Udacity Multi Agent Control Project

Prepared for: Udacity Deep Reinforcement Learning Nano Degree
Prepared by: Divyanshu Kalra
19 March 2019

SUMMARY

Algorithm

The algorithm used is Soft Actor Critic Algorithm which is derived from entropy based RL learning.

Explanation:
Similar to TD3 simultaneously learn two Q-functions, Q_ϕ1, and Q_ϕ2, by mean square Bellman error minimisation. The Q- functions are learned by comparing MSE on the same target as shown Equation 1 and the policy is learned by maximising Q_ϕ1 similar that of DDPG and TD3 as shown in Equation 2.

$$L(\phi_i, \mathcal{D}) = \underset{(s,a,r,s',d)\sim\mathcal{D}}{\mathrm{E}} \left[\left(Q_{\phi_i}(s,a) - \left(r + \gamma(1-d) V_{\psi_{\text{targ}}}(s')\right) \right)^2 \right]$$

Equation 1.

$$\max_{\theta} \underset{\substack{s\sim\mathcal{D} \\ \xi\sim\mathcal{N}}}{\mathrm{E}} \left[Q_{\phi_1}(s, \tilde{a}_\theta(s,\xi)) - \alpha \log \pi_\theta(\tilde{a}_\theta(s,\xi)|s) \right]$$

Equation 2.

$$L(\psi, \mathcal{D}) = \underset{\substack{s\sim\mathcal{D} \\ \tilde{a}\sim\pi_\theta}}{\mathrm{E}} \left[\left(V_\psi(s) - \left(\min_{i=1,2} Q_{\phi_i}(s,\tilde{a}) - \alpha \log \pi_\theta(\tilde{a}|s) \right) \right)^2 \right]$$

Equation 3.

HYPERPARAMETERS

Actor NN:

6 hidden layers:

Activation: relu

Number of nodes:

1024 -> 512 -> 256 -> 128 -> 64 -> 32

Critic NN:

7 hidden layers:

Activation: relu [tanh on output layer]

Number of nodes:

1024->512->256->128->64->32->1

Optimiser:

LR: 3e-4

Optimiser used: AdamOptimizer

Other Parameters:

Replay Buffer Size = 100000

MiniBatch Size = 64

Discount Factor = Gamma = 0.99

Soft Update Factor = 0.005

L2 weight Decay = 0

Entropy Weight Parameter = 0.0025

IMPLEMENTATION DETAILS

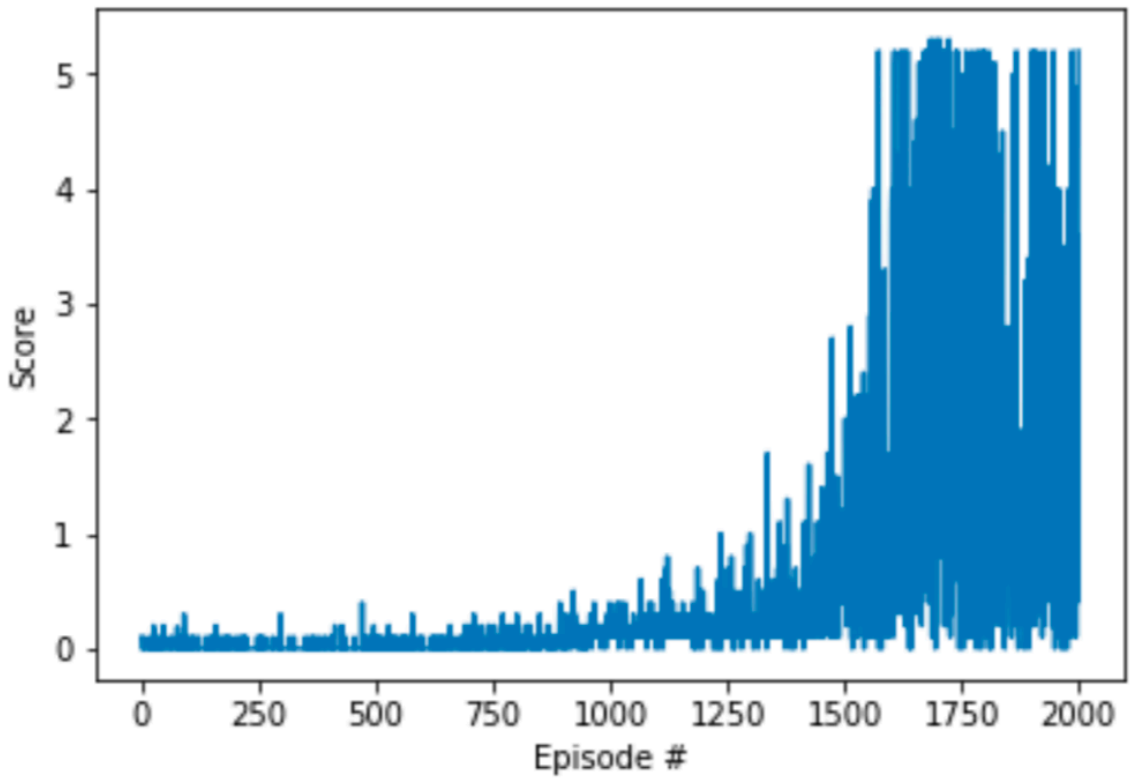
Model.py :

The actor model used for transforming the state information vector into the action vector is written in the class Actor. The actor class also returns the log probability of the action sampled and the mean and standard deviation of the action distribution. The Q - critic model used for transforming the state information vector and

action vector into the action-value function is written in the class Critic. he V - critic model used for transforming the state information vector into state value function is written in the class Critic.

Tennis.ipynb:
The Train function runs multiple episodes of the agent's interactions with the environment for a certain number of transitions. The network stores the average reward collected by the agent over the 100 episodes. Once the agent accumulates over the 0.5+ reward in 100 episodes the environment is considered to solved and model's weight are saved.

REWARD PLOT



FUTURE IDEAS

1) Implementing Actor Attention Critic

2) Implementing prioritised replay.