REPORT

Model

A simple default model was used:
Actor and critics both use 3 linear layer

```
BUFFER_SIZE = int(1e5)  # replay buffer size
BATCH_SIZE = 128        # 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 = 1e-4        # learning rate of the critic
WEIGHT_DECAY = 0        # L2 weight decay
UPDATE_EVERY = 10
```

UPDATE_EVERY (network update every 10 steps) was done to increase score, a more stable network to avoid fluctuation). 10 was my chosen number because I found my success in deep q network.

Learning algorithm:
DDPG: Deep deterministic Policy gradient was used:kind of actor critic method, an approximate Dqn (allow to train in continuous space)
, however in this case I used 20 agents to train.
Actor - critic architecture were used to perform this task

Additional highlight:

Alongside with adding **UPDATE_EVERY** hyperparameter, I notice in function ddpg(increasing **max_t** variable - number of steps for agent to explore from 300 to 1000) helps my score to increase instead of having the score plateaued at 10.

Future for improvement
As in discussion another option can be Gradient clipping and update every 10 steps after update every 20 steps

Improving Neural network and hyper tuning parameter can be option

Plot of reward

```
Episode 10      Average Score: 1.36
Episode 20      Average Score: 5.32
Episode 30      Average Score: 10.07
Episode 40      Average Score: 13.70
Episode 50      Average Score: 17.35
Episode 60      Average Score: 20.88
Episode 70      Average Score: 23.37
Episode 80      Average Score: 25.23
Episode 90      Average Score: 26.68
Episode 100     Average Score: 27.83
Episode 110     Average Score: 31.50
Episode 120     Average Score: 34.37
Episode 127     Average Score: 35.78
Environment solved in 27 episodes!      Average Score: 35.78
```