Alvaro Prat Balasch

CID: 01066209, Reinforcement Learning Part 2: Coursework Part 2

The complexity of the problem has increased significantly from the previous exercise: the goal is further from the initial state and the obstacles are more convoluted. Thus, the first change made to adapt the code to this exercise was to boost the episode length to 2000 steps, however this was further augmented to 5000 steps, increasing exploration around the goal state as well as a (dynamic) break which re-initialises the episode once the goal is reached (line 72). Consequently, as more training data was available, the *batch size* was incremented to 128, allowing gradient descent to stabilise further. Note a size of 128 was selected, making cache entries faster (power of 2). Saving memory, the *replay buffer* size was downsized to 100,000 as during training under given time constraints, no more than 50,000 steps were stored.

The *reward function* is perhaps the conundrum of this exercise. It was found that continuous rewards would impede the agent from going through large obstacles. Hence, in order to relax reward gradients far from the goal state, a semi-discrete reward function was implemented (lines 145-155). This incites the agent to move right and gives no reward for moving up and down. Through observation it was noticed that in some cases the greedy policy did not reach the goal even though ϵ-greedy converged during training. This was found to be due to the agent getting stuck to the walls. A solution to this was to introduce a *wall penalisation* (line 147). Additionally, it was found that appropriate *exploration* was crucial as some maps were much harder to be captured by the Q-network. Hence, ϵ-greedy exploration with very low decay δ was initialised for the first 2500 steps (lines 82-102). Subsequently, a *dynamic* δ was used as the Q-network converges allowing smoother shifts from exploration to exploitation. Moreover, in order to help escape local minimas, ϵ is boosted when the goal is not reached in a long time (line 64). Other hyperparameters of interest (lines 38-53) are kept similar to those in the first coursework.

Randomly assigning transitions in the batch can result in slow convergence of the Q-values in the maze. For this reason, a *prioritised experience replay buffer* was devised (lines 336, 331). This allows transitions with high temporal difference errors to be trained in preference (line 341) to those which have been observed often enough for their prediction to be properly captured by the Q-network. A weighted factor α of 0.7 was set allowing some of the old transitions to be introduced and thus increasing generalisation of the Q-values over the whole maze space.

Although over-fitting for this task, the *action space* was reduced by removing "Left". This made a significant difference in the ϵ-greedy policy convergence, as the goal was reached more often in exploration, correcting the optimal trace. Additionally, the *target Q-network* update frequency was reduced w.r.t coursework 1 to every 30 steps (lines 251, 268) since the problem is more complex and requires more steps in order to stabilise, ameliorating self-bootstrapping of the Q-network.