

COMPETITIVE TASK - REPORT

The agents are trained to play tennis via DDPG, an algorithm derived from DQN optimized for continuous action spaces.

If an agent hits the ball over the net, it receives a reward of +0.1. If an agent lets a ball hit the ground or hits the ball out of bounds, it receives a reward of -0.01.

Training lasts as long as the score performed by the winner agent (the average over 100 episodes) is not higher than 0.50; in any case, training is stopped if the desired score is not achieved in 10000 episodes.

Each episode consists of 1000 steps at most.

For each step, every agent picks the *best action* predicted for its current state according to a collective model and actuates it (one action is a tuple of torque values). More details are below.

The action is followed by 'reaction' of the environment expressed by the next state (new positions/velocities) and one reward (positive in new position is in the target location)

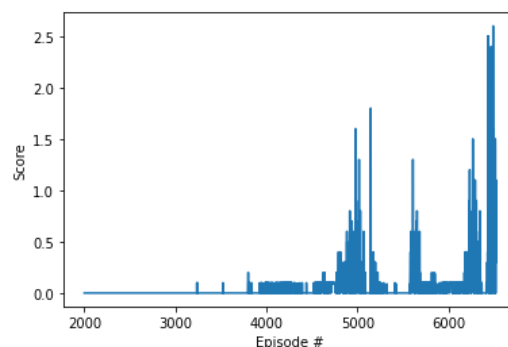
The SARS sequences (current state, action, reward and next state) collected by all agents is hence 'memorized' in a collective memory buffer (the buffer can store at most 10.000 sequences) and the 'experience' of 128 randomly picked sequences from the buffer (as soon as available), is used to train the collective model.

The model basically consists of two deep neural networks: one - the 'actor' - is trained to estimate the *best action*, the other – the 'critic' - is trained to estimate the value (future expected rewards) of the action suggested by actor; in turn, this value is used to train the actor in estimating which the best action is in any state.

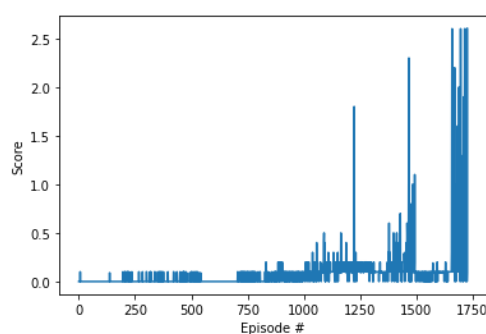
Two additional clones of the models (target models) slowly track the previous ones (the parameters are slightly shifted in their direction, 1‰ at each iteration) and are employed to improve convergence properties of the training algorithm.

The training consists of stochastic gradient descent and back-propagation (learning rate is $1e-4$ for the actor, $4e-4$ for the critic).

With the described algorithm and hyperparameters, the agent has achieved the desired score in roughly 6000 episodes (and has kept it in the next 100 episodes):



In a preliminary training session, the desired score was reached in roughly 2000 episodes:



Sadly, such result was not replicated in any of the next sessions.

Future work may be dedicated to improving performances by testing other algorithms (like D4PG) or to include a regularization term in the loss function to reduce the useless oscillations of rackets that can be seen during the game (see video); the term should be proportional to the square of velocity of the racket/agent...

Further tests may be run to check also the DDPG performances after excluding velocity variables from the observation vectors: indeed, such information may be redundant since the three consecutive time stamps stacked in the observation vector implicitly enable the possibility to infer directions and velocity of racket and ball.