Report PPO by Nina Braunmiller k11923286

First word

In this exercise I tried out lots of different things. However, my main struggle was that I interrupted training when the reward was after 150 epochs still negative. That was the big mistake as my model delivers not until ca. 734 epochs a positive reward. Through that I learned that I should not to closely listen to others as I heard that positive rewards should already appear after 100 epochs.

Skeleton

The provided notebook of the lecture

Model

For the actor and critic nets I tried out linear layers with normalization and ELU activation. I kept them shallow (each 3 linear layers) as it was recommended by the lecturer. For the actor net I had first the idea to connect one half of the hidden layer‘s neurons with the linear layer of the searched mu and the other one with the sigma. However, it worked better to connect the hidden layer with the parameters‘ layers fully.
The ActorCriticNet was implemented as follows. Mu, the mean of the action distribution we want to sample from, was clipped by the interval [-1+sigma, 1-sigma]. When sampling from that the main part of the distribution (ca. 84 %) we get more likely actions within the needed [-1, 1] value range. To be sure, the sampled actions are also clipped to [-1, 1]. To create a multidimensional normal distribution, I used Normal() and kept the mu input shape of Nx#actions as the different actions are dependent on each other. The reason is that the actions give the angle information for the legs of the walking agent. Originally, I tried out the shape (N*#actions,) which would make the actions independent from each other when using Normal(). Of course that not worked out as good as the one described above.

Agent

Here it is important to define for which parts we want to determine the gradients to optimize. v_target is only an estimate but is treated as target value meaning that it represents some ground truth. Therefore, gradient is not taken from it. Also the advantage function isn‘t the aim of optimization. Therefore, ignore it in optimization.
Sampling ratio relating to the actor policy and the computed value v regarding the critic net are crucial for optimization.
The value loss is computed as mean squared error as it was mentioned by the exercise slide 18.

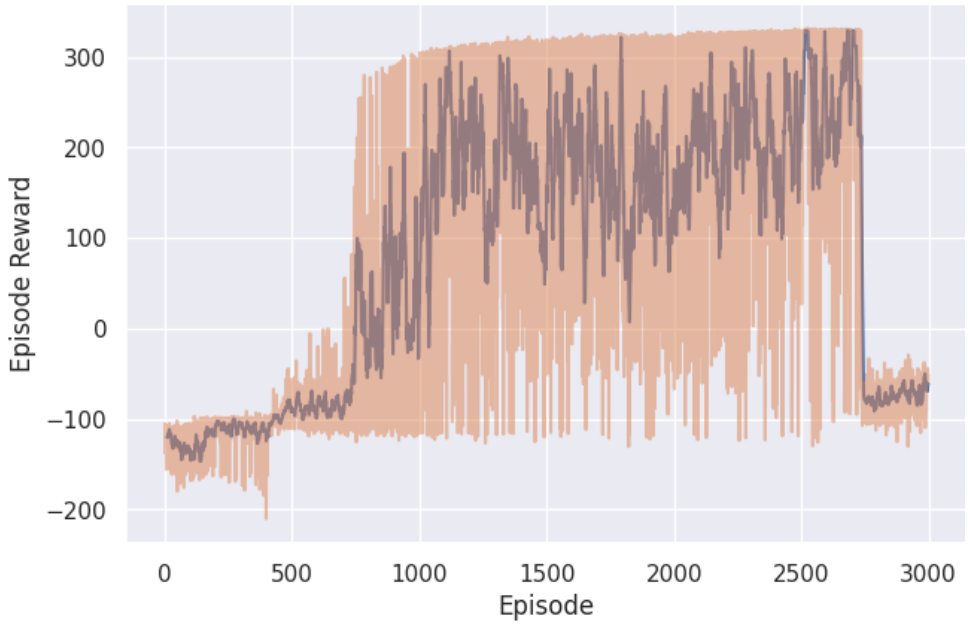
Runner

The runner was similar to the one provided by the lecture notebook. The idea of exercise slide 18 was implemented here. I saved the best models such that the trainer doesn‘t have to end to get the final model. The model I submit is also not the final model but one which was stored in the course of training. At the end of the notebook I added the section ‚Testing on submission server file‘ where I used your submission server tester to determine how good my stored models are. The submitted model reached a score of 145.09036934049215.

Hyperparameters

hidden_size: To compare each action with each other one by weight a 4**4 matrix makes sense. Also weighting the individual action gives us 4 as there exist 4 actions. Therefore, hidden_size=4**4 + 4 made intuitively sense for me.

loss_scales: I had a look on which size range the value, policy and entropy losses are. To balance each of the weights on the same level I used the weights [0.35, 1, 1e-3].
Episodes: I limited them to 2500 as after that limit the agent‘s performance drops surprisingly to the starting performance. Maybe a case of overfitting?



Final performance on submission server

158.848