Predicting Stock Price using Actor-critic methods
&
Time-series Analysis

Sivashanmugam Saravanan Omprakash

Department of Mechanical & Aerospace Engineering

University at Buffalo (SUNY)

Buffalo, NY 14260

saravan3@buffalo.edu

Abstract

For a society to thrive, a strong financial system is crucial. With COVID sending people out of job or making people work from their homes, there has been a surge in people looking for investing in stock markets as a secondary source of income. Adding to this there are many companies adopting ML methods to ease people into the world of stocks & bonds. For these companies, time-series analyzing models are crucial for their operations. So, the project is to focus on combining ML models for algorithmic trading (time-series analysis) & automating trading process (actor-critic method), which can predict the price of stocks based on real life data.

1. Premise

The aim of the forthcoming work is to combine the actor-critic reinforcement learning ,methods & time-series analysis to predict the best policy based on the prevailing stock price. The interacting networks to be implemented, the actor part, will compute the best action (i.e.) the policy that refers to probability distribution over actions. And the second part, the critic part, will evaluate the policy computed by the critic.

The actor will predict one of the following actions: Buy, Wait, & Sell. The actions will be selected based on the expected profit & the actual stock price.

2. Dataset

The dataset chosen is the S&P 500 (Standard & Poor's). It is one of the commonly followed equity (captalization-weighted) indices as it measures the performance of the listed top 500 companies in the United States.

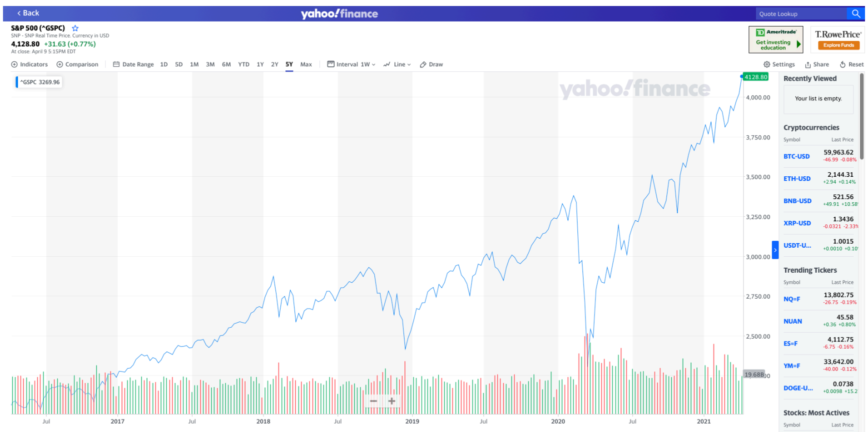

Figure 1: Standard & Poor's 5-year data

For the purpose of this project, the data containing stock prices for 4,086 days is split 85-15 to train & test the model. The training data contains stock prices over the period of 1st January 2005 to 18th October 2018 (i.e.) 3,473 days of data. Meanwhile, the testing data contains stock prices from 19th October 2018 to 26th March 2021 (i.e.) 613 days.

The trend of the prices over the year looks like the following.

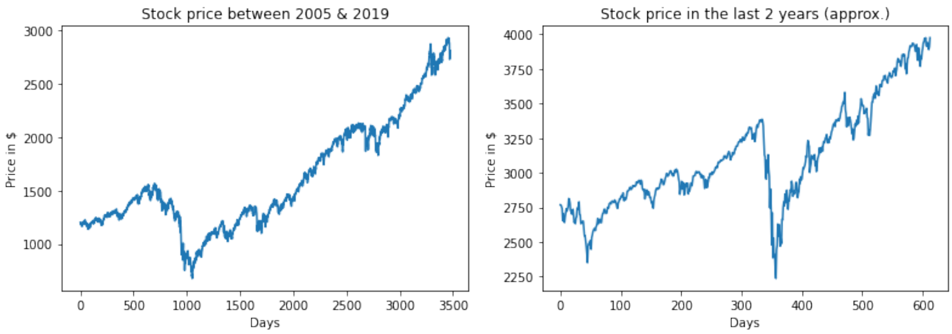

Figure 2 (a) & (b): Stock price of S&P 500 between 2005 & 2021

3. Solution

In this work, an actor-critic reinforcement learning algorithm with infused time-series helps us to predict the best action (i.e.) Buy, Wait, or Sell, based on the prevailing stock price. So in this case, there will be 2 networks interacting with each other.

The purpose of the actor network is to predict the best possible action/policy (i.e.) the distribution of probability over all available actions. And the purpose of critic network is to perform evaluation on the action chosen by the actor for a given state. The objective is to reduce the TD error.

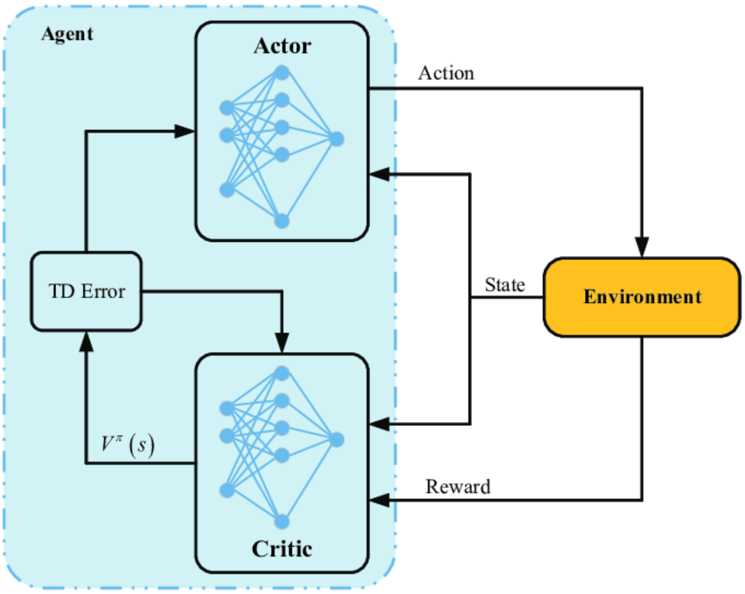

Figure 3: Actor-Critic Deep Reinforcement Learning structure

Similar to traditional deep-Q network, the value of taking the action for a given state is returned. The critic computes the approximation which will b used to update the actor weights along its gradients. Then these 2 networks are trained simultaneously, which means that the critic gets better at estimating the Q-value, while the actor learns to predict better policy.

The first step to this approach is not just the AC method, but to prepare the data. In this step, the values are reformatted to our necessity. For eg, the date is split into 3 columns, one each for day, month, & year. The aim is to get the agent to learn the pattern in time series.

When compared to the Atari implementations, the data is now our environment. The state will be a window of data to give context to each tilmestep we perform. By default, the window size if set to 20, which means that for every time step 20 records from the data is sent to the network.

Adding a soft update function to the agent will control the drastic model changes which occurs as a result of choosing the max Q-value over and over again. Also, the random sampling of experiences will break the cohesiveness of the data and results in more effective learning. This further results in better convergence behavior when training a function approximator.

4. Observation

The ideal output should list the profit as a stock is bought and sold. Also, the agent should notice the pattern in price, like for eg. price hike during holidays. But this implementation is working as expected and still needs some work to pass data more clearly and come to calculate the profit over a said period. It is suspected that the update function is not working as expected and hence the action selected is always sell at the slightest of decrease in the price.

5. References

- Reinforcement Learning - An Introduction by Richard S. Sutton & Andrew G. Barto [Chapter].
- Berkley's Deep RL Course - Actor-critic algorithms.
- Standard & Poor's historical 10-year dataset.