Untitled

January 14, 2022

```
<h1>
    <strong>Report</strong>
</h1>

<h5>
    January 11, 2022
</h5>
```

0.1 Introduction

For this project, we will train an agent to navigate and collect bananas, in a square world.

A reward of +1 is provided for collecting a yellow banana, and a reward of -1 is provided for collecting a blue banana. The goal of your agent is to collect as many yellow bananas as possible while avoiding blue bananas.

The state space has 37 dimensions and contains the agent's velocity, along with ray-based perception of objects around the agent's forward direction. Given this information, the agent has to learn how to best select actions. Four discrete actions are available, corresponding to:

0 - move forward. 1 - move backward. 2 - turn left. 3 - turn right.

The task is episodic, and in order to solve the environment, the agent must get an average score of +13 over 100 consecutive episodes.

0.2 Model

The model used as a function aproximator. The model has 3 linear layers, the firts 2 with a Relu activation function. The first layer with 37 input parametters and 64 output parameters. The second one with 64 input parameters and 64 input parameters. And the third one with 64 input parameters and 4 (the posible actions). The optimization was made using ADAM optimizer to find the difference between the returned value and the expected one.

This method is called Deep Q-learning and allows the agent to use the DNN to detect certain patterns and find an optimal way to solve the situations

0.3 Agent

```
In [1]: from agent import Agent
```

The agent is built with the following architecture: A target DNN: This DNN returns the predicted value. This is the answer that the model finds correct at the moment of making a choice.

```
In [2]: print(Agent(37,4,0).qnetwork_target)
```