Adrià Ciurana Lanau

Udacity:

Project 2: Reacher

Project description:

In this project it is proposed to use an autonomous agent for use our arm to follow an sphere. A difference of A2C Actor-Critic, this problem has an continuous state and actions. For this reason is a better approach use Continuous Actor-Critic family like: **DDPG**.

In the first instance, we should try to formalize the problem proposed in one of RL.

Variable name	Description
state	A set of 33 sensors that describe the information of the current state.
actions	A set of 4 continue actions with a range between -1 to 1.
reward	+0.1 at each step that the agent arm follow sphere.

Once the problem is defined, we must remember that an RL problem is simply a **maximization problem**. In contrast the previous project, in this case we use a maximization problem that combine: Actor-Network and Critic-Network.

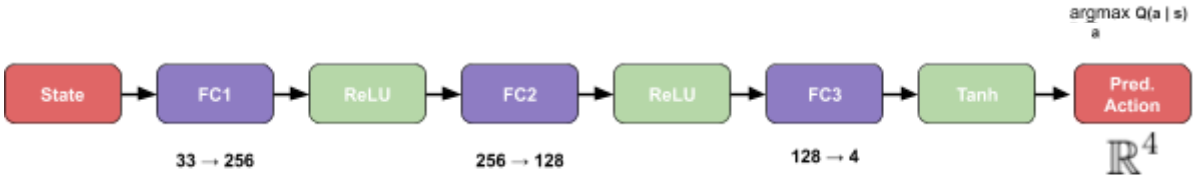
Actor-Critic networks uses a combination of Policy based (Actor) and Value based (Critic). To minimize the variance of the Actor and the bias of the Critic. This aproximation tries to get the best of both worlds.

In our case, we will use DDPG with multiple **unsynchronized** agents. This means that we have 20 agents storing experiences in a common buffer and a single network that obtains the experiences to be learned.

In this case we must differentiate both networks.

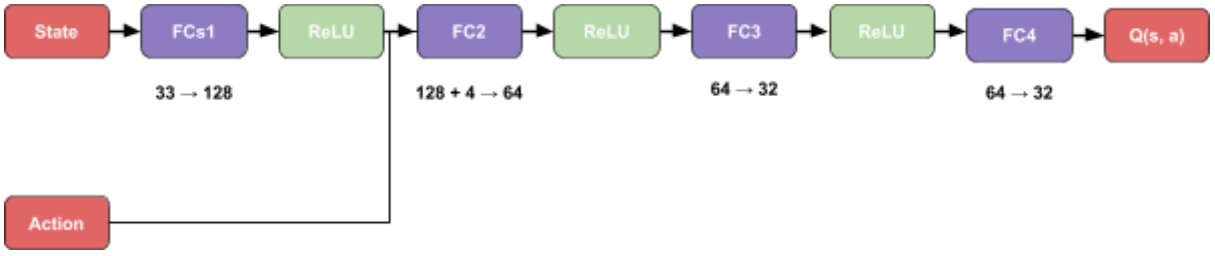
Actor Network:

As the problem is continuous, it is different from the initial concept of the policy based network. Because how could we obtain a function of distribution of continuous values as output? It would be somewhat difficult to work with, for that reason the output is modified and instead of being an output based on a distribution, this becomes a **deterministic output** that is the action that maximizes the probability.



Critic Network:

In this case, as the function is continuous, we must introduce the action within the critic network. The output is only a value that estimates Q (s, a).



Loss definition:

Any optimization problem has a loss function associated with it. In this case, we have defined a loss based DDPG paper. This loss is composed of two phases: critic and actor.

Critic Loss:

$$q(s', a') = Critic(Actor(s'), a')$$
$$q(s, a) = \bar{Critic}(s, a)$$
$$\mathcal{L}_{Critic} = ||\bar{q}(s, a) - [r + \gamma \cdot q(s', a')]\,||^2$$

Actor Loss:

$$\mathcal{L}_{Actor} = -\bar{Critic}(s, \bar{Actor}(s))$$

Agent step:

In this case the learning process is divided into two parts:

1. **Learning of the critical network:** The best next action is predicted using the actor network (**fixed**) and then the critical loss function is minimized.
2. **Learning of the actor network:** The current action is predicted and it is treated as having the highest Q (s, a) possible (**using critic as fixed**).

Subsequently, a smooth update is used to transition the learning of the new networks to the fixed networks. Unlike **DQN**, this transition is very smooth.

When the agent acts with the environment and is in the process of learning, noise is added to the action. Since the action is between -1 and 1, it is quite easy and it is enough to add **Gaussian noise**. This allows to improve the **exploration** of the agent.

To work with multiple agents at the same time, a replay buffer and a global neural network have been used. While agents asynchronously have been filling it with experiences.

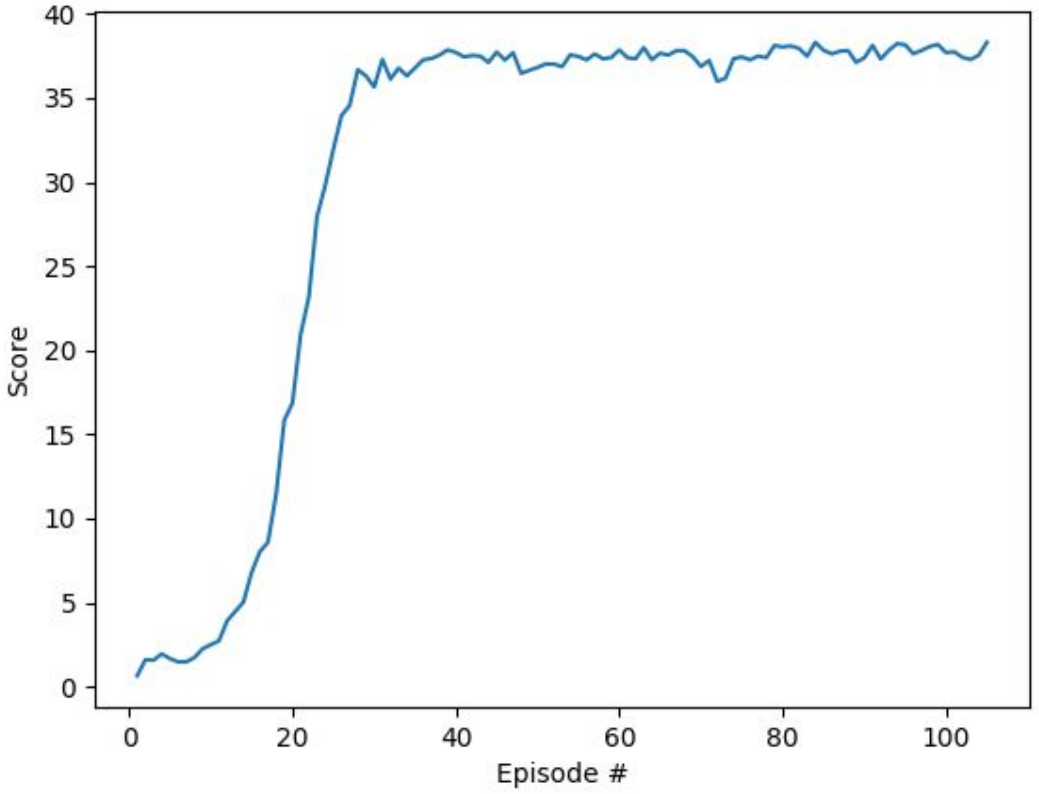
Hyperparameters:

The hyperparameters have been based on the DDPG-Pendulum project, only modifying the **batch size to 1024** to allow a better generalization (since the problem is much larger).

In addition, the **experience buffer is increased to $1x10^6$.**

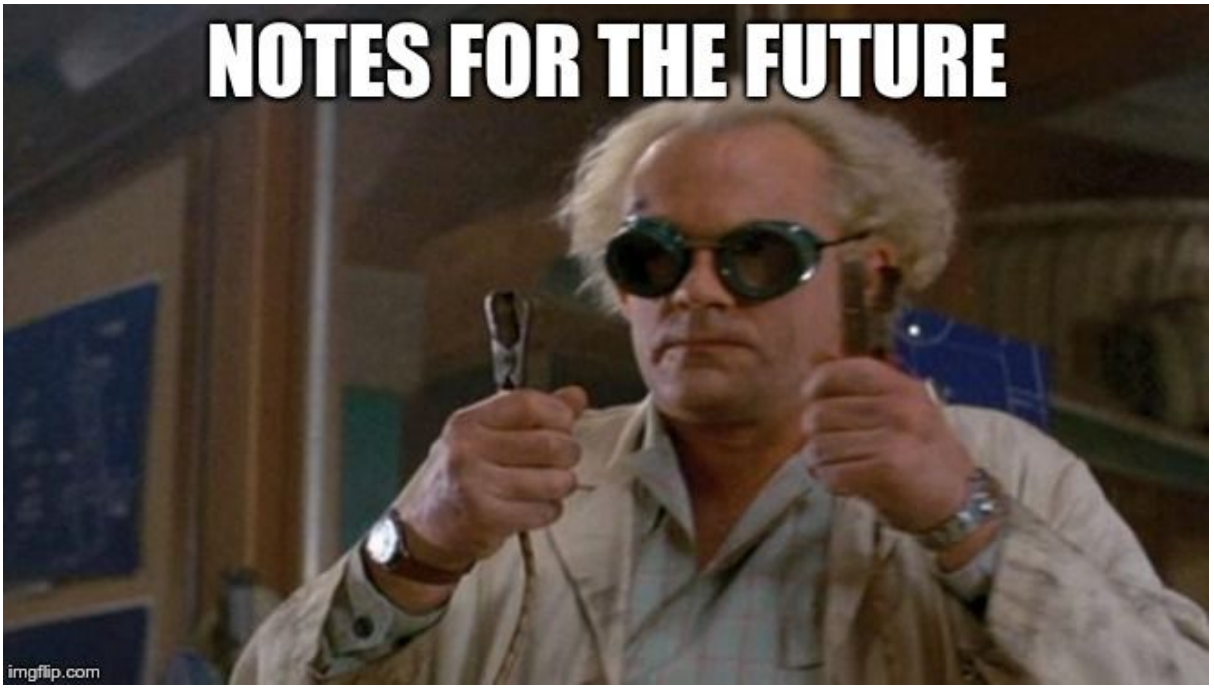
Results:

Below are the results obtained during the episodes. Learning has been stopped when the average score exceeds 30. In this case, it has been **chosen to use the twenty agent environment at the same time.**



Results:

```
Episode: 101
        - Current Score: 37.741499 (+/- 1.063937)
        - 100 Avg Score: 30.687429 (+/- 12.746203)
================================================
Episode: 102
        - Current Score: 37.410999 (+/- 1.434520)
        - 100 Avg Score: 31.045569 (+/- 12.422840)
================================================
Episode: 103
        - Current Score: 37.282499 (+/- 2.229894)
        - 100 Avg Score: 31.402674 (+/- 12.078970)
================================================
Episode: 104
        - Current Score: 37.534999 (+/- 1.354657)
        - 100 Avg Score: 31.758544 (+/- 11.724980)
================================================
Episode: 105
        - Current Score: 38.306999 (+/- 0.967430)
        - 100 Avg Score: 32.124954 (+/- 11.345224)
Average score of 30 achieve
```



In the future…

- Introduce stochastic processes such as Dropout to make new connections.
- Test new hyperparameters.
- It would have been interesting to try other network architectures, together with new RL ideas (for example: A2C, A3C, ...).
- Try to solve unique agent with a more robust method.