

Exercise 1: Deep Q-learning

In this exercise, you will implement DQN by Mnih et al.. DQN was the first algorithm that was able to play Atari games while only using images as an input, showing the potential of deep reinforcement learning. This exercise only uses the very basics of PyTorch, but if you are interested, the official **PyTorch** documentation has a nice 60min tutorial.

- 1.1 Run `pip install -r requirements.txt` to install some additional packages.
- 1.2 Instead of updating all the states all the time, the algorithm explores the state and action space, and saves the obtained data in a replay buffer. For training, data is sampled randomly from this replay buffer. Examine the function `example_replay_buffer` in `replay_buffer.py` and run it to observe a simple implementation of a replay buffer.
- 1.3 By using function approximation methods like neural networks we can learn a parameterized Q-value function that can be applied on the whole state, instead having to rely on tabular methods which become infeasible for too many states. Complete the class `Q` in `dqn.py` to define a Q-Network.
- 1.4 Now the task is to complete the implementation of the `DQN` class. Run the function `start_training` to train the DQN agent on the Cart Pole environment. If the implementation is correct, you should achieve an episode reward of 200 in the first 100 episodes.
- 1.5 (Bonus) Try DQN on other environments, e.g., Acrobot-v1 or MountainCar-v0 .

Useful Resources

- Official PyTorch documentation
- 60min PyTorch tutorial