DRLND - Collaboration Competition Project Report

Fourat Thamri

March 2021

1 DDPG Actor and Critic Models Specifications

The deep RL method used to train the agents in a multi-agent setting is a Multi-Agents Deep Deterministic Policy Gradient (MADDPG) with experience replay and added noise (Ornstein-Uhlenbeck noise process) like the one learnt from the Udacity 4th course module. Experience replay is a method that saves agent experience in batches and then learn from this experience in a random way rather than learn from ordered episode sequences. This technique is proven to improve the learning process a lot.

The Actor and Critic networks used are fully connected ANNs consisting of 2 hidden layers with 128 nodes each. The actors acts only using their own observations of the environment while critics can use all the state/actions of all other agents in the system.

The table below lists all the hyperparameters used to achieve the results presented in the next section. The model parameters can also be found in the file agents.py and multiagents.py and the associated jupyter notebook.

Parameters	**Value**	**Note**
Actor and Critic Hidden Layers nb	2	-
Nodes nb	128	-
NN Batch size	256	-
Learning Rates (Actor and Critic nets)	0.0001	-
Replay Buffer size	100000	-
Update Frequency	2	How often to update the network
Tau	0.001	For soft update of target parameters
Gamma	0.99	Discount factor

Table 1: Hyperparameters list

2 Results

The model learned its goal well and efficiently by solving the task in 1388 episodes. The task is considered solved when the agent achieves a mean score of >0.5 over 100 consecutive episodes. The plot below shows the reward history of the agents.

```
Episode 100     Average Score: 0.02
Episode 200     Average Score: 0.02
Episode 300     Average Score: 0.02
Episode 400     Average Score: 0.03
Episode 500     Average Score: 0.06
Episode 600     Average Score: 0.08
Episode 700     Average Score: 0.08
Episode 800     Average Score: 0.10
Episode 900     Average Score: 0.11
Episode 1000    Average Score: 0.14
Episode 1100    Average Score: 0.24
Episode 1200    Average Score: 0.32
Episode 1300    Average Score: 0.38
Episode 1388    Average Score: 0.51
Environment solved in 1388 episodes!    Average Score: 0.51
```

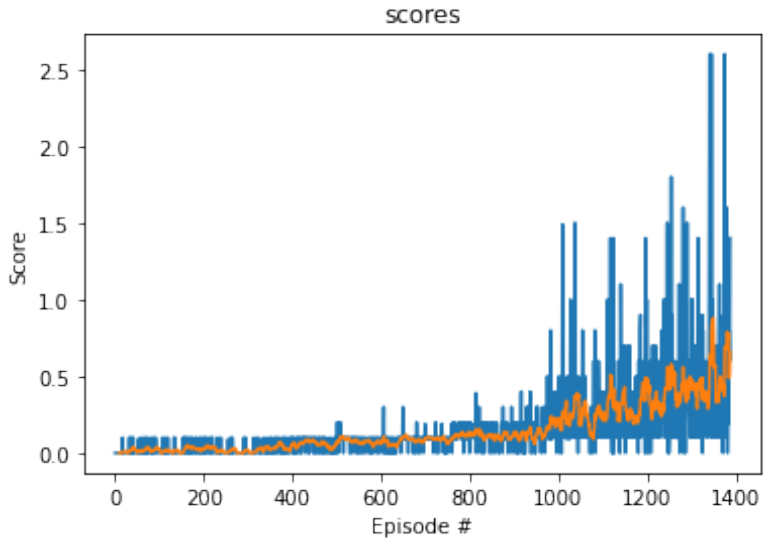

Figure 1: Training scores plot (blue is the score of each episode, orange is for the average score of the last 100 episodes

3 Future work

To further improve the model, a longer training is needed along with a proper optimal hyperparameters search. An implementation of Prioritized Experience Replay would improve the model too.

Anothe big step, would be also to try out to implement the QMix method (Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning) that seems to be promising for MARL systems.