International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Stock Prediction on Reinforcement Learning Policies

[1]P. A. Patil, [2]Sonali Bodake

[1,2]Department of IT Engineering, AISSMS IOIT, Pune, India.

Abstract—

Stock market is well known to all. It is an option for investment and trading. But it is a really difficult thing to predict the future stock price. It is helpful for the investors and traders to know the future price, so that they can enter and exit the market at the right time and right price. The automation of profit generation in the stock market is possible using DRL, by combining the financial assets price ‘‘prediction’’ step and the ‘‘allocation’’ step of the portfolio in one unified process to produce fully autonomous systems capable of interacting with their environment to make optimal decisions through trial and error. This work represents a DRL model to generate profitable trades in the stock market, effectively overcoming the limitations of supervised learning approaches. Financial trading is about buying, holding, and selling securities in the hope of making a profit. Automation is a trending area in the engineering domain that can help maximize the outcome of interest. Machine learning approaches like reinforcement learning have a great potential to solve automation in certain business domains, thereby maximizing the work outcome. Reinforcement learning has the sole objective of attaining maximum profit or reward in the given environment where the agent acts. Hence, the proposed work deals with leveraging the state of the art Actor-Critic Reinforcement learning algorithms .

Keywords—Deep reinforcement algorithm, Stock, Purchase, sold, profit, loss.

I. INTRODUCTION

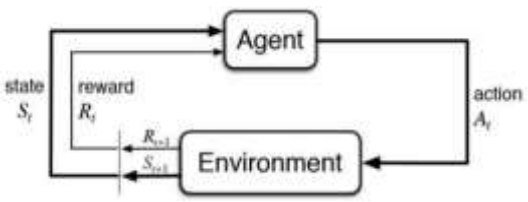
Stock trading is the process of buying and selling stock to obtain investment profit. The key to stock trading is to make the right trading decision at the right times. stock price fluctuates greatly when they use some specific strategy like, static strategy or dynamic strategy. Now days most of the companies implemented dynamic trading strategies which are based on deep reinforcement learning algorithm.Reinforcement learning is process in which an agent learns from environment and make decisions through trial and error. Deep Reinforcement Learning approach that combines technical indicators with sentiment analysis to find an optimal trading policy for assets in the stock market. RL algorithms are classified based on how to represent and train the agent into three main approaches:Critic-only approach, Actor-only approach. Actor-Critic approach.

II. OBJECTIVE

The prime objective of any investor when investing in any financial market is to minimize the risk involved in the trading process and maximize the profits generated. In supervised learning, a decision is made on the basis of the input provided at the beginning, whereas reinforcement learning gives ample ways to make decisions sequentially.Every decision making is independent of each other in supervised learning so that labels are provided for each decision, in contrast to that, in reinforcement learning every decision is dependent on other entities, in accordance with that labels are designed to all the dependent decisions.

DRL in stock market taking your client’s assets, putting it into stocks, and managing it on a continuous basis to help the client achieve their financial goals. With the help of Deep Policy Network Reinforcement Learning, the allocation of assets can be optimized over time.In supervised learning we can’t give proper guidance to the client for releted best stock, to overcome this we use DRL where we recommendation systems based on reinforcement learning techniques can be a gamechanger. These systems can help in recommending the right stocks to users while trading.

III. METHODOLOGY



A. Steps of Deep reinforcement learning Modelling

1.Create the environment: First you need to define the environment within which the reinforcement learning agent operates, including the interface between agent and environment. The environment can be either a simulation model, or a real physical system, but simulated environments are usually a good first step since they are safer and allow experimentation.

2. Define the reward: Next, specify the reward signal that the agent uses to measure its performance against the task goals and how this signal is calculated from the environment. Reward shaping can be tricky and may require a few iterations to get it right.

3. Create the agent: Then you create the agent, which consists of the policy and the reinforcement learning training algorithm.

4. Train and validate the agent: Set up training options (like stopping criteria) and train the agent to tune the policy. Make sure to validate the trained policy after training ends. If necessary, revisit design choices like the reward signal and policy architecture and train again. Reinforcement learning is generally known to be sample inefficient; training can take anywhere from minutes to days depending on the application.

5. Deploy the policy: Deploy the trained policy representation using, for example, generated C/C++ code. At this point, the policy is a standalone decision-making system. training an agent using reinforcement learning is an iterative process. Decisions and results in later stages can require you to return to an earlier stage in the learning workflow. For example, if the training process does not converge to an optimal policy within a reasonable amount of time, you may have to update any of the following before retraining the agent.

- Reinforcement learning algorithm configuration
- Policy representation
- Reward signal definition,
- Action and observation signals
- Environment dynamics

B. Technology and analytical work/

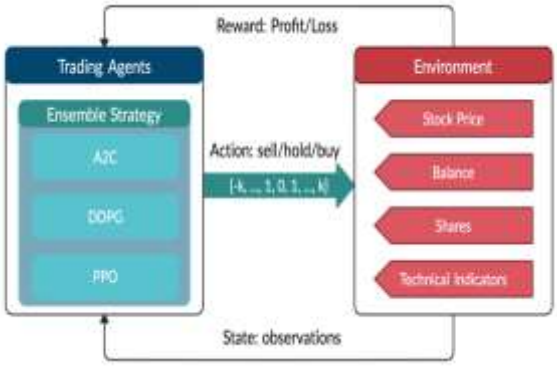
MDP in reinforcement learning:

Markov Decision Processes (MDP) is used to model stochastic processes containing random variables,transitioning from one state to another depending on certain assumptions and definite probabilistic rules. MDPs are a perfect mathematical framework to describe the reinforcement learning problem. In this framework, researchers call 106 the learner or decision maker the agent and the surrounding which the agent interacts with (comprising everything outside 108 the agent) the environment. The learning process ensues from the agent-environment interaction in MDP, at each time step t ∈ {1, 2, 3, . . . , T } the agent receives some representation (information) of its current state from the environmentst ∈ S, and on that basis selects an action at ∈ A to perform. One step later, due to its action, the agent finds itself in a new state, and the environment returns a reward Rt+1 ∈ R to the agent as a feedback of its action's quality.

Bellman equations:

Bellman equation value functions are being used by almost all RL methods to estimate how good (in terms of expected return) it is for 131 the agent to be in a given state or to perform an action in a given state. This evaluation is being made based on the future expected sum of rewards. Accordingly, value functions are determined with respect to the future actions the agent will take. We call a particular way of acting a Policy (π) which is a function that maps from the environment's states to probabilities of selecting each possible action. Bellman equations are the fundamental property of value functions used in dynamic programming as well as in reinforcement learning to solve MDPs, and they are essential to understand how many RL algorithms work. Bellman equation states that the value function of state s (Vπ (s)) can be calculated by finding the sum over all possibilities of expected returns, weighting each by its probability of occurring a policy.

C. Architecture of DRL in stock trading./



DRL uses a reward function to optimize future rewards, in contrast to an regression/classification model that predicts the probability of future outcomes.The rationale of using DRL for stock trading. The goal of stock trading is to maximize returns, while avoiding risks.The concept of reinforcement learning can be applied to the stock price prediction for a specific stock as it uses the same fundamentals of requiring lesser historical data, working in an agent-based system to predict higher returns based on the current environment. Q-learning is a model-free reinforcement learning algorithm to learn the quality of actions telling an agent what action to take under what circumstances. Q-learning finds an optimal policy in the sense of maximizing the expected value of the total reward over any successive steps, starting from the current state.

D. Approaches in Deep reinforcement learning /

Critic-Only Approach :

The critic-only learning approach, which is the most common, solves a discrete action space problem using, for example, Q-learning, Deep Q-learning (DQN) and its improvements, and trains an agent on a single stock or asset. **The idea of the critic-only approach** is to use a **Q-value function** to learn the optimal action-selection policy that maximizes the expected future reward given the current state. Instead of calculating a state-action value table, **DQN minimizes** the mean squared error between the target Q-values, and uses a neural network to perform function approximation. The major limitation of the critic-only approach is that it only works with discrete and finite state and action spaces, which is not practical for a large portfolio of stocks, since the prices are of course continuous.

Q-learning: is a value-based Reinforcement Learning algorithm that is used to find the optimal action-selection policy using a Q function.

DQN: In deep Q-learning, we use a neural network to approximate the Q-value function. The state is given as the input and the Q-value of allowed actions is the predicted output.

Actor-Only Approach :

The idea here is that the agent directly learns the optimal policy itself. Instead of having a neural network to learn the Q-value, the neural network learns the policy. The policy is a probability distribution that is essentially a strategy for a given state, namely the likelihood to take an allowed action. The actor-only approach can handle the continuous action space environments.

Policy Gradient: aims to maximize the expected total rewards by directly learns the optimal policy itself.

Actor-Critic Approach :

The actor-critic approach has been recently applied in finance. The idea is to **simultaneously update the actor network that represents the policy, and the critic network that represents the value function.** The critic estimates the value function, while the actor updates the policy probability distribution guided by the critic with policy gradients. Over time, the actor learns to take better actions and the critic gets better at evaluating those actions. The actor-critic approach has proven to be able to learn and adapt to large and complex environments, and has been used to play popular video games, such as Doom. Thus, the actor-critic approach fits well in trading with a large stock portfolio.

A2C: A2C is a typical actor-critic algorithm. A2C uses copies of the same agent working in parallel to update gradients with different data samples. Each agent works independently to interact with the same environment.

PPO: PPO is introduced to control the policy gradient update and ensure that the new policy will not be too different from the previous one.

DDPG: DDPG combines the frameworks of both Q-learning and policy gradient, and uses

IV. CONCLUSION AND FUTURE SCOPE

We explored the potential of using an Actor-Critic algorithm to solve the portfolio allocation problem.Deep Reinforcement Learning approach that combines technical indicators with sentiment analysis to find an optimal trading policy for assets in the stock market.financial data is highly time-dependent (function of time),making it a perfect fit for Markov Decision Processes (MDP), which is the core process of solving RL problems.We define our reward function as the change of portfolio value when action a is taken at state s and arriving at new state s+1.Actor-only , Critic-only , Actor-Critic approaches used for minimizes the error and improve the performance. We use deep reinforcement learning approaches in stock prediction for maximizing profit with minimum capital investments.

V. REFERENCES

[1] Avraam Tsantekidis & Nikolaos Passalis., “Price Trailing for Financial Trading Using Deep Reinforcement Learning", IEEE, ISSN:3251-6114 , Volume-32, Issue-7, July-2021.

[2] Taylan Kabbani , Ekrem Duman,“Deep Reinforcement Learning Approach for Trading Automation in the Stock Market”, IEEE Access, ISSN:2169-3536, Volume-10, Sep-2022.

[3] Lixin Ma , Yang Lu, "Application of a deep reinforcement learning method in financial market", International Conference on Measuring Technology and Mechatronics Automation, ICMTMA, ISSN: 2157-1473, Oct-2019.

[4] Iure V. Brandão, Bruno J. G. Praciano, Rafael T. de Sousa," Decision support framework for the stock market using deep reinforcement learning", 2020 Workshop on Communication Networks and Power Systems,WCNPS, INSPEC Accession Number: 20227127,Nov-2020