DDPG SIMPLIFIED PROCESS ALGORITHM

Twin-Delayed DDPG (TD3)

Initialization:

Step 1: We initialize the Experience Replay memory, with a size of 20000. We will populate it with each new transition.
Step 2: We build one neural network for the Actor model and one neural network for the Actor target.
Step 3: We build two neural networks for the two Critic models and two neural networks for the two Critic targets.

Training Process - We run a full episode with first 10,000 actions played randomly, and then with actions played by the Actor model. Then we repeat the following steps:

Step 4: We sample a batch of transitions (s, s', a, r) from the memory. Then for each element of the batch:
Step 5: From the next state s', the Actor target plays the next action a'.
Step 6: We add Gaussian noise to this next action a' and we clamp it in a range of values supported by the environment.
Step 7: The two Critic targets take each the couple (s', a') as input and return two Q-values $Q_{t1}(s',a')$ and $Q_{t2}(s',a')$ as outputs.
Step 8: We keep the minimum of these two Q-values: $\min(Q_{t1}, Q_{t2})$. It represents the approximated value of the next state.
Step 9: We get the final target of the two Critic models, which is: $Q_t = r + \gamma * \min(Q_{t1}, Q_{t2})$, where γ is the discount factor.
Step 10: The two Critic models take each the couple (s, a) as input and return two Q-values $Q_1(s,a)$ and $Q_2(s,a)$ as outputs.
Step 11: We compute the loss coming from the two Critic models: Critic Loss = MSE_Loss($Q_1(s,a)$, Q_t) + MSE_Loss($Q_2(s,a)$, Q_t)
Step 12: We backpropagate this Critic loss and update the parameters of the two Critic models with a SGD optimizer.
Step 13: Once every two iterations, we update our Actor model by performing gradient ascent on the output of the first Critic model: $\nabla_\phi J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s,a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)$, where ϕ and θ_1 are resp. the weights of the Actor and the Critic.
Step 14: Still once every two iterations, we update the weights of the Actor target by polyak averaging: $\theta'_i \leftarrow \tau\theta_i + (1-\tau)\theta'_i$
Step 15: Still once every two iterations, we update the weights of the Critic target by polyak averaging: $\phi' \leftarrow \tau\phi + (1-\tau)\phi'$

DDPG ARCHITECTURE