Deep Learning Lab Course
Exercise 4

Rolf-David Bergdoll
Matr.Nr. 3338345

08.01.2019

Cartpole

After completion of the code stubs I found that the DQN agent usually gets the hang of the cartpole scenario after about 1000 episodes, using the default hyperparameters. However, there appeared to be a strong variance in the learning curves of different training runs, which I assume is due to randomness aspects of the approach. So I decided to train three models using the same parameterization for 2000 episodes each. Their episode rewards given by Tensorboard, along with the results of the greedy evaluations during training (mean of 5 runs, conducted every 50 training episodes) are shown in Figure 1.
The first and third model provided perfect test results, both producing the highest reachable score of 1002.0 for all 15 test runs. Both seemed to solve the task by quickly move a bit to the left in the beginning (regardless of the initial state) and then balance the pole in place by rapid alternation of left and right movement. The second model did not manage to learn this trick, producing a mean test score of 395.5.

Car Racing

For the car racing environment I implemented the suggested modifications. After some short test runs I used the following setup for my first (and longest) experiment:

- 2-layer CNN with 16 3×3 filters followed by a 128-unit dense layer
- Capacity of 10^5 for the replay buffer
- A non-uniform probability distribution for the actions during exploration (giving straight movement and acceleration 40% each, brake 10% and left and right 5% each)
- Frame-skipping value of 10

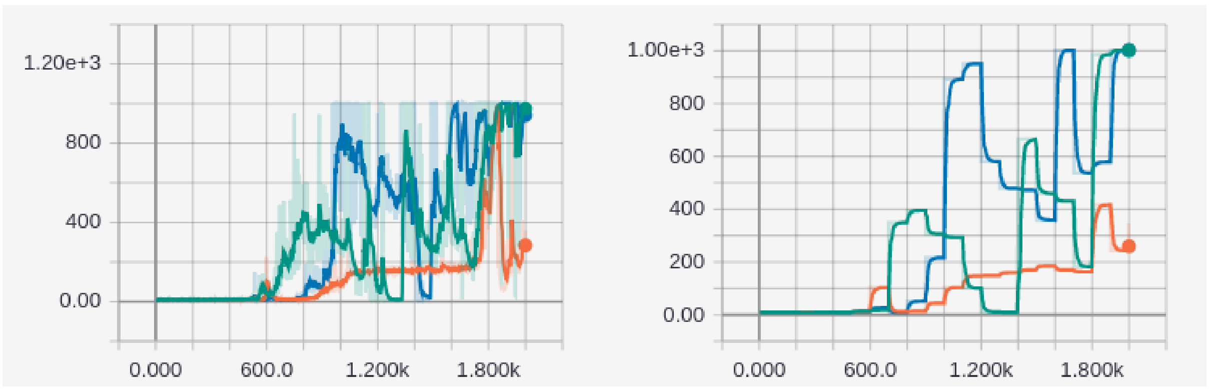

Figure 1: Episode rewards (left) and greedy evaluation rewards (right) for cartpole. The first run is colored green, the second one orange, the third one blue.

- Initial maximum of 50 time steps, increasing in several stages to 1000 by episode 500.
- No state histories (history length = 0).

I conducted this training run on my notebook's CPU, which ran for about 34 hours resulting in 700 training episodes. Out of curiosity I ran relatively frequent rendered greedy evaluations, namely every 20 episodes. After approximately 600 episodes those evaluations yielded acceptable results. By this point the agent managed to take most turns of the track without major difficulties. However, the agent moved relatively slowly and often started to go of track on longer straight patches. Rather than trying to correct its course it mostly just reduced its speed, which sometimes resulted in a standstill besides the road.
Since the agent showed no lasting improvement during the next 100 episodes but alternated between getting better and worse, I decided to stop the training in order to leave some time for different approaches. The episode rewards and evaluation results are shown as the green curves in Figure 2. The model saved at the 700 episode mark yielded an mean test score of 451.6 with a standard deviation of 217.3.
The agent stopping next to the road is most likely due to it having learned to reduce its speed when getting off road in order not to drift. However, since it had no real way of knowing its current velocity, it never stops braking when off track. I suspect that this problem might be solvable by using histories, therefore I started a second training run with a history length of 2 along with some additional parameter changes, i.e. a reduction skip_frames to 5, a faster increase of max_timesteps and evaluations only every 50 episodes. Since those changes slow the training process even more (which I did not really consider when starting it) this second run has at this point in time just reached its 200th episode (blue curves in Figure 2). While this model will most likely need at least a few hundred more episodes to reveal its real potential, it is numerically already almost on par with the first one.

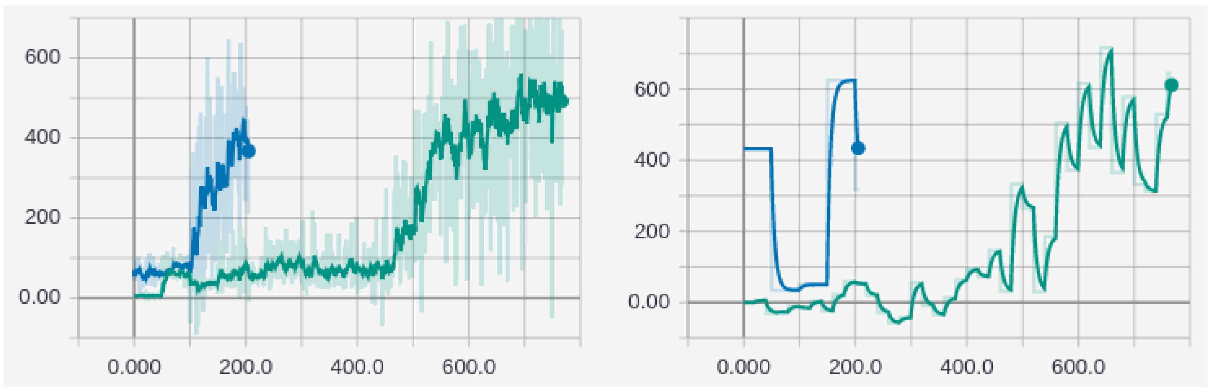

Figure 2: Episode rewards (left) and greedy evaluation rewards (right) for car racing. The first run is colored green, the second one blue.