Learning Algorithm

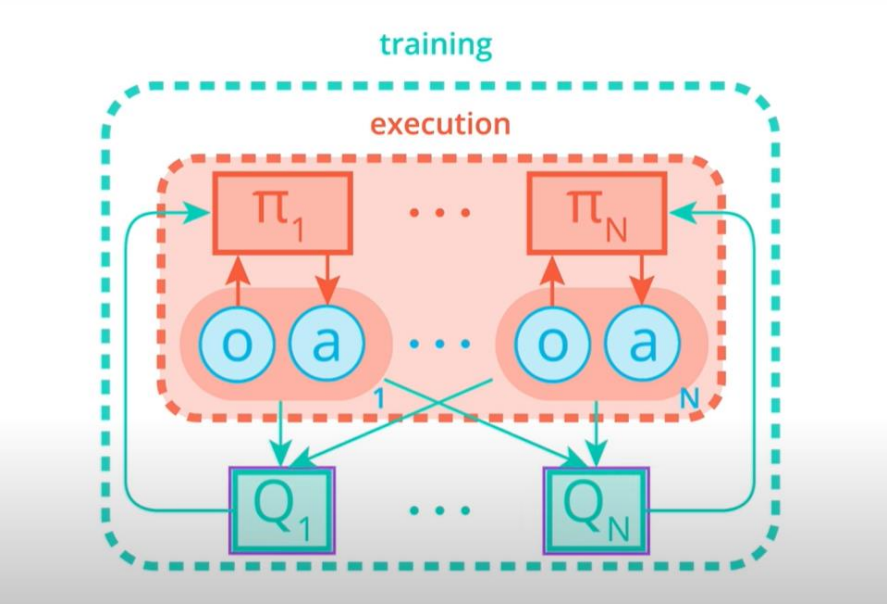
The learning algorithm is used to solve an environment with a 24 variable state space and a continuous action space. Two agents are trained in this algorithm. The algorithm uses a multi agent deep deterministic policy gradient method to output continuous actions from a value reward. Each agent is trained with an actor-critic method, where the actor outputs the actions for the agent to take and the critic outputs the reward value. A shared replay buffer and soft updates are used in this implementation.

$$\langle n, S, A_1, \ldots, A_n, O_1, \ldots, O_n, R_1, \ldots, R_n, \pi_1, \ldots, \pi_n, T \rangle$$

n : number of agents
S : set of environment states
$A : A_1 \times A_2 \cdots \times A_n$
$R_i : S \times A_i \to R$
$\pi_i : O_i \to A_i$
$T : S \times A \to S$

Model Architectures

The model architecture uses a actor-critic method for each agent. The critic neural network consists of 2 ReLu layers and is concatenated the output a value. The actor neural network consists of 2 batch normalized layers, 3 linear layers, and a final tanh layer to output continuous values.



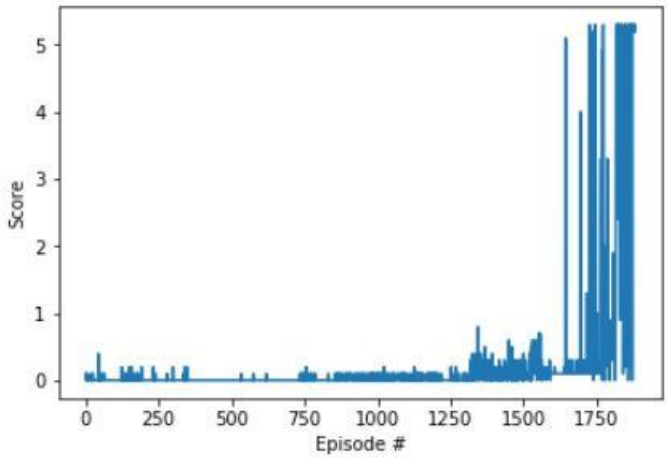
Hyperparameters

Buffer Size	1000000
Batch Size	256
Gamma	0.99
Tau	0.001
Actor Learning Rate	0.001
Critic Learning Rate	0.0001
Update Interval	20

Plot of Rewards

The environment was solved in 1883 episodes with an average score of 3.01.

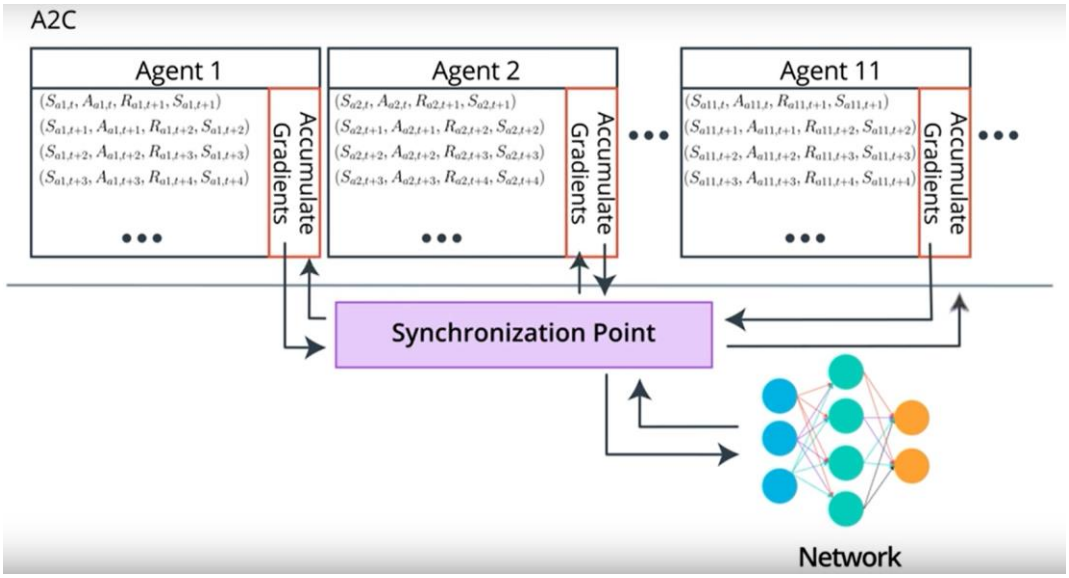
```
Episode 200     Average Score: 0.02
Episode 400     Average Score: 0.01
Episode 600     Average Score: 0.00
Episode 800     Average Score: 0.02
Episode 1000    Average Score: 0.04
Episode 1200    Average Score: 0.04
Episode 1400    Average Score: 0.11
Episode 1600    Average Score: 0.14
Episode 1800    Average Score: 0.85
Episode 1883    Average Score: 3.01
Environment solved in 1783 episodes!    Average Score: 3.01
```



Ideas for Future Work

A2C Algorithm

Implementation of the A2C algorithm can be used compare the training results with DDPG in the GPU.



PPO Algorithm

The PPO algorithm could be used to obtain the optimal policy to train multiple agents with non-interacting parallel copies of the same agent. This would be useful for training the 2 agents in this competition.