

Exercise: Deep Q-Learning

Find the assignment here: <https://classroom.github.com/a/tfnAZTm5>.

1. Deep Q Learning

This week's exercise aims to develop an intuition about how adding deep learning to value function approximation impacts the learning process. You have been provided with a Q-Learning procedure that uses a shallow neural network as a function approximator for **LunarLander-v2** environment. Your tasks are the following:

- Complete the DQN implementation in *deep-q-learning.py* by adding a deep network as a function approximator and a replay buffer to store and sample transitions from.
- Vary the network architecture (wider, deeper) and the size of the replay buffer. Record your observations for each choice of architecture and buffer size. Please plot the training curve for your experiments with the **number of episodes** on the x-axis and the **mean reward** on the y-axis. The plots should have your choice of architecture as the title and should be stored in a new folder **plots/**
- **Optional:** The seed can drastically impact your experiment outcome, so it is a common practice in Reinforcement Learning to repeat experiments across multiple seeds and record the training curves as mean values across these seeds with a standard deviation around this value. An example of such a plot can be found here. You could also try to run your experiments across multiple seeds and record your observations in such plots. However, please note that we will not explicitly grade this task. The purpose of this is to show you how experiments are usually conducted in the RL community.

Note: The tests provided in for this exercise are only an indicator of whether the plots and answers were generated or not, and whether the Q network learned something or not. We will look into the plots and the answers to determine the quality of the submitted solutions.

Please record your answers in **answers.txt**