# Learning algorithm

## Overview of the technique

The agent is implemented using the Twin-delayed deep deterministic policy gradient (TD3 for short) method. The paper for this algorithm can be found here:   
[Addressing Function Approximation Error in Actor-Critic Methods (arxiv.org)](https://arxiv.org/pdf/1802.09477v3.pdf)   
This algorithm is an upgraded version of the Deep deterministic policy gradient (DDPG). A quick recap, DDPG is an algorithm which concurrently learns a Q-function and a policy. It uses off-policy data and the Bellman equation to learn the Q-function, and uses the Q-function to learn the policy. DDPG is an off-policy algorithm and can only be used in continuous action space.  
TD3 added these following extensions to DDPG:  
 - Use Clipped Double Q-learning technique. Concretely, TD3 use 2 critics network to calculate Q-value and the one with smaller value will be used to perform loss update  
 - Use “Delayed” Policy Update. TD3 updates the policy (and target networks) less frequently than the Q-function. The paper recommends one policy update for every two Q-function updates  
 - Apply Target Policy Smoothing. TD3 adds noise to the target action, to make it harder for the policy to exploit Q-function errors by smoothing out Q along changes in action   
These changes resolve DDPG problem of overestimating Q-value as well as hyperparameters tuning

The algorithm in detail:

Text, letter

Description automatically generated

## Network architecture

The network architecture for actor and critic are described below (the number is # neurons at each hidden layers)

Chart, waterfall chart

Description automatically generated

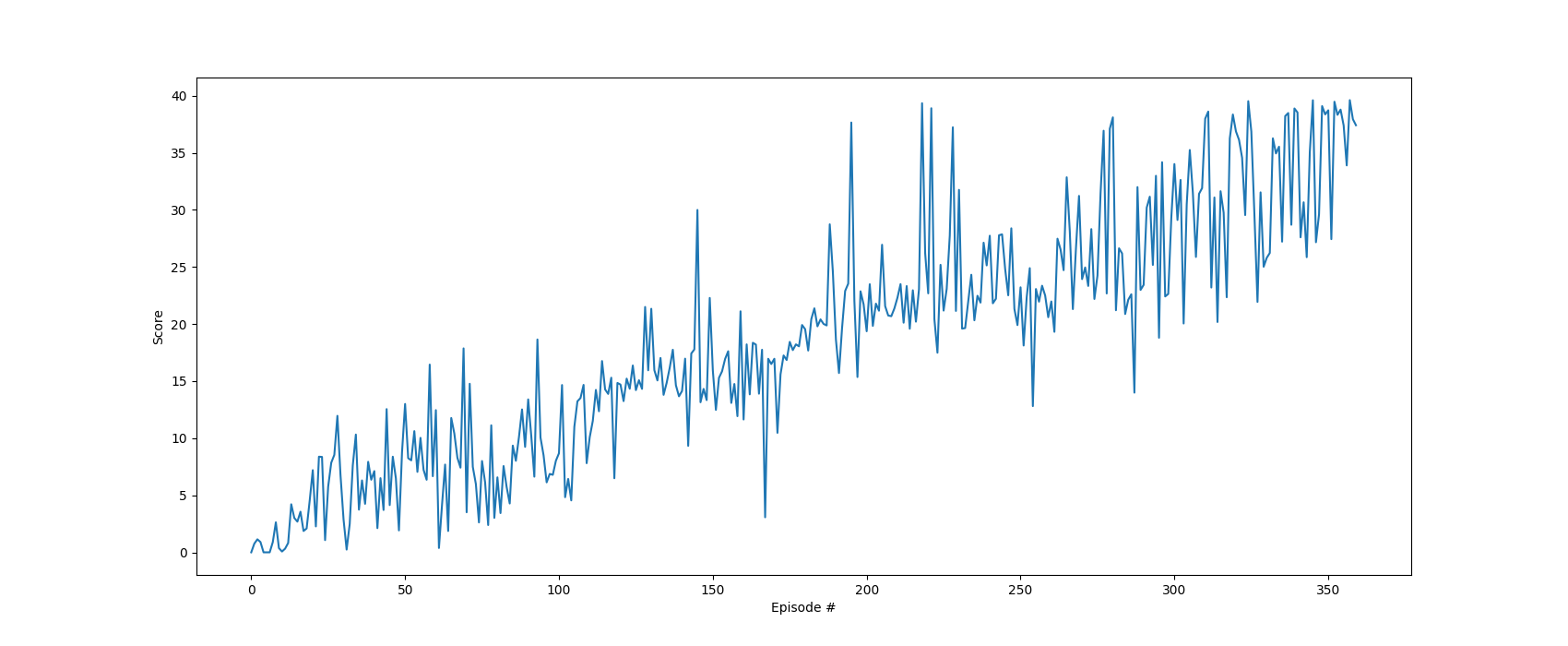
Diagram

Description automatically generated

## Hyperparameters select

|  |  |  |
| --- | --- | --- |
| Name | Value | Meaning |
| seed | 73 | Random seed for stable result |
| actor\_lr | 0.0001 | Learning rate for actor’s Adam optimizer |
| critic\_lr | 0.0001 | Learning rate for critic’s Adam optimizer |
| gamma | 0.99 | Discount factor |
| tau | 0.005 | Soft update target network hyperparameter |
| buffer\_size | 1000000 | Number of transitions stored in replay buffer |
| batch\_size | 100 | Batch size for learning phase of the agent |

# Plot of rewards



# Ideas for future works

The result I have shown is quite good, which solve the problem in around 200 episodes. To be honest, I have played around with DDPG before TD3 and the result is terrible. I cannot solve the problem with 5000 episodes. Here is the plot of DDPG:

Chart, histogram

Description automatically generated

Although the result I got from TD3 is good enough, I think some modifications will make the result even better:

* Tuning hyperparameters such as learning rate and the architecture of the actor and critic network probably enhance the learning ability of the agent
* Moreover, many other algorithms can be used in order to help the agent learn better such as Distributed Distributional DDPG (D4PG), Proximal Policy Optimisation (PPO) or Soft Actor Critic (SAC)