Reacher project report

1- Model

For this project, we used the Deep Deterministic Policy Gradient (**DDPG**) model and run it on the 20 agents.

a) Architecture

The model architecture is based on actor and critic methodology.

- **Actor** : the actor contains 3 fully connected linear layers with the following dimensions
  - (33, 512) with 33= state size
  - (512, 256)
  - (256, 4) with 4 = action size
- **Critic** : the critic also contains 3 fully connected linear layers with the following dimensions
  - (33, 512) with 33= state size
  - (516, 256) with 516 = 512 + action size
  - (256, 1)

b) Learning

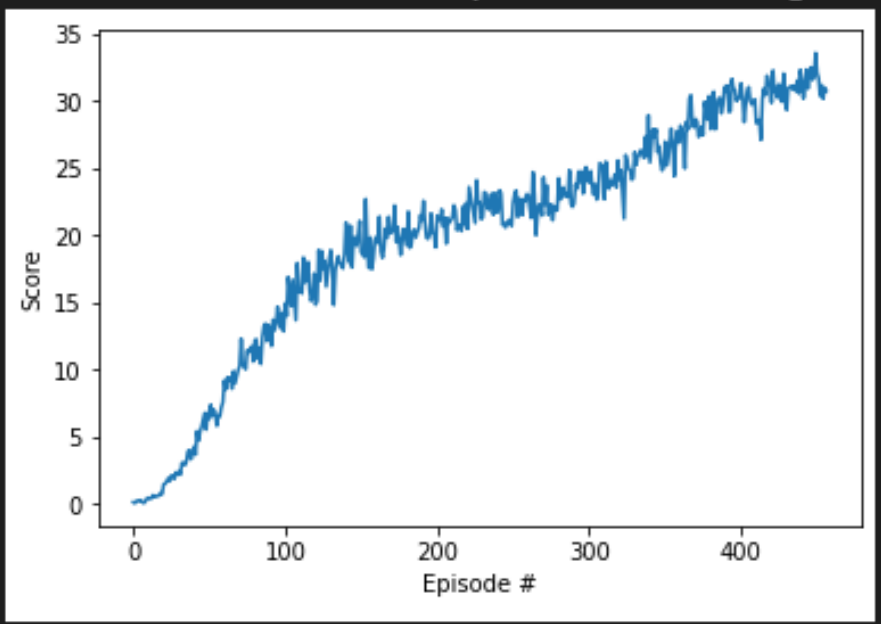
We made the **DDPG** model learn 10 times after passing 8 steps from the memory. In fact, at each step, we save all the agents' information with a positive reward to the shared memory and the zero rewards are saved with a probability of 0.9. We do this selection to make sure that the DDPG agent has less failed steps to learn from and make it learn faster.

c) Hyperparameters

```
BUFFER_SIZE = int(1e6)  # replay buffer size

BATCH_SIZE = 512        # minibatch size

GAMMA = 0.99            # discount factor

TAU = 1e-3              # for soft update of target parameters

LR_ACTOR = 1e-4         # learning rate of the actor

LR_CRITIC = 3e-4        # learning rate of the critic

WEIGHT_DECAY = 0        # L2 weight decay
```

2- Training

```
Episode 100     Average Score: 6.40
Episode 200     Average Score: 18.64
Episode 300     Average Score: 22.26
Episode 400     Average Score: 26.86
Episode 457     Average Score: 30.02
Environment solved in 357 episodes!     Average Score: 30.02
```



3- Next steps

To improve the results we have we can try the following steps:

- Run the model for a longer duration. The model didn't show a sign of declining yet so training it more should improve its performance. It could take days though!

- Try out other actor critique models and compare the results.