Neural Networks and Deep Learning
Homework 3

Amedeo Giuliani - 2005797

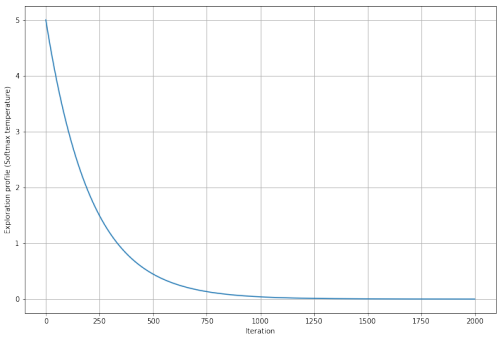
January 16, 2022

1 CartPole-v1 gym environment with compact state representation

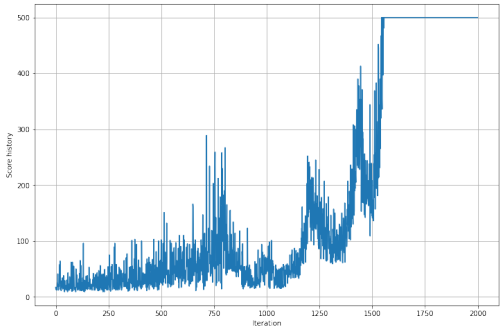
This gym environment at each time step returns a tuple (cart position, cart velocity, pole angle, pole angular velocity) and thus we can exploit these information to allow an agent to control the cart with the goal of not letting it fall. In particular, the agent can only perform two actions: push the cart to the left, and push the cart to the right. To approximate the Q-values we use a *Deep Q-Network* (DQN) a fully-connected deep *Feed-Forward Network* (FFN), having the input layer of 4 neurons, one for each state information, one hidden layer of 128 neurons and tanh activation functions, and an output layer of 2 neurons, one for each possible action. Regarding the learning, we implement a reward penalty on the basis of the cart position, to make the agent understand that keeping the cart in the center is better.

1.1 How softmax exploration profile impacts learning curve

We try with different exploration profiles to see how the learning is affected by them. The first exploration profile is depicted in Fig. 1a, while the relative score history is shown in Fig. 1b. It can be seen that at iteration ~ 1600 the maximum score of 500 is reached, and remains the same until the 2000th iteration. The training took 1 hour and a half, more or less.



(a) First exploration profile tried.

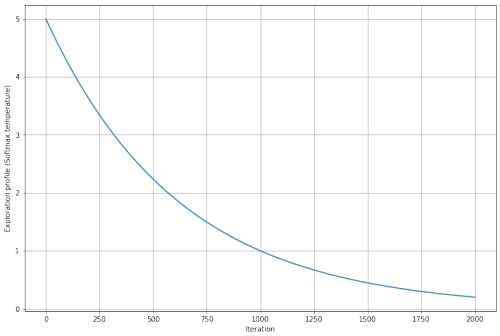


(b) Score history with the profile on the left.

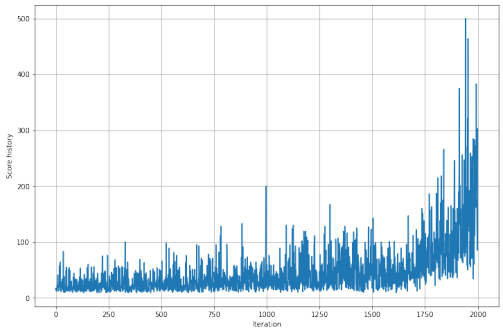
Figure 1

Instead, with a more "relaxed" exponential profile as in Fig. 2a, the learning time is way smaller, but we get the score history in Fig. 2b, which is not very good. In fact, the curve after staying

on average under a score of 100 until the 1750th iteration, it begins to grow, but it reaches only a score of 200 at the 2000th iteration, except for a few lucky attempts. This is due to the fact that the agent does not have enough time for refine its behavior since the temperature remains high for a consistent portion of epochs, that is, the agent prefers exploration to exploitation.



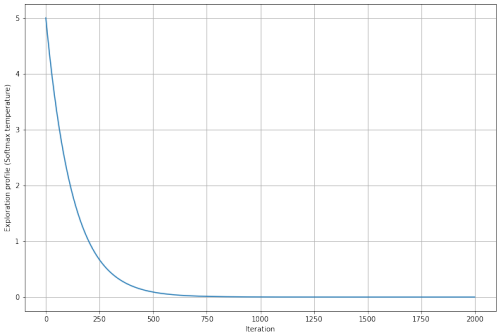
(a) Second exploration profile tried.



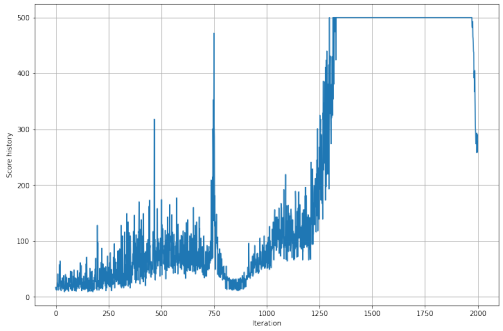
(b) Score history with the profile on the left.

Figure 2

Lastly, with a more "aggressive" exponential profile as in Fig. 3a, the learning time was of 30 minutes because now the temperature falls to zero much sooner. However, the score history, shown in Fig. 3b, now reaches a maximum score of 500 points at the 1300th iteration and remains constant just before the end, where it falls to 300 points for some reason. This is thanks to the fact that now the agent explores a lot within first stages, but then it settles down to the actions yielding the highest rewards.



(a) Third exploration profile tried.



(b) Score history with the profile on the left.

Figure 3

The first case instead is a compromise between these last two situations.

1.2 Choosing reasonable hyperparameters

A grid search or a random search with k-fold cross-validation would be too computationally demanding, but we can try to adjust the values of the hyperparameters to achieve better results. For example, if we keep the exploration profile in Fig. 3a, it can be easily seen that the number of iterations can be greatly reduced. If we try with 1500 iterations, we get the graph in Fig 4:

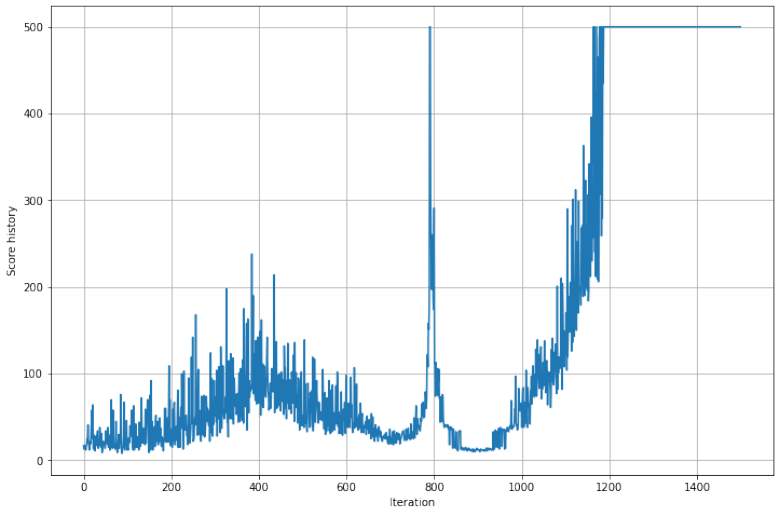


Figure 4

Now we set the optimizer to RMSprop, $\gamma = 0.999$, and the iterations to 1000, obtaining the graph in Fig. 5:

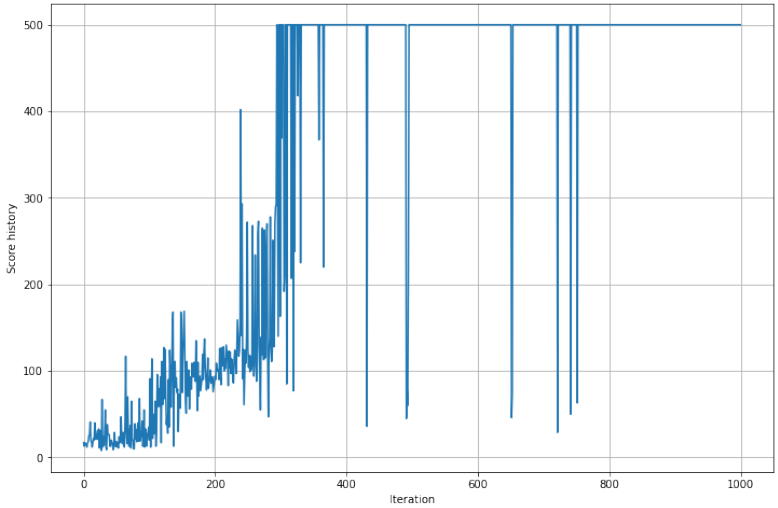


Figure 5

Again, we notice that we can further reduce the iterations to 550, and get the plot in Fig. 6.

2 CartPole-v1 gym environment with screen pixels as state

If the network have to learn to move the cart pole by using directly screen pixel, then some changes must be applied to the model. First of all, for the DQN we use a *Convolutional Neural Network* (CNN), with 3 convolutional layers followed by a fully connected layer, that takes in

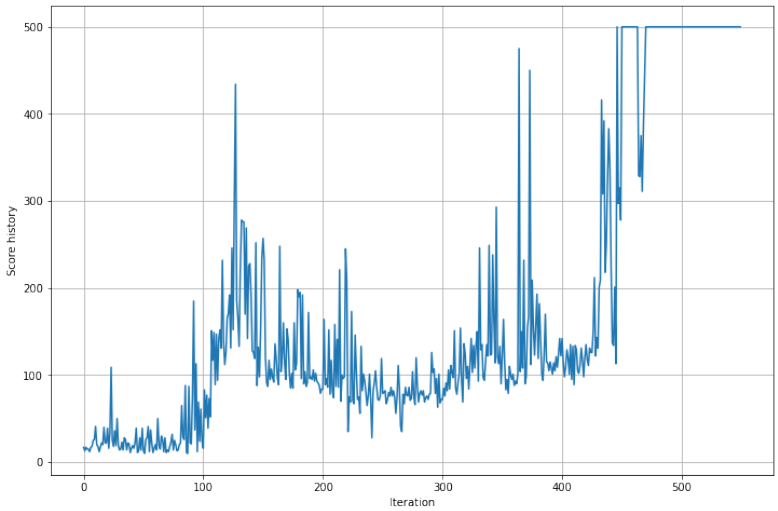


Figure 6

input the portion of the screen returned by the gym environment containing the cart and with always an output layer with 2 neurons since the agent can perform only two actions, still. The screen portion containing the cart is updated at every time step, in order to follow it. So, the state is no more the tuple (cart position, cart velocity, pole angle, pole angular velocity), and instead we use the difference between the current screen and the previous one, to account for temporal context. The reward penalty computed on the position of the cart is still implemented, though. Furthermore, the policy according to which an action is chosen is no more the softmax policy, but the ϵ-greedy policy, with an exponentially decaying ϵ value, as shown in Fig. 7. Now

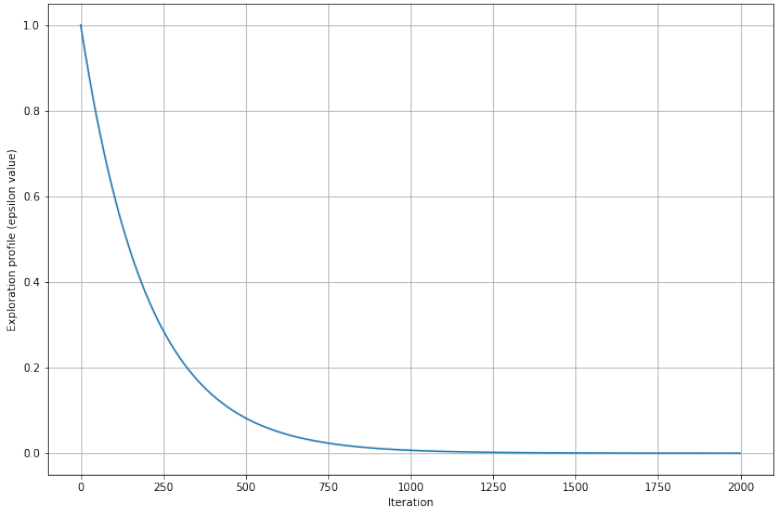


Figure 7

the problem is way harder than before, and in fact a higher number of iterations is needed. In

particular, the epochs number has been set to 2000. However, even with such a high number, the model is not capable of reaching previous scores. In Fig. 8, we can see that the network starts learning and the best results are achieved around epoch 1500 but then the score collapse back to some tens of points.

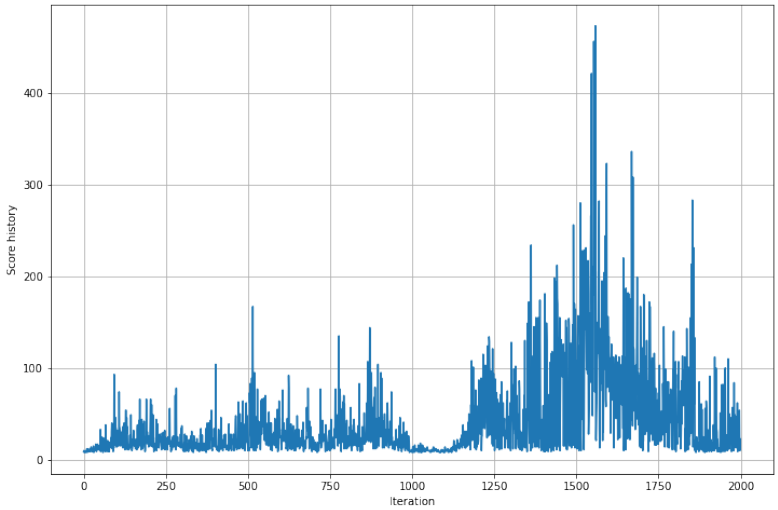


Figure 8