Transitioning from Supervised Learning systems to Multi-Agent Reinforcement learning for financial platform

Report By: Prakhar Gurawa*
p.gurawa1@nuigalway.ie
*while working under Dr. Enda Howley and Dr. Enda Barett

Abstract:
Our company has seen immense growth when it transitioned to Algorithmic trading, in particular, high-frequency algorithmic trading (HFT) in the last decade. The availability of labeled data prescribing outputs to inputs made it possible to create a supervised approach to this complex problem for buying, selling, or holding stocks and even portfolio management i.e. the continuous process of reallocating capitals to multiple assets [1]. In this report, we will explore a novel approach to tackle this problem using the concepts of agents, multi-agents, and reinforcement learning. The goal of this report is to show how multi-agent systems and reinforcement learning can be applied to our use case where the macro agent optimizes on making the decision on buy, sell or hold an asset whereas the micro agent optimizes on placing a limit order using the approach described in [2].

Keywords:
Agents and multi-agent systems, reinforcement learning, algorithmic trading, Markov decision process, computational finance, Q-learning, Deep Q learning, Machine learning,

Introduction:
Intelligent agents have widespread use cases from traffic management systems, autonomous GO playing systems to dynamic selection of virtual machines in cloud systems. This novel approach to machine learning is also affecting financial systems. Our company is one of oldest firms in this market and was trading using simple unintelligent programs to keep track of prices till it started using supervised learning which leads to a tech-driven shift. It is high time we try to exploit another approach which includes autonomous multi-agents and reinforcement learning to make our system more robust. As supervised learning can work only with labeled data which is seriously restricting the growth of our organization considering the amount of unlabelled data our organization procures each year and the unavailability of cheap labeled data in our domain. In this report, we are going to describe a reinforcement learning-based system containing macro and micro agents. Here the macro agents will make the decision to buy, sell or hold the assets whereas the micro agent will use the output of the macro agent to make a decision as to where to place the order. For sure our RL approach will lead to saving some money as we wont be requiring labelled data which is costly and not easily accessible.
The financial market is considered as markovian in nature than conveys that the price of an asset depends only on the current price and not on its history. For each agent, we will define states, actions, transition probability distribution, reward function, and discount factor required to formulate our Q-learning model. We will be exploiting the amazing architecture of neural networks through the means of Deep-Q learning where neural nets are used to find the Q function based on states provided to it as action. Deep-Q learning is beneficial for Q-learning in cases where value iterative processes like random exploration are not feasible due to the vastness of data or data that heavily depends on time [11]. The working of basic RL agent and Deep reinforcement learning is depicted in the next two figures on the upcoming page.

Challenges:

- The states possible for our use case are huge in number so it eliminates the use of algorithms like dynamic programming and Value iteration.
- As we are exploiting the deep Q learning algorithms it also brings some challenges like non-stationary or unstable targets etc.
- “Curse of dimensionality” can also be a challenge and also a need of high-quality data is required [11]
- Considering prediction time, the decision time should be considered due to the use of heavy models like neural networks and prediction accuracy, considering financial markets are second-order chaotic systems.

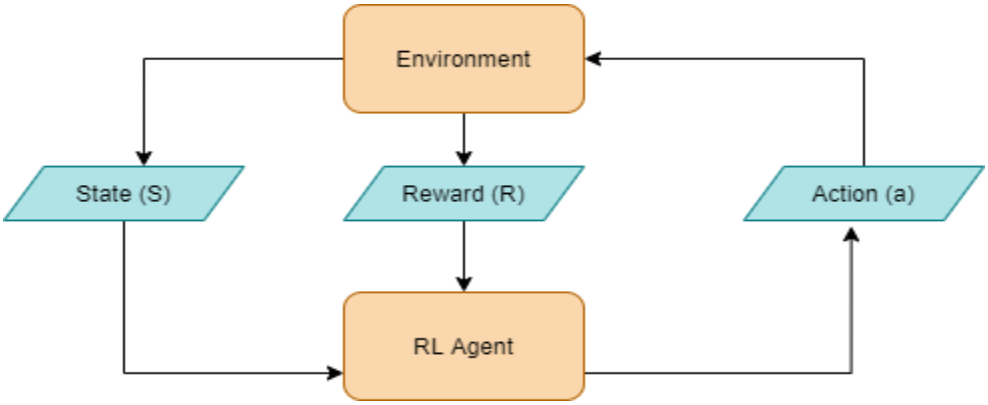


Fig 1. Standard RL situation. At each timestamp t, the RL agent observes the current state of the environment which belongs to set S, and takes an action which belongs to set A which is influenced by the rewards it gets in the process.

Background (A brief overview of Reinforcement learning):

The basic RL system includes states, rewards, action, environment, and the agent. For each agent, the tuple (S, A, P, r, γ) describes the problem :

- S: the set of states
- A: the set of actions
- P: S×A×S → R, the transition probability distribution which determines how the environment changes based on the state and action of the agent
- r: S → R, is the reward function
- γ ∈ (0,1) is the discount factor that determines the importance of future rewards

The sum of future rewards is defined as $R_t => r_{t+1} + r_{t+2} + ... + r_T$

At a time t, the future discounted reward is defined as :

$$R'_t => R_{t+1} + \gamma*R_{t+2} + \gamma^{2}*R_{t+3} + ... + \gamma T*R_T = \sum \gamma^{i}*R_{t+i+1}$$

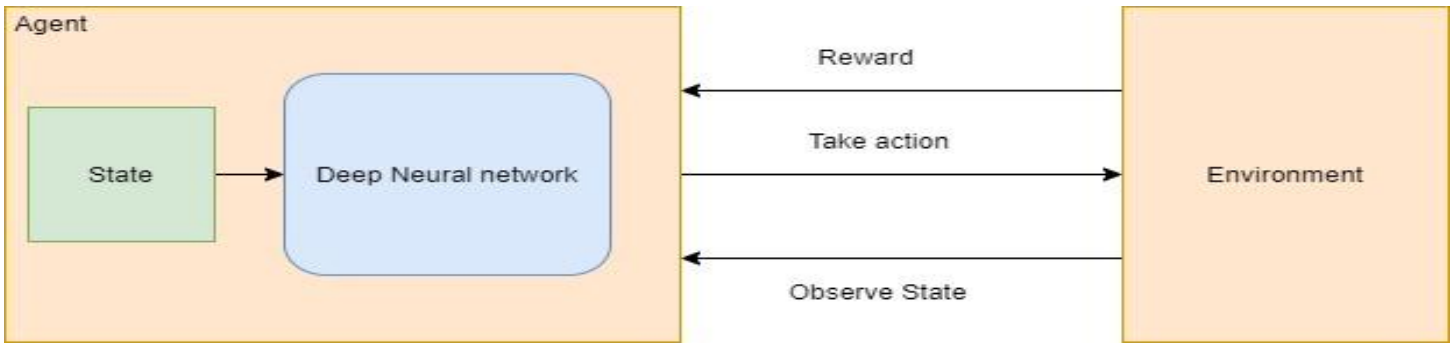


Fig 2.An agent uses DRL to search for optimal policy function Π for each observable state s, providing action to obtain the maximal expected reward. This figure is obtained from [14]

Understanding agents,multi-agents, and cooperation among multi-agents:

Agents:

Russell and Norvig [3] state agents as anything that can perceive the environment through sensors and act upon it using its actuators. Macal [2] describes agents as objects with characteristics such as identifiable - as a discrete individual that govern decision-making capacity, located - settled in an environment where it can interact with other agents with a goal-driven mindset with flexibility, adaptability, and self-containedness. In our context of the financial environment, we can state agents as a software system that is capable of individual actions on behalf of the user by responding to states and events in their environment acting on behalf and in the interest of the owner to meet its design objective. Few important features of intelligent agents are described below in the diagram.

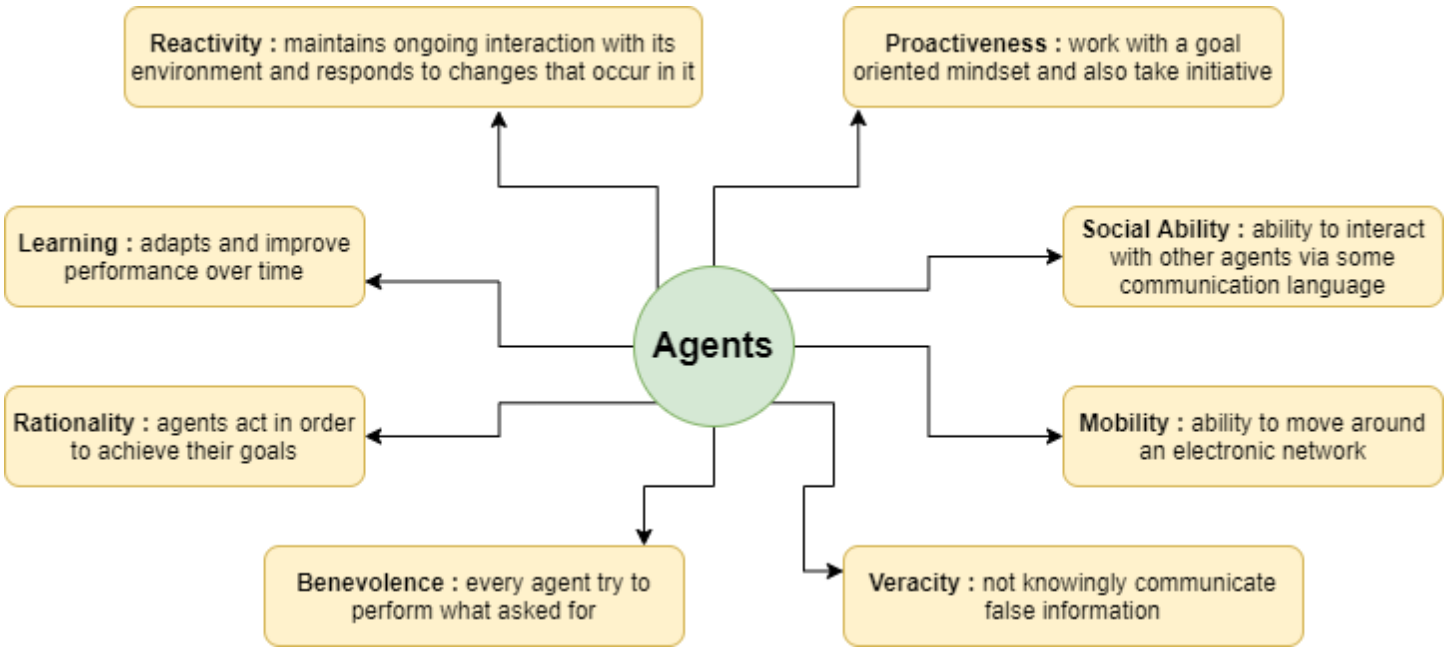


Fig 3. Features of intelligent agents

Agents are features that make them different from objects where objects encapsulate some states and communicate via message passing having no autonomy, agents are autonomous, smart, and active. Also to build purely reactive agents can be a simple task, but not desirable. Considering different aspects like autonomy, learning, and corporation, Nwana[5] bifurcates agents as four kinds (i) collaborative agents, (ii)collaborative agents with memory (learning), (iii) interface agents, and (iv) truly intelligent agents.

Multi-Agents systems:

A multiagent system is preferably used in solving problems that are difficult (or impossible) to be solved by individual agents. Stone and Veloso [4] defines MAS as a loosely coupled network of problem-solving entities (agents) working together to find answers to problems that are beyond the individual capabilities or knowledge of each entity (agent). To successfully interact they need to cooperate, coordinate and negotiate with each other just like humans do. It has become a core area of research of contemporary artificial intelligence. After solving the problem of agent design it comes to designing society for a multi-agent system. The most crucial aspect to consider in the case of multi-agents to make them work with harmony and coordination.

Interactions and cooperation among multi-agent systems:

The corporation is essential for evolution among agents, described mostly in many research papers like [7] and [8] using the Prisoner's Dilemma example which helps us understand the interaction among agents. The problem is described as each player must make a decision to cooperate(C) or defect(D) and finally get the relevant rewards or payoff as described before.

	C	D
C	R, R	S, T
D	T, S	P, P

Table 1. Payoff Matrix for PD problem with R = Reward, P =Penalty,S= Sucker, T= Temptation

The inequalities can lead to different orientation and cooperation level among agents such as if R>P it will lead to mutual corporation preferred over mutual defection, R>S leads mutual corporation preferred over being exploited by the defector, with 2R> T+S mutual corporation is preferred to equiprobable unilateral cooperation and defection, T>R leads exploiting a cooperator (Greed) and P>S causes mutual defection and not exploitation (Fear).

Fixed biased tagging helps create subgroups and leads to an accurate mechanism for biasing agents based on their relatedness as described in [7]. Levels of cooperation can be varied according to one's use case and requirement based on adding tags to agents which act as marking or social cues. The concept of tagging makes sense in problems like Iterated prisoner's dilemma where two or more entities (agent) have to interact more than once and thus or a harmonious working of the system, tagging is an appropriate method.

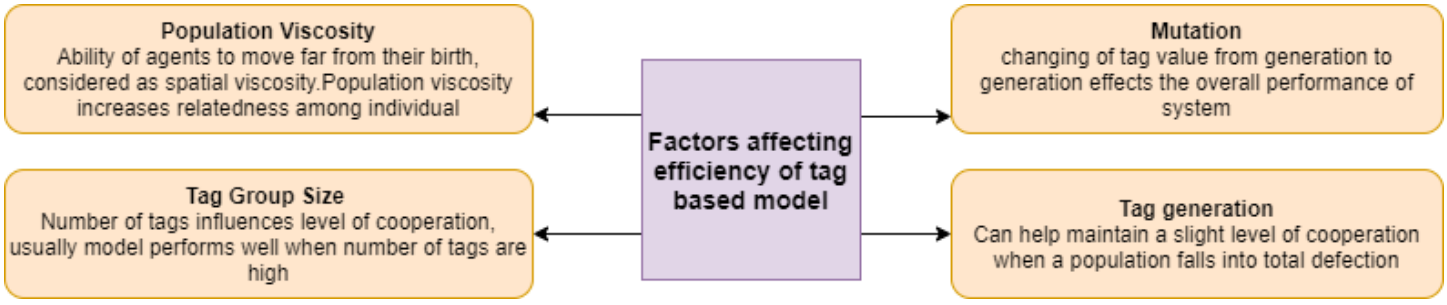
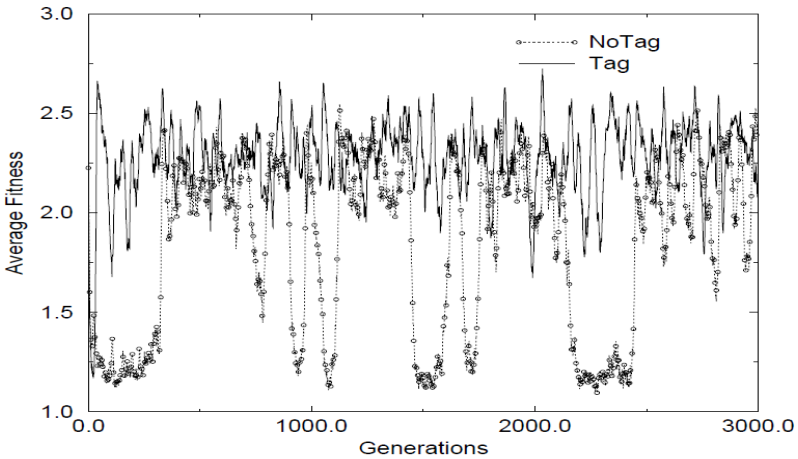


Fig 4. Factors affecting the efficiency of a tag-based autonomous model



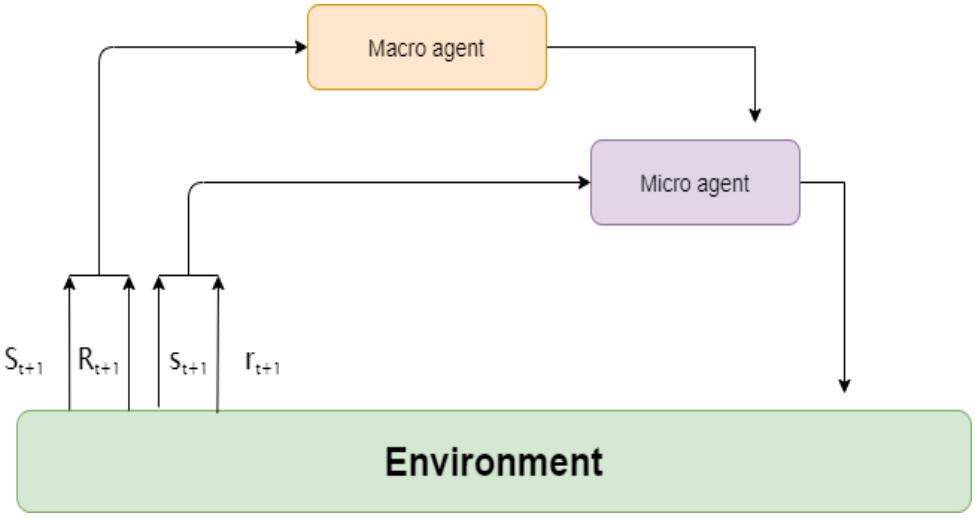
The overall tag-based model works better than that of the non-tag-based model which can be inferred from the below figure which plots average fitness over generation taken from the work of Rick L.Riolo [9]

Fig 5. Fitness vs generation for a tag-based model for Iterated Prisoner's Dilemma problem

Leading agents to mutual agreements:
Another important aspect for a multi-agent system is making an agreement among them for the upliftment of the whole system, as these agents are self-motivated and interested it is important to show them ways for mutual benefits. Auctions are the most commonly used mechanism for mutual agreement explained in Wooldridge [10] where interaction takes between an agent (auctioneer) and a group of agents (bidders) where bids may use an open cry or sealed bid method working in one shot, ascending or descending way for bidding.' It is important for negotiation standards to be stable, simple, efficient, distributed, and symmetric to make agents reach common interests.

Application of Reinforcement Learning on our financial systems:

Describing the financial MDP:
For every MDP we need to define its state space (states in which agents can be), action space (different actions that agent can take), and reward space (reward agent gets at each state taking particular action). In order for our financial trading use case



Level of Agents:
We will have agents as macro agents (that deal with the buy, sell, or gold of assets) and micro agents (to make the decision of where to place the order). The action output of the macro agent will be input for the micro agent which will finally place an order in the limit book orders. Both the agents work in an optimistic assumption were both works to maximize the overall reward

Fig 6. The multi-agent reinforcement learning framework for financial use case using macro and micro agent

State-space: We can build a state space consisting of technical indicators such as MACD (Moving average convergence divergence (Anghel 2015 [15])), RSI (Relative strength index (Bhargavi, Gumparthi, and Anith 2017 [16])), Williams %R (a momentum indicator invented by Larry Williams, Dahlquist, 2011), Weighted Bar direction (a parameter that tells us the direction and importance of the candlestick (William and Jafari 2011 [17]) formed by assigning weights) and previous day High – Low range. These indicators are chosen because of their simplicity and popularity.
The mathematical way to represent our position state will be [A, B, C] where A represents the number of contacts already brought, B is the number of contacts already sold and C is the corresponding loss at the current position.

Action space: The action space for our use case is small, containing only three actions represented as 0 for hold assets, 1 to buy the asset, and 2 for sell. The choice of actions belongs to state A = {buy, sell, hold} also known as asset trading signal.

Reward space: As our aim is to maximize our return, so our rewards will be proportionate to profits at a particular state. The rewards are defined as below with p_t as the close price of the asset at time t and β as commission.

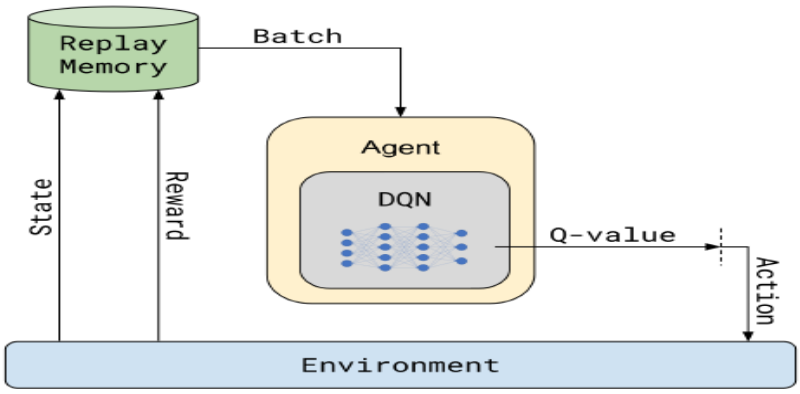
$$r_t(s_t,t_t) = 100*(\Sigma\ v_t\ /v_{t-1}\ -1\ -\beta)$$

Learning Algorithm: Q Learning

In this report, we will use a famous temporal difference algorithm q learning which is a model-free algorithm to learn the optimal policy ∏ : S→ A that maps from state to actions. Here Q learning is preferred over SARSA due to the fact that the Q learning algorithm learns optimal policy while SARSA learns near-optimal policy and also avoids high risk which is crucial for our case. The q values will be updated using bellow update:

$$Q_{t+1}(s,a) = Q_t(s,a) + \alpha[r + \gamma*max_a(Q_t(s',a)) - Q_t(s,a)]$$

For exploitation-exploration trade off, we will use a decaying e-greedy policy where the agent takes random actions with probability e and takes policy actions with probability 1-e. Also, e decays with time.



Working of macro and micro agents:

For the macro agent, the state space at time t consists of historic price data from t-h to t where h can be considered as a hyperparameter representing how far we need to go back in history to predict the price at time t. Algorithm for the macro agent reward function taken from Yagna [2] below.

Fig 7.The working of macro-agent training. Here the neural net is used to learn optimal policy

```
if action = hold then
        reward ← 0
else if action = buy then
        Append current open price to current assets list
        reward ← 0
else if action = sell then
        if there are no assets to sell then
        reward ← 1
else
        reward ← sell off all assets from current assets list and determine profit based on current open price
end
Clip Rewards
```

Finally, the micro agent determines the action to place order in the limit book. The quantity and side of the order book is determined by the data provided by the macro agent. Here order book represents a data structure that holds all bids at ask price/volume value[2]. Also overall state space for macro agents is the historic price of stock and the quantitative measures we decided before

Overall working of pipeline with macro and micro agent:

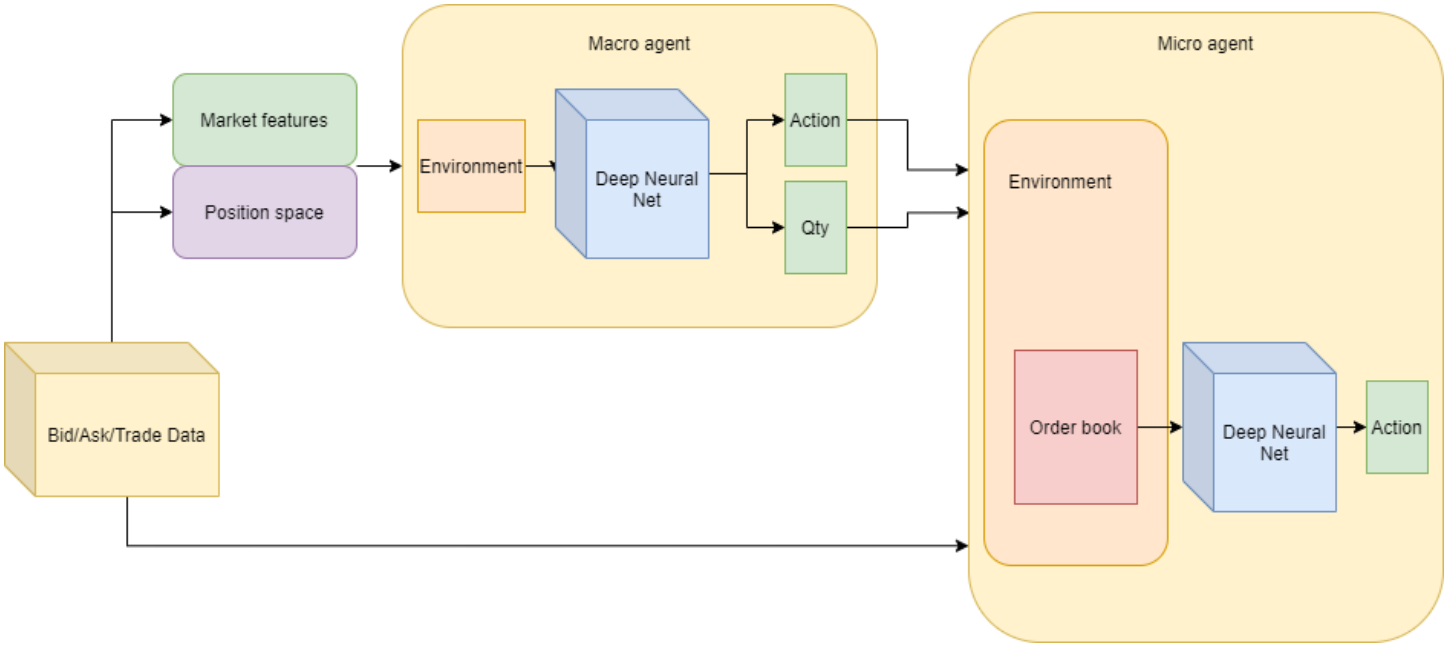


Fig 8 . Working if overall pipeline involving macro and micro agent for financial use case

How intelligent agent approach will add value to the company:

- Our current approach to supervised learning is limited due to its needs for labeled data, which is limited and costly for our domain. The reinforcement learning agent approach will help us fight this problem. Using algorithms like Q learning will lead to guaranteed convergence towards optimum where exploration strategy can speed up the convergence. The cost of labeled data helps us reduce the investment amount and thus increase our profits with algorithms with better precision.
- Q learning will not require a state-action table for all the states which is not possible for our case, it will perform function approximation i.e. using the neural net which can give better accuracy and precision than our current approach.
- Using Q learning, we will get all benefits of temporal difference prediction methods such as no requirement of a complete model of the environment.
- Using an approach where we can use both our older approaches and new approaches involving RL agents can make our system overall more robust. We can name this a combination of Inductive learning and reinforcement-based learning approach which will be much more explainable than that of using a single RL-based approach.
- RL agents are explorative in nature, which can be beneficial for our use case. Where the system can try to find new approaches to maximize reward i.e. profits. An appropriate e value for e-greedy policy helps our agent learn exploratory action, a suitable e value can be 10 to 15 percent.
- As our algorithm is considering profits as rewards, our overall approach will be profit centric which is the most crucial requirement especially in our domain of finance and can lead to better policies leading high profits. It will make our system more goal oriented.
- Our current approaches are not using neural networks and it's high time we start to integrate these advanced models in our system as they are better at learning complex functions as compared to normal machine learning algorithms.
- Using our agent approach over existing supervised learning approach will make it more adaptable or a need to retrain as it will adapt to changes on the fly whis will save us the retraining cost required for

supervised learning approach and save both money and time. Also this will make our system learn online and in real time.

- Our new approach of agents will be less prone to bias which is very common in supervised learning as supervised approach only learns from data which can itself have some kind of bias.
- Even though the RL agent approach is an innovative and value-adding method we need to consider that it may take a long time to converge, also the learning results may not be transparent.

Conclusion:

A multi agent reinforcement learning approach seems perfect to implement in current space considering the cost of labelled dataset and need of an autonomous system that can adapt on the fly without the need of retraining the system. Also it looks like agents and multi-agents will be better at working with the chaotic markets that compare with the existing supervised approach if applied and implemented with appropriate hyperparameters in the described system.

References:

1. Zhenhan Huang, Fumihide Tanaka “A Modularized and Scalable Multi-Agent Reinforcement Learning-based System for Financial Portfolio Management”
2. Yagna Patel “Optimizing Market Making using Multi-Agent Reinforcement Learning”
3. Russel S, Norvig P. “Artificial Intelligence—A Modern Approach”. 3rd ed. England: Pearson Education Limited
4. Macal CM. “Everything you need to know about agent-based modeling and simulation. Journal of Simulation.”
5. Stone P, Veloso M. “Multiagent systems: A survey from a machine learning perspective. Autonomous Robots.”
6. Nwana H. “Software agents: An overview. Knowledge Engineering Review. 1996”
7. Enda Howley, Colm O’Riordan “The Emergence of Cooperation among Agents using Simple Fixed Bias Tagging”
8. Martin A.Nowak, Robert M.May “The spatial dilemmas of evolution”
9. Rick L.Riolo “The effects of Tag-Mediated Selection of Partners in Evolving Populations Playing the Iterated Prisoner’s Dilemma”
10. Micheal Wooldridge “An introduction to Multi-Agent Systems”
11. Souradeep Chakraborty “Capturing Financial markets to apply Deep Reinforcement Learning”
12. Yagna Patel “Optimizing Market Making using Multi-Agent Reinforcement Learning”
13. Zhenhan Huang, Fumihide Tanaka “A Modularized and Scalable Multi-Agent Reinforcement Learning-based System for Financial Portfolio Management”
14. Yuh-Jong Hu and Shang-Jen Lin “Deep Reinforcement Learning for Optimizing Finance Portfolio Management”
15. Anghel, Gabriel Dan I. 2015. “Stock Market Efficiency and the MACD. Evidence from Countries around the World.”
16. Bhargavi, R., Srinivas Gumparthi, and R. Anith. 2017. “Relative Strength Index for Developing Effective Trading Strategies in Constructing Optimal Portfolio.” International Journal of Applied Engineering Research 12 (19): 8926–36.
17. William, Ron, and Sheba Jafari. 2011. “Candlestick Analysis,” no. September.
18. Y. Li, “Deep reinforcement learning: an overview”
19. B. J. Heaton et al., “Deep learning in finance”
20. Prakhar Ganesh, Puneet Rakheja. “Deep Reinforcement Learning in High Frequency Trading.”