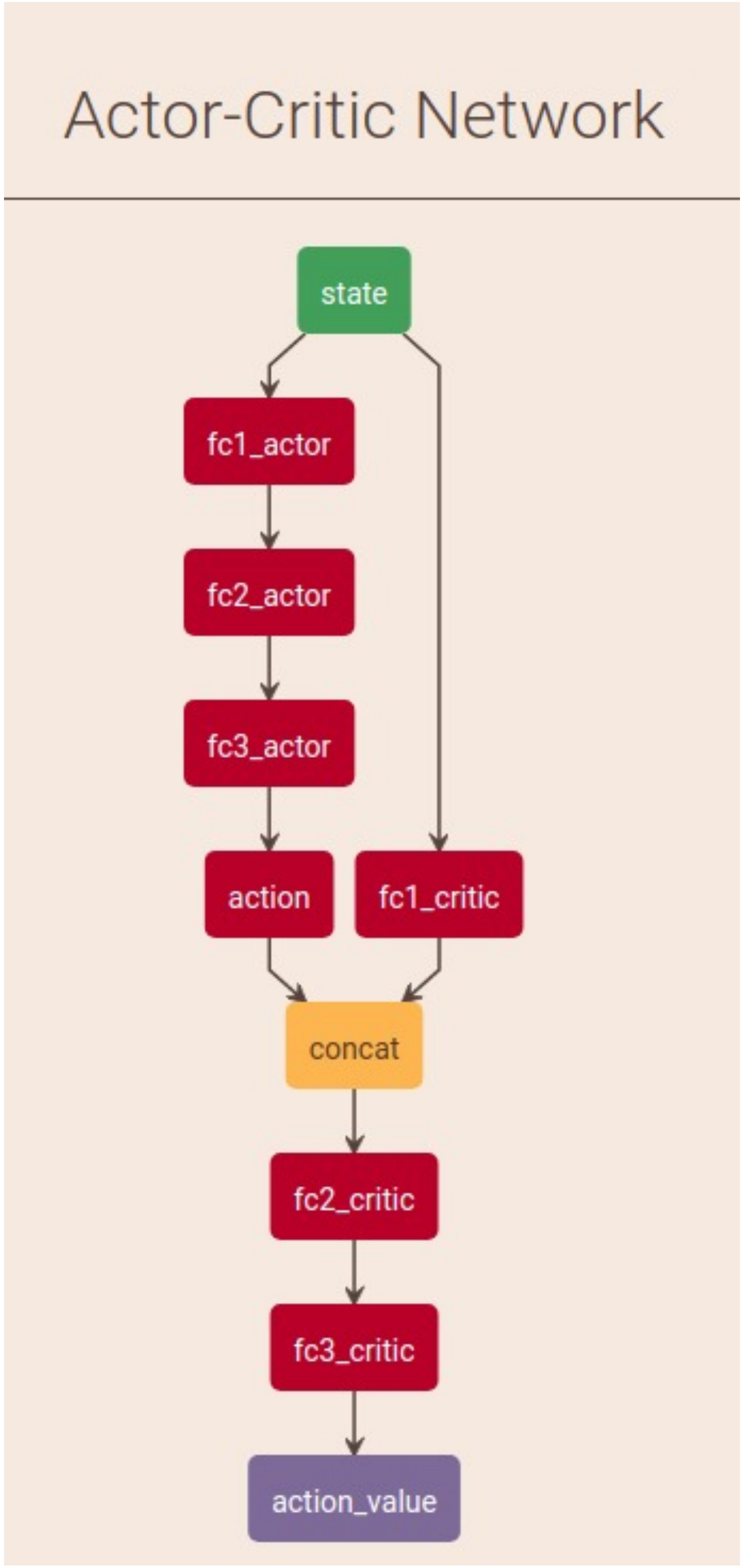
Motivation

By discretizing the action space

- DQN with experience replay
- Actor - Critic with/without experience replay
- REINFORCE

In the continuous action space

- DDPG
  - Deterministic Policy Gradient
  - Off-Policy Learning
  - Actor – Critic
  - Experience replay with cleaning the buffer
  - Action noise
  - Original reward function



Algorithm 1 DDPG algorithm

Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ.
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer R
for episode = 1, M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial observation state s_1
  for t = 1, T **do**
    Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise
    Execute action a_t and observe reward r_t and observe new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in R
    Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R
    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
    Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
    Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_i}$$

    Update the target networks:

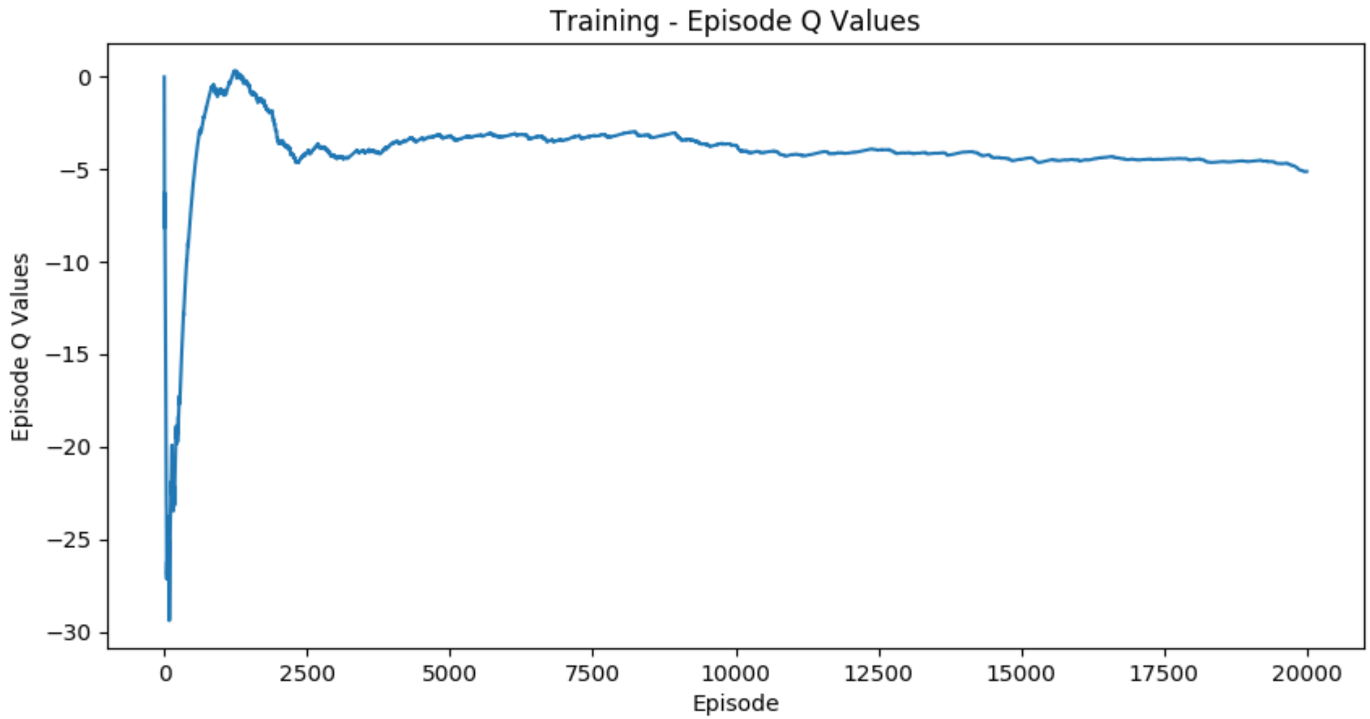
$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$

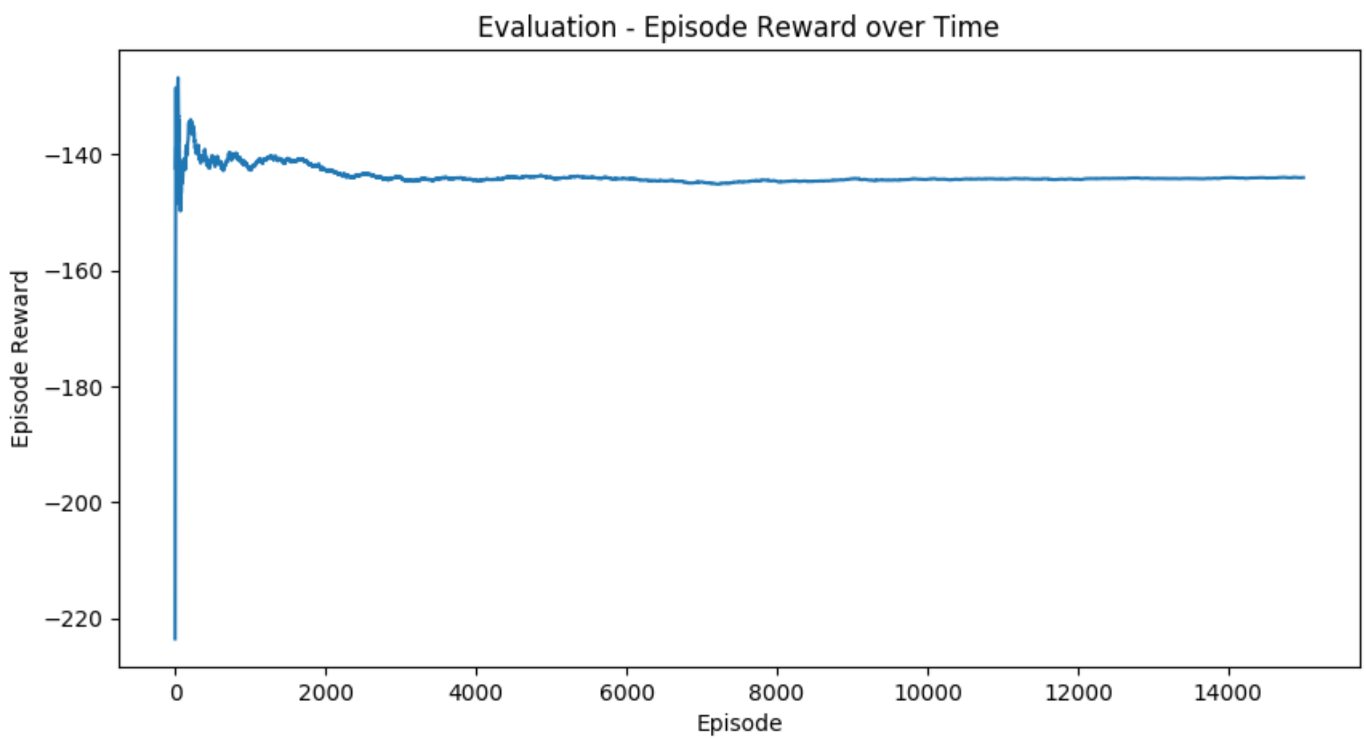
$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

  end for
end for


Training - Episode Reward over Time
Episode Reward
Episode
−400
−600
−800
−1000
−1200
−1400
0
2500
5000
7500
10000
12500
15000
17500
20000


Training - Rewards Quartile
Rewards
Training
0
−250
−500
−750
−1000
−1250
−1500
−1750
1


Training - Episode Q Values
Episode Q Values
Episode
0
−5
−10
−15
−20
−25
−30
0
2500
5000
7500
10000
12500
15000
17500
20000


Evaluation - Episode Reward over Time
Episode Reward
Episode
−140
−160
−180
−200
−220
0
2000
4000
6000
8000
10000
12000
14000

Discussion

- Discretizing the action space didn‘t work well.
- Following the action gradient is a good approach for continuous action spaces instead of looking for a global maximum at each time step.
- Convergence happened after around 1000 episodes.
- Initializing the networks‘ weights had a significant improvement in our case.

Improvements

- Parameter space noise, which can lead to more consistent exploration and a richer set of behaviors – Results from [1] show that RL with parameter noise learns more efficiently than traditional RL with action space noise.
- Priority algorithm for a more sophisticated sampling from the replay buffer instead of uniformly sampling.
- Different hyperparameters settings.
- Minibatch normalization for having unit variance and mean.

[1] PARAMETER SPACE NOISE FOR EXPLORATION - https://arxiv.org/pdf/1706.01905.pdf