Exploration would be turned off for evaluation.

4. Set the exploration rate in the copied solution to 1. Now compare your solutions again. What do you see? Is the result as you expected? Why (not)? Do you think this result is specific to the current task or do you think this is a common result?

5. We're turning off the exploration when we are evaluating our system. Name one circumstance where this would be a good idea and one where you would want to keep on exploring and explain your examples.

Submit your solutions

Write your answers to these questions in a file named "ex2_" followed by your name ".txt". Since not a lot of programming was required, you won't need to turn in your Python solution file(s) unless you're doing something special. Submit your homework in MySchool.