# **Agent Based Learning for Trading**

*Yahya JANBOUBI, Ghali LARAQUI*

**Project Description :**

Algorithmic trading has been a practice increasingly more prevalent in the world of financial markets and services. Employing complex algorithms to automate trading decisions based on predefined criteria and training has indeed been seen and proven to yield higher rewards, if done right. This method of trading leverages the speed and computational power of computers to execute a large volume of orders at very fast speeds, but also to be able to capture complex dynamics and unravel patterns that represent an edge to the quantitative trader. Furthermore, as liquidity has become more fleeting in nature and of smaller chunks at all levels in the order book, algorithmic trading is not only a gauge of fast execution, but also of human error minimization, which ultimately reduces trading risk.

Reinforcement Learning provides a very compelling approach to training algorithms to trade. By designing numerical agents that are able to interact with markets live, there is a substantial degree of freedom to build systematic sets of behaviors (policies) that optimally tackle ever-changing markets dynamics and *zeitgeist.* Recent developments, particularly through the lenses of deep reinforcement learning, and more particularly dueling architectures, prove even more efficient and performant in a wide variety of tasks, and thus have been naturally used in the context of algorithmic trading.

In this project, we aim to design and implement various architectures to test whether great returns can arise from numerical agents’ interactions with market signals. While there is myriad of classical and alternative data to leverage, both because of computational resources and restricted access limitations, we focus on a single ticker, namely Bitcoin, and attempt to design a profitable trading paradigm for this specific non-conventional asset.

We first will start by creating a simple action space (Buy, Sell, and Hold), with restricted inventory positioning (only 1 position open at a time maximum), before trying to complexify our model to include more strategies and a more flexible inventory management. Also, we incorporate both trading fees and market impact costs to the computation of a trade’s profit. Finally, the agent’s rewards giving mechanism is kept fairly simple, and is based on whether we are in a profitable or losing position.

We will base our deep learning model architectures on some prominent peer-reviewed research, and while there is extensive work from the academia to support this journey, we will focus on the following works:

* *‘A deep reinforcement learning approach for automated cryptocurrency trading’*, by G. Lucarelli and M. Borrotti (2019).
* *‘Deep Reinforcement Learning with Double Q-Learning’* by H. Van Hasselt and A. Guez (2015)
* *‘Dueling Network Architectures for Deep Reinforcement Learning’* by Z. Wang and T. Schaul (2015).

Other information sources, such as Medium, TowardsDataScience, and, naturally, Wikipedia and GitHub, are thanked in advance for their contributions in furthering our understanding of the technologies and methodologies we will implement to develop our algorithmic trading agents.