Deep Reinforcement Learning

Assignment 1 – From Q-learning to Deep Q-learning (DQN)

# Section 1 – Tabular Q learning

1. Why methods such as Value-Iteration cannot be implemented in such

environments? Write down the main problem.

Value-iteration is a model-based method and as such it uses the transition matrix for calculation of the value function:

In scenario like the one described, the transitions (), are hard to model so model-based method will not fit well to this problem.

1. How do model-free methods resolve the problem you wrote in previous

question? Explain shortly.

Using the Q-function to pick action and directly getting the next state by interacting with the environment help us to avoid modeling the unknown transition matrix of this environment.

1. What is the main difference between SARSA and Q-learning algorithms?

Explain shortly the meaning of this difference.

The main difference between these two algorithms is in the Q function update rule. Q-learning updates the q-function under the assumption that the next action will be the one that maximize the Q function ()

And SARSA updates the q-function using action picked by the policy (e.g., .

For this reason, Q-learning try to learn under more optimistic assumption in contrast to SARSA that try to learn under more realistic assumption.

1. Why is it better than acting greedily (choosing an action with 𝑎𝑟𝑔𝑚𝑎𝑥"𝑄(𝑆′, 𝑎))?

When using a greedy policy our agent will always take actions to exploit known rewards and avoid exploring new unknown states and rewards (will remain in subspace of the problem, local maxima). a non-greedy policy such as *decaying - greedy* can “help” the agent to explore new states and rewards.

## 3 layers

### Hyper-parameters

Hidden layers: 32, 32, 32

Epsilon: 0.5

Epsilon decay factor: 0.9995

Epsilon decay steps: 1

Min epsilon: 0.002

Discount factor: 0.97

Learning rate: 0.004

Batch size: 512

Target update episodes: 2

Experience replay capacity: 50000

Learning rate decay factor: 0.98

### Chart, line chart Description automatically generatedRewards

### Losses

Chart, line chart, scatter chart

Description automatically generated

## 5 layers

### Hyper-parameters

Layers: 16 32 32 16 16

Epsilon: 0.5

Epsilon decay factor: 0.9995

Epsilon decay steps: 1

Min epsilon: 0.002

Discount factor: 0.97

Learning rate: 0.005

Batch size: 512

Target update episodes: 2

Experience replay capacity: 50,000

Learning rate decay factor: 0.995

### Chart, line chart Description automatically generatedRewards

### Losses

Chart, line chart

Description automatically generated

# Section 3

### Hyper-parameters

Layers: 16 32 32 64

Epsilon: 0.7

Epsilon decay factor: 0.9991

Epsilon decay steps: 1

Min epsilon: 0.002

Discount factor: 0.97

Learning rate: 0.004

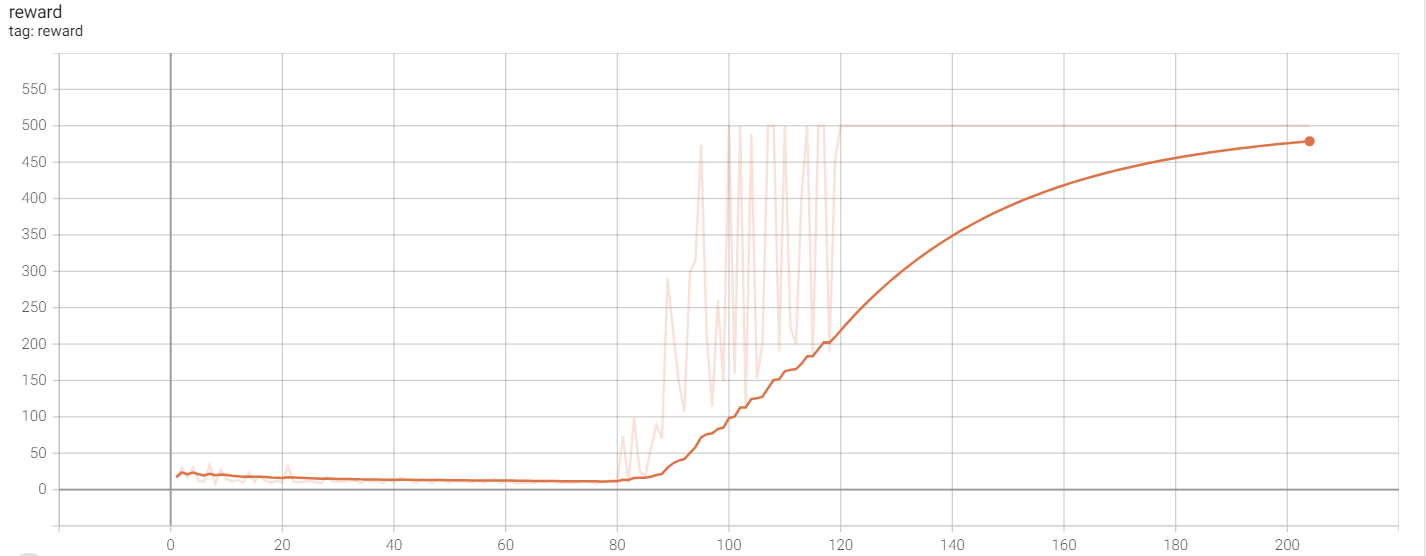
Batch size: 512

Target update episodes: 3

Experience replay capacity: 50,000

Learning rate decay factor: 0.99

### Rewards



### Losses

