Short summary of the exercise topic

The second assignment is about using the Deep Q-Network (DQN) algorithm, to train a model that can navigate and solve tasks within the Minigrid environment. The environment used for this assignment is "MiniGrid-DoorKey-6x6-v0" provided by OpenAI's Gymnasium. The objective is for the agent to find a key and use it to open a door, which involves making real-time decisions based on the current state of the environment. The exercise requires the creation of a neural network that can evaluate the current environment state, predict the outcomes of potential actions, and choose the optimal action to perform at each step.

How you solved the problem

The solution for the assignment can be split into different blocks. For the soft update:

- I implemented a soft update strategy for the target network, which consists of the update of the weights of the target network with those of the main network with a scaling factor τ. This ensures a smooth evolution and stable learning.

For the buffer replay:

- I implemented a buffer replay to store transitions state, action, reward, and next state.

The core of the learning algorithm involves the temporal difference approach. The learning loop involves different steps, which are:

- Sampling mini-batches randomly from the replay buffer to break correlation and ensure a diverse range of experiences
- Computing Q-Values for the current and next states
- Computing the prediction for training using the reward and the maximum Q-value of the next state
- Computing the loss using the mean squared error between the predicted Q-values and the targets.

Performance of your agent in your own evaluation (>0.9 in server)

The model performs a score of 0.964 in the server within 1000 iterations. To reach this, the size of the mini-batch was increased from 5 to 256. This change is the one that produces more performance boost.

Other adjustment like learning rate, buffer size or epsilon decay, do not produce the same boost especially if you also consider the computational time.