RAPPORT DE PROJET

Alvergnat Matthieu - Decreusefond Benjamin - Martin-Gaffe Lucien

Contents

1 Introduction

Having to make a project based on AI for university, we decided to create a trading bot for the stock exchange using deep learning.
Based on what we wanted to achieve we turned to reinforcement learning in order to create a constantly learning bot. We created multiple different agents in order to compare them and see which is the most efficient for a given market.
We also trained on multiple data sets which will be detailed later on.

In this report we will give the details of which agents has been implemented and how they work.

2 Datas

We first had to consider which data we were going to use in order to train and test our agents. We first turned to the abcbourse.com website in order to download the cac40 index value over the last 5 years.

We then find the yahoo finance website which allowed us to download the last 20 years of stock exchange for brands such as Danone, Microsoft, etc..

In those data arrays we stored the following values: date, opening, high, low, closing and volume.

3 Implementation

In this section we will give the details of how our agents were implemented. For the training of our agent we used the Gym environment which is really useful for reinforcement learning. Gym creates a convivial environment allowing the agents to be more determinist and makes the project more portable for the reinforcment learning community.

3.1 Agents

3.1.1 Random agents

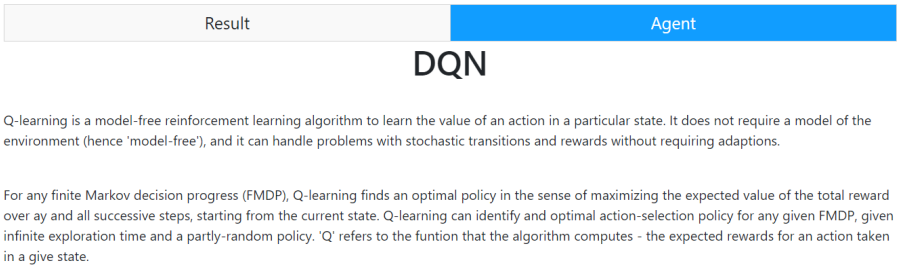
This is the simplest agent of our series. What it does is pretty simple, every day the agent pick a random decision between Buy stock, sell stock, and skip day. The bot will keep repeating until the end of the data set. This agent is very useful to compare the other. This will be our mainline. We must beat a random agent.

3.1.2 Deep Q Network

We choosed the DQN for a lot of reasons. This is an agent that use deep neural network, which is not to complicated to implement (at least for an reinforcement agent)

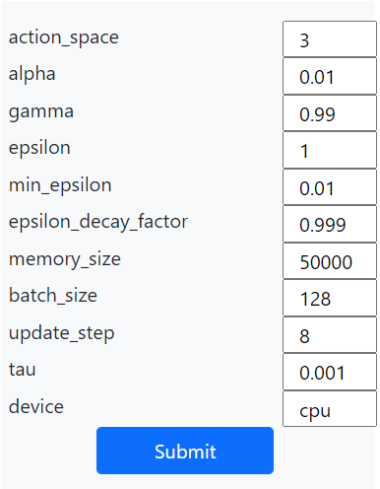
3.2 Interface

Beside the agents implementation, we also wrote an interface in order to make the agents easier to use. We used the Dash python library which use the localhost to display a web interface in order to interact with the bot.
Moreover, you can have informations about our agents in the section "Agent":



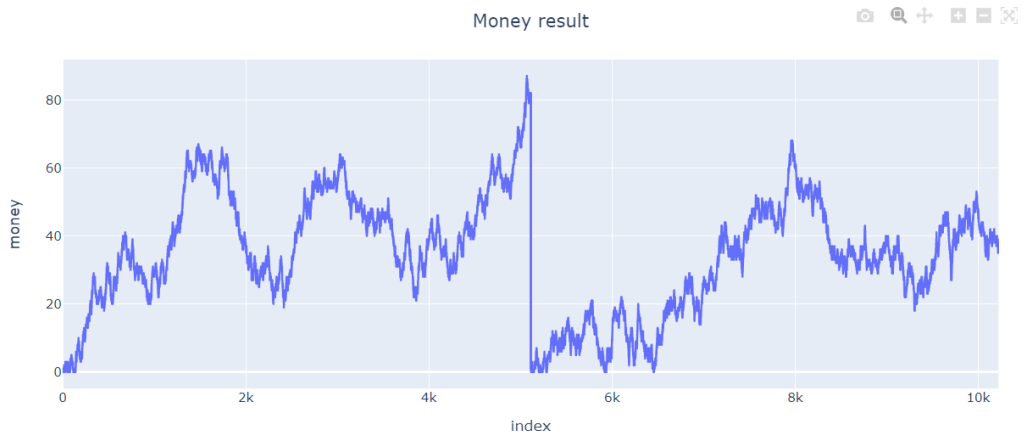
Informations about our DQN Agent

It is also possible to set the parameters of the agents from the interface just after choosing the agent :



Settings for DQN Agent

Once the parameter are set, just click submit and the training will run. You will then be able to see graphs summarizing the results of the agent such as the money over time or the 'score' of the agent.



Random Agent's money graph for 2 episodes

4 Experience

4.1 Agents

Actually, only one agent have been tested, the DQN agent. The experiences were mainly focused on the environment part. We worked hard on the reward function. Indeed, if the reward function is not properly defined, no matter what algorithm is used, it will not work.

4.2 Interface

In this project we tried different solutions in order to accomplish what we wanted for the interface.
In the first place, we tried to use Flask API which allows to do basic web interface in python. However, after spending some time making it work, we realized that it would be time-consuming to design the web pages ourselves from scratch. Indeed, Flask reads the given html page it is given in input but does not provide ready to use design.

We then turned to Dash Flask interface which is a python framework built on top of Flask. Dash has been created to create interactive web applications and figured out to fit our needs for the project.

5 Results

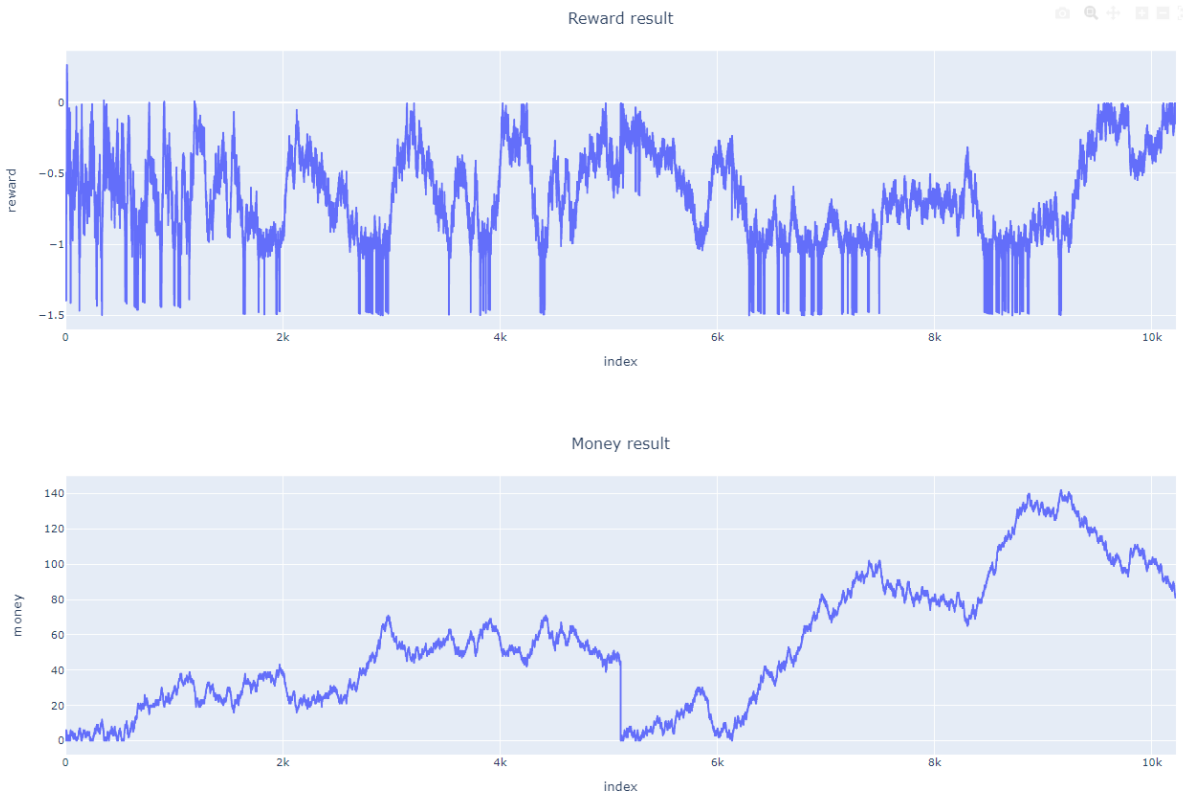
In this section, we will give the results obtained by each agents and compare them in the different environment they were trained in.

5.1 Random agent

This agent being quite simple picking random actions, it almost always lose the entire money it was given at the beginning.
The results of the random agent are not very interesting. It is just a baseline.

5.2 Deep Q Network

After much experimentation, the best results achieved were the following:
The agent converges to a decision to buy nothing in most cases. This is a logical choice that increases the reward but does not achieve our goal. Before testing other algorithms, we need to revise the reward function to reward risk taking.



DQN Agent

6 Conclusion

In conclusion, the expected results are far from those hoped for. Automatic decision making remains an open and difficult problem. However, the work done on this project has provided

a solid base of experimentation for the research community with a first naive but very scalable implementation.

7 References

The book to introduce yourself to the concepts of reinforcement learning: Reinforcement Learning: An Introduction Second edition, in progress Richard S. Sutton and Andrew G. Barto

A practival introduction to deep reinforcement learning

ABC bourse is a very good introduction to stock trading for beginners