# Learning algorithm

## Overview of the technique

The agent is implemented using the Deep Q-Learning method (along with Replay Buffer and Fixed Q-Target technique). At each time step, the agent will take an action, store the transition in its replay buffer (Sample) and then sample a minibatch from the buffer to update the primary Q-network with the target Q-network. The target network is updated either after a number of steps by copying the primary network’s weights or slowly update by tau (a hyperparameter):

target = tau \* primary + (1-tau) \* target

The algorithm in detail:

Text

Description automatically generated

## Network architecture

## Hyperparameters select

Text

Description automatically generated

# Plot of rewards

# Ideas for future works

The result I have shown is not good, which solve the problem in about 4k episodes ☹ I think some modifications will make the result better:

* Tuning hyperparameters such as epsilon (start, end, decay) and the architecture of the Q-network probably enhance the learning ability of the agent
* Moreover, many other technique can be used in order to help the agent learn better such as Prioritized Replay Buffer, Distributing DQN, … or even combine all – Rainbow.