

Navigation Project

RL Agent

I am using temporal difference variant of DQN Algorithm. Here is the [paper](#) of this algorithm.

$$\overbrace{\Delta w = \alpha \left(\underbrace{R + \gamma \max_a \hat{q}(S', a, w)}_{\text{TD target}} - \underbrace{\hat{q}(S, A, w)}_{\text{TD predicted}} \right)}^{\text{TD error}} \Delta_w \hat{q}(S, A, w)$$

This algorithm employs two neural networks with same architecture. Each network has two hidden layers (Linear + Relu).

MSME loss criterion has been used for optimisation. Adam optimizer has been used.

`Agent.py` and `model.py` have been taken from DQN exercise.

Hyperparameters

```
BUFFER_SIZE = int(1e5) # replay buffer size

BATCH_SIZE = 64 # minibatch size

GAMMA = 0.99 # discount factor

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

LR = 5e-4 # learning rate

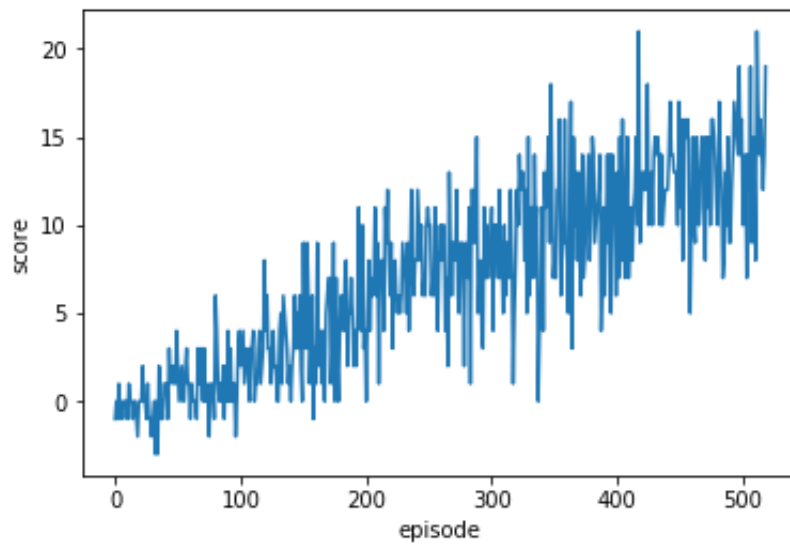
UPDATE_EVERY = 4 # how often to update the network
```

Results

I ran experiment for 1000 episodes. But The target of 13 points has been reached near 450 episodes.

Further optimisation is possible with tuning hyperparameters or using better algorithms involving policy gradients.

Plot of Rewards



Future Improvements

- Implementing **Double DQN** would help agent to achieve reward faster
- **Hyperparameters** can be tuned further for better performance
- **Prioritised Experience replay** would also improve the performance of agent