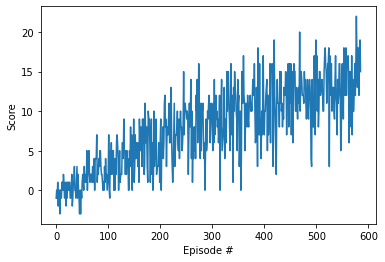
To solve this environment, I use Deep Q-learning algorithm. Since this task is quite simple, there is no need to use a deep neural network with many layers, so, I limit myself to a neural network with four layers. The neuron numbers of the first and the fourth layers correspond to the dimensions of states (37) and actions (4). The neuron numbers of the two hidden layers are 64.

I use the following memory buffer sizes, batch size, gamma, learning rate values:

100000, 64, 0.99, 0.0005,

respectively. The exploration is defined by the parameter epsilon initialized to 1 and exponentially decaying with the decay rate equaling 0.995. The agent successfully solved the environment after 486 episodes reaching the score equaling 13.04.



Also, I consider another version of the environment in which the agent learns from pixels. To make the network understand the motion, I use 4 consecutive frames instead of one. I also set epsilon = 0 to prevent jerky movements of the agent slowing down the learning process. It turns out that the agent trains better on color frames than on black and white frames. However, the training process is significantly slower, despite the fact that I use a much deeper network. In my case, after 1000 episodes, the agent reaches an average score of 8.8, and then the results begin to deteriorate. From my point of view, the performance can be improved by using such algorithms as Policy Gradients, PPO or SARSA, since the reward propagation is slow due to a large number of states.