1. Learning Algorithm:

The source code and weights of a trained agent are placed in the main directory. The algorithm is implemented following these guidelines:

The reinforcement learning algorithm is the MADDPG (Deep Deterministic Policy Gradient)based on the implementation used in the lesson.

DDPG belongs to the category of Actor-Critic algorithms. The Actor directly maps states to actions through a policy while the critic performs estimates through a value-based method.

Both actor and critic use a local and target networks. The reason is the networks in use are not constantly updated, thus the learning is more stable.

Also, the two agents shared experience, meaning that whatever any of the agents learned, was reflected in the same model, thus affecting both agents next actions.

The "soft update" function with TAU hyperparameter = 0.05 is used to do the soft update.

- Neural Network configuration:
    - Actor Network (x2):
        - Hidden layers are composed of -> 256 -> ReLU -> action_size -> tanh
        - The optimizer is ADAM and
        - The learning rate=5e-4
    - Critic Network(x2):
        - Hidden layers are composed of -> 512 -> LeakyReLU -> 256 -> LeakyReLU -> 128 -> LeakyReLU -> 1
        - The optimizer is ADAM and
        - The learning ate=5e-4

- Other algorithm configurations:

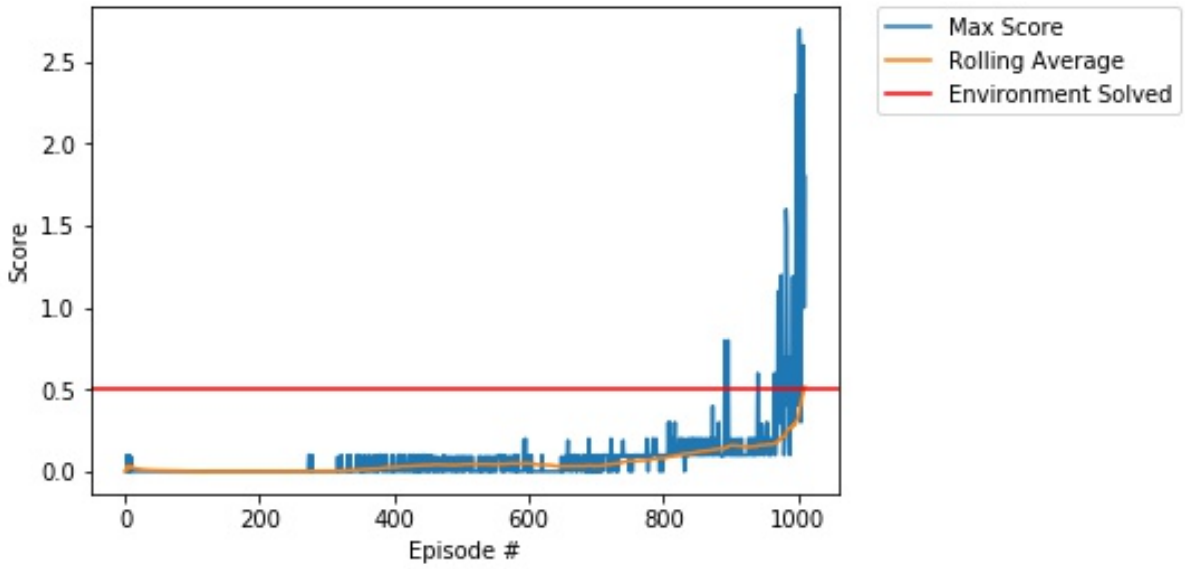
    - BATCH_SIZE = 512 # minibatch size to be processed by the algorithm
    - GAMMA = 1 # discount factor for future expected rewards
    - update_every = 2 # update cycle for the actor/critic network
    - TAU = 0.05 # soft update of target parameters TAU is the rate at which this update takes place so that the network weights do not change suddenly but smoothly.

Experience Replay is also implemented. In this technique, DDPG model is trained by mini-batches from a replay buffer with a size 1e6

DDPG does not "explore" naturally well, thus a noise function is included. The sigma used is 0.2 based on empirical tests.

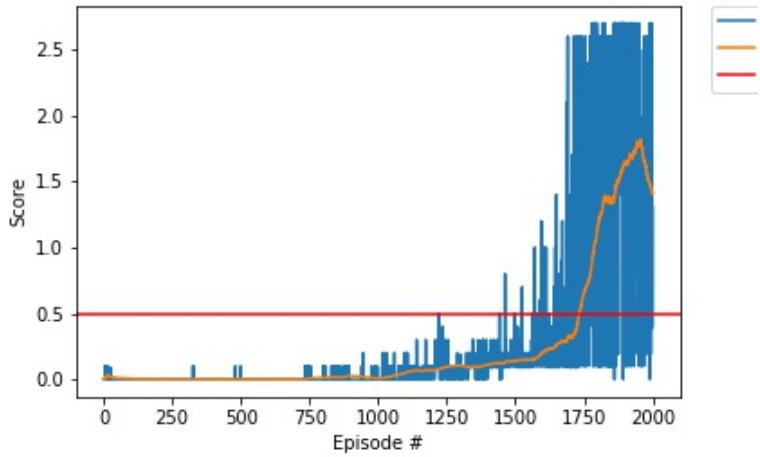
2. Agent Results:

The following plot of rewards shows the training results. At Episode 1010 the algorithm solved the environment.



3. Future Work:

Being this assignment the last assignment many of the "future work" has already been done. My first attempt solved at episode 1732 as you can see below. *(The legend was out of bounds and thus was not saved. Please refer to the previous plot. This issue was solved by making the chart smaller*)



I played around the parameters and tried different "update_every" values, as well as different noise levels among other parameters like TAU and GAMMA. I also tried out to concatenate the states and actions in different layers of the critic. In addition I tried out different models, with 3 and 4 layers.

A very interesting future work would be to solve the environment with another type of agent, like PPO or A3C for example.