Report – Challenge 2

Michael Meier (k12443157)

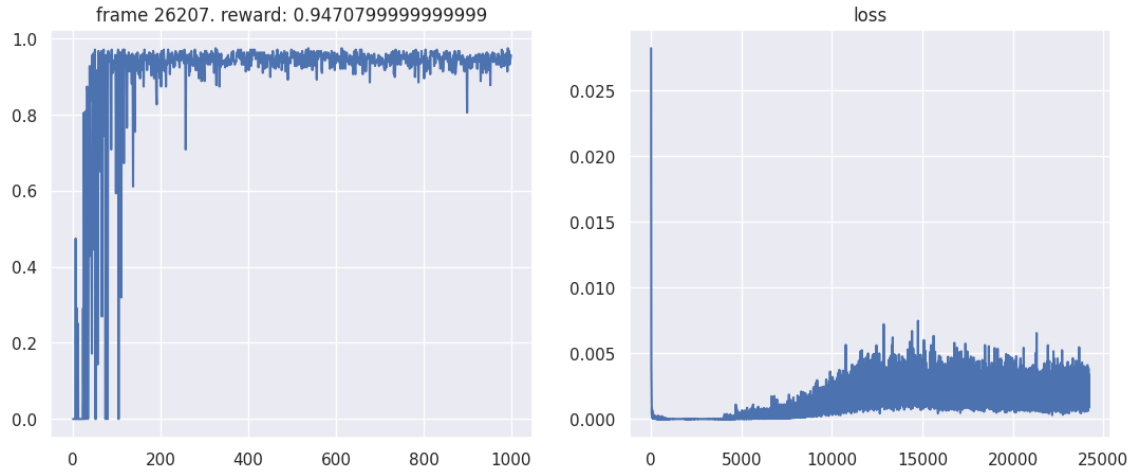
Introduction

The second task was to do Q-Learning in a Key-Door-Framework. The Agent had to maneuver to the key in a environment, pick up the key and unlock a door with it. As environment two different mini-grids were used, a empty 5x5 grid to find the right parameters and than a 6x6 grid with a key and a door for the final training of the agent. To train the agent Deep Q-Network (DQN) learning was used. This reinforcement learning technique combines Q-learning with deep neural networks to solve tasks like the key-door-task. Q-learning itself is a model-free algorithm that iteratively updates Q-values using the Bellman equation. The neural network than approximates the Q-function by taking states as an input and outputs possible actions.

Methods

In order to solve this key-door-task the provided Jupyter notebook was used. It introduced a ReplayBuffer class, which stores the transitions consisting of a state, action, reward and a next state. The model was also already provided, it consisted out of four linear layers with a ReLU activation. If no saved model was provided, a new one was initialized. For the initial training the 5x5 empty grid was used to find good working parameters. At the end the pre-trained model was tuned on the 6x6 key-door map with a learning rate of 0.0001 over 1000 episodes. The mini-batches for the training had a size of 265 and a mean squared error loss was used.

Results



In the final training the reward was close to 0.95 after a few epochs. In the evaluation the agent reached a average score of 0.965 on the local environment where it was run over 50 episodes. The training showed that the size of the mini-batch plays a significant role in the performance. When testing a trained agent with a mini-batch size of only 128 the result was significantly worse reaching only a score of 0.679.

Overall, the agent was able to learn the task within a few episodes and performed quite good without any further adjustments on the code.