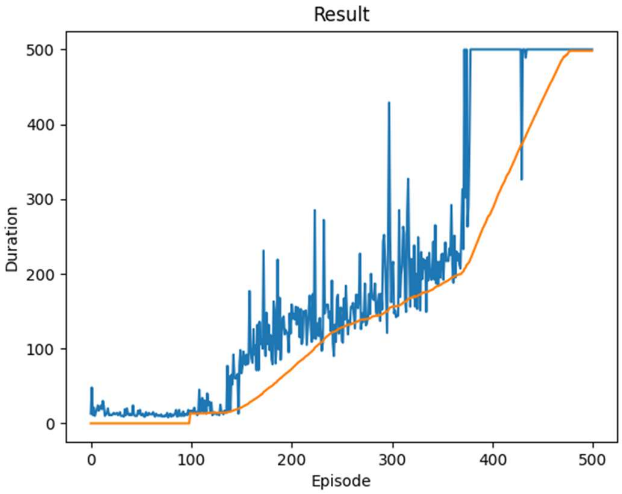
Various network architectures and parameters for CartpoleDQN

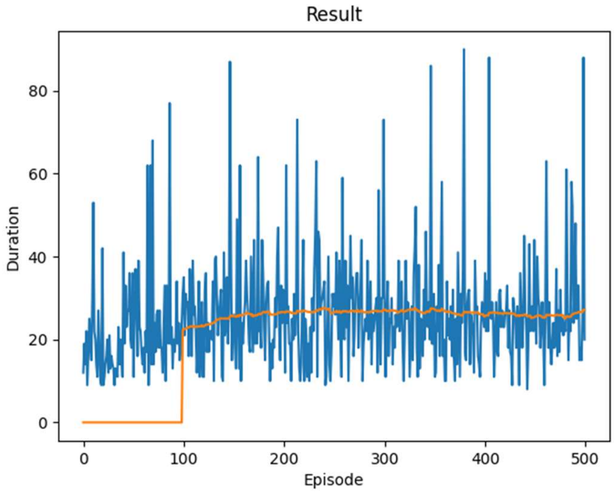
Group K: Fioritto Alessandro, Pilutti Filippo, Simonitti Samuele

CartpoleDQN (1 hidden layer) - example from the lecture



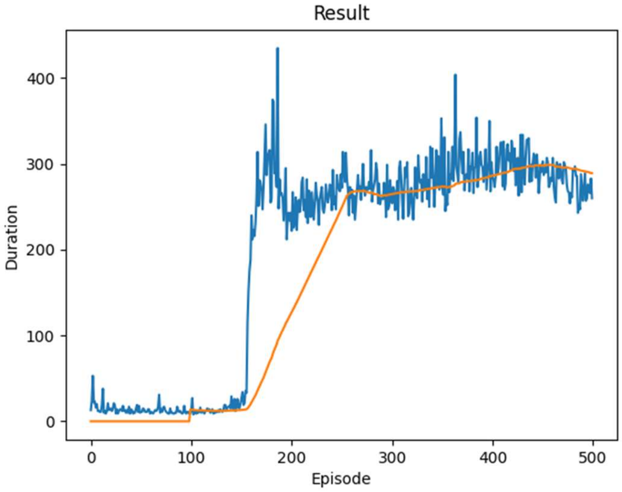
These are the results we got from running the example from the lecture, without changing the model architecture or any hyperparameters.

CartpoleDQN (1 hidden layer) - GAMMA = 0.10



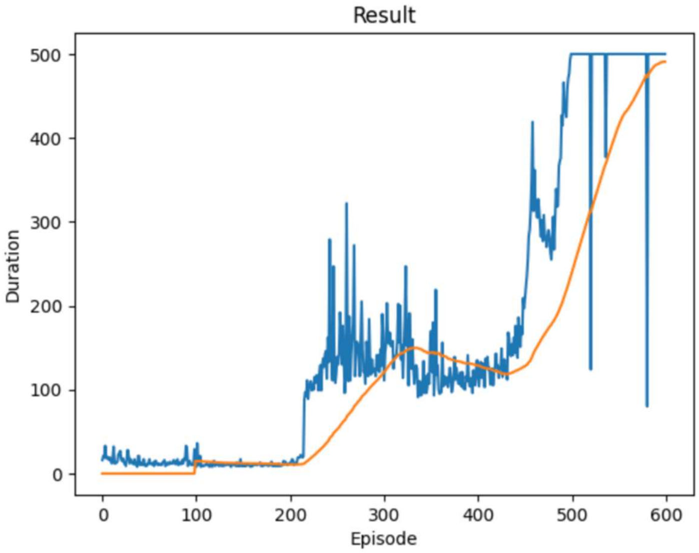
Obviously, training without an insufficient GAMMA parameter leads to the model not being able to successfully predict an efficient policy to maximize the reward on the long run.

CartpoleDQN (1 hidden layer) – Batch size = 32



With a significantly smaller batch size, the model does not reach peak duration time in 500 episodes. It would require training for more episodes to converge to higher duration.

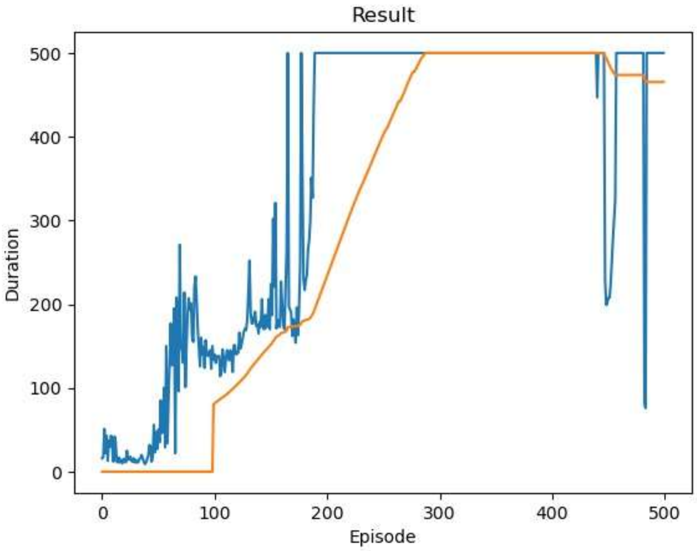
CartpoleDQN (1 hidden layer) – Batch size = 256



We experimented with a bigger batch size, it reaches an optimal duration after about 500 episodes.

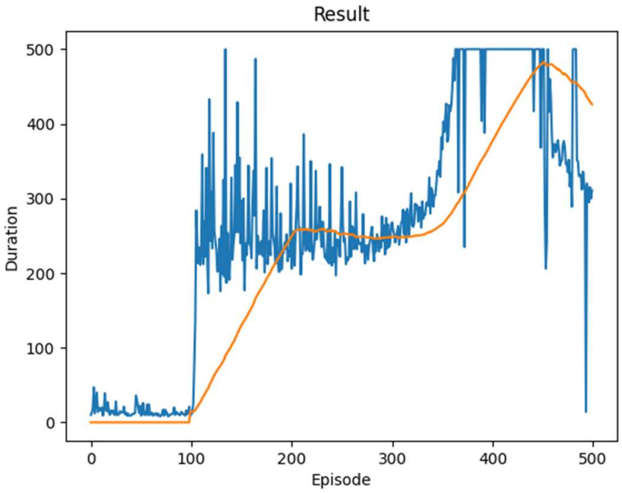
CartpoleDQN (1 hidden layer)

Batch size = 256, learning rate = 0.0003, EPS_START = 0.9, EPS_END = 0.01, EPS_DECAY = 2500



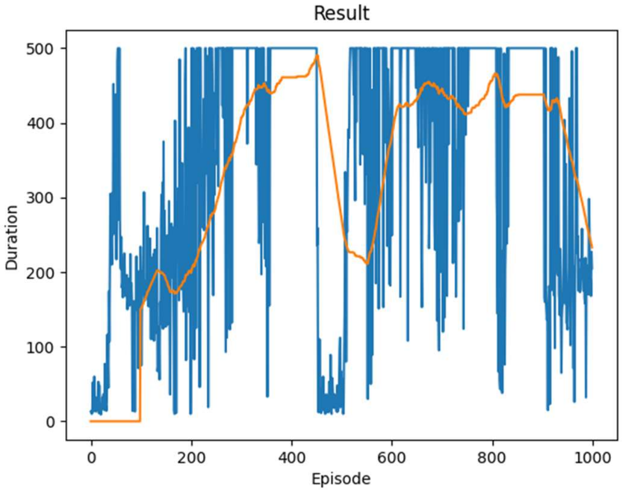
We tried to change more hyperameters at the same time, we wanted to start exploiting the policy quicker, so we changed the decay rate and the final value for the epsilon parameter. We see that the duration starts rising in a smaller number of episodes and reaches the optimal duration quicker.

CartpoleDQN (2 hidden layers)



We changed the model architecture, adding an additional hidden layer with 128 neurons. The model required more time to train, and the results were worse than the starting model. It reached the best duration in almost the same number of episodes as the starting network but showed a higher fall off in the long run.

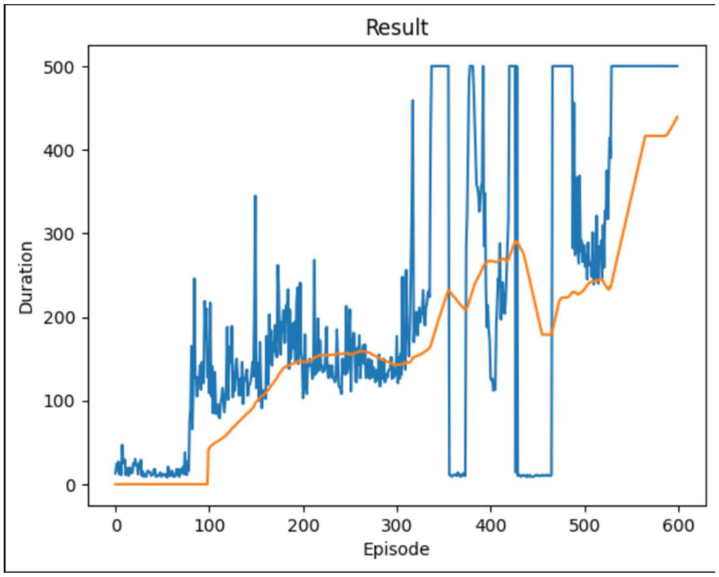
CartpoleDQN (2 hidden layers) – Batch size = 32, learning rate = 0.001, 1000 episodes



We tried to see how the 2-hidden layers architecture performed when we drastically changed the parameters, so we used a small batch size and a higher learning rate and let it train for 1000 episodes. As we can see, the results are bad, probably mostly because of the learning rate.

CartpoleDQN (3 hidden layers)

Batch size = 128, EPS_START = 0.9, EPS_END = 0.01, EPS_DECAY = 1500, learning rate = 0.0003



We tried changing the model's architecture, by adding additional layers with a varying number of neurons: inputs -> 16, 16 -> 32, 32 -> 64, 64 -> 32, 32 -> outputs. We also changed the epsilon decay rate, to try and exploit the found policy quicker.