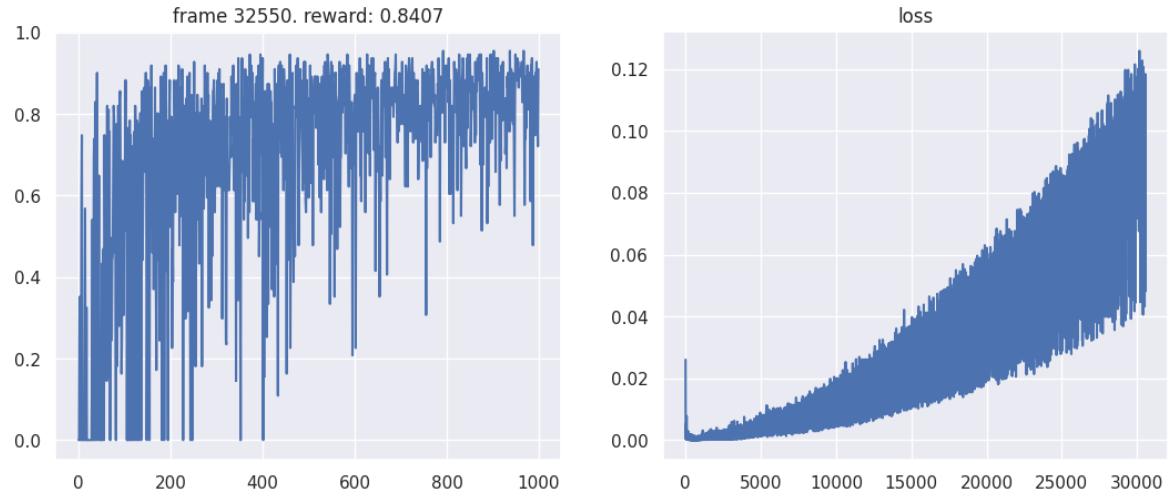
Report: Pong Minigrid DQN Assignment 2

Minigrid DQN

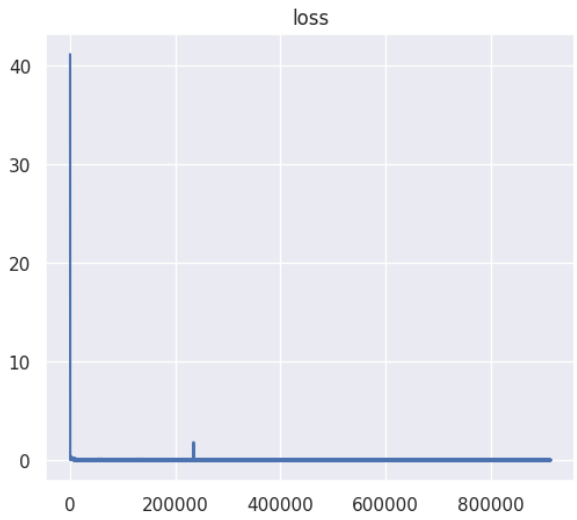
- The hyperparameters were used as they were given
- I simply followed the comment instructions to implement the DQN algorithm
- A small struggle was the use of torches as I converted only some variables first to numpy. So. of course there where device location issues. However, that was good solvable by keeping all torches as torches.
- Printing out shapes of interim results helped me to have an overview whether all operations worked as expected.
- As I implemented the mathematical code lines, I had a close look at the exercise slides. For example, I set all predicted q-values which were located at the end of a sequence to 0:
  `max_Q_batched[mb_done] = 0`
- I got following results:



Also in the visualization the agent finds the target state.

Pong DQN

- First, I implemented the network architecture. I followed the command comments.
- As I used google drive, I made use of your keep-alive strategy. It worked out.
- First, I had the problem that my network wasn‘t able to learn. I explain that with the fact that I set the learning rate to 1e-6 which is quite small.
- New hyperparameter setting:
  - number of episodes: 400
  - size of mini-batches: 28
  - learning rate: 1e-4
  - update after: 1000 steps
  - size of buffer: 1e5
  - Rest was kept as it was given
- The DQN algorithm was in the same way implemented as for the minigrid environment. It resulted in:



So, there was a huge variance in given rewards. I explain it with the fact that I used a small mini-batch size such that the algorithm‘s results were more drastically influenced by the individual samples. In addition, more training epochs could increase the given reward. However, the final model reached in the submission server a score bigger than 19. Therefore, the submission was a success.