Udacity Reacher Project

Thomas Lecat

April 2021

1 Environment

The environment is described as Markov Decision Process (S, A, R, p, γ) with unknown dynamics. The goal is to approximate the optimal policy π^* that maximizes the upcoming cumulative reward in every state.
The action space A is continuous and consists in 4 actions in the $[-1, 1]$ range. 20 agents run independently in parallel on the environment.

2 Agent

2.1 DDPG

A DDPG agent is implemented as the base solution.
DDPG uses two distinct neural networks:

A "critic" network parameterized by θ^Q seek to approximate the optimal Q value Q^*, defined as:

$$Q^*(s,a) = \max_\pi \mathbb{E}_\pi(\sum_t \gamma^t r_t | S_0 = s, A_0 = a), \forall (a,s) \in A \times S$$

The network takes the current observation and action as input and outputs a single value.

A deterministic "actor" network parameterized by θ^μ that approximates the function $\arg\max_a Q(\cdot|s)$. It takes the current observation as input.

The loss function of the critic is defined as the mean squared TD error of the Bellman optimality equation:

$$\mathcal{L}_{critic}(\theta^Q) = \mathbb{E}_{s,a,r,s'}\left[((r + \gamma Q(s', \mu'(s'|\theta^{\mu'})|\theta^{Q'}) - Q(s,a|\theta^Q))^2\right]$$

where $\theta^{Q'}$ and $\theta^{\mu'}$ are the parameters of target networks which are slowly updated towards their respective local networks at every training step. The target networks makes the TD target $r + \gamma Q(s', \mu'(s'|\theta^{\mu'})|\theta^{Q'})$ independent from the parameters θ^Q, which stabilizes the training.

The loss function of the actor strives to maximise the Q value:

$$\mathcal{L}_{actor}(\theta^\mu) = -\mathbb{E}_s Q(s, \mu(s|\theta^\mu)|\theta^Q)$$

The losses are minimized by Adam optimizers, a variant of the Stochastic Gradient Descent algorithm. At each step, the gradient is approximated from a minibatch of experiences $(s, a, r, s', d)_i$. To avoid a strong correlation between the steps of the minibatch, transitions are stored in a large replay buffer during rollouts and then sampled uniformly.

3 Hyperparemeters

The actor network is composed of 3 fully connected layers, with respectively 128, 128 and 64 neurons each, with a batch normalisation layer between the first two fully connected. The actor optimizer is Adam with a learning rate of 10^{-4}.
The critic network is composed of 3 fully connected layers, with respectively 128, 128 and 64 neurons each, with a batch normalisation layer between the first two fully connected. The output of the first layer and the action are concatenated before flowing into the second layer. The critic optimizer is Adam with a learning rate of 10^{-3}.

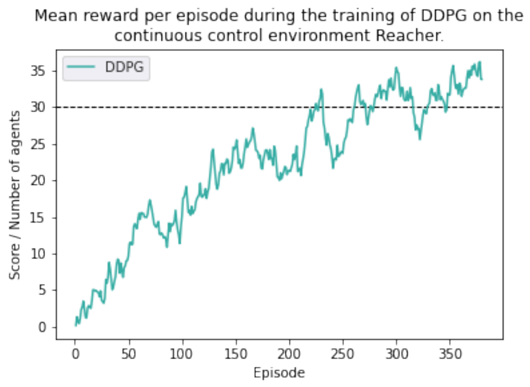
To increase exploration, random noise created by an Ornstein-Uhlenbeck process is added to the actions during rollouts.

The table below summarizes the main hyperparameters:

Hyperparameters		
actor learning rate	α^μ	1e-04
critic learning rate	α^Q	1e-03
discount factor	γ	0.99
target networks update coefficient	τ	0.001
buffer size		1,000,000
batch size		128
critic weight decay		1e-04
Update frequency (number of steps)		20
Number of Adam steps per update		10

4 Results

We plot the mean reward per agent per episode during training:



The environment is solves in less than 400 iterations.

5 Next steps

The next steps for the project consist in implemented other RL agents for continuous control, such as PPO or SAC.
It would also be interesting to discretize the action space and compare a vanilla DQN agent with continuous control agents.
Hyperparemeters could also be further tuned using strategies such as HyperOpt or Population Based Training.