Stock Price Prediction using Actor-Critic method & Time series analysis

CSE 546 - Reinforcement Learning

University at Buffalo The State University of New York

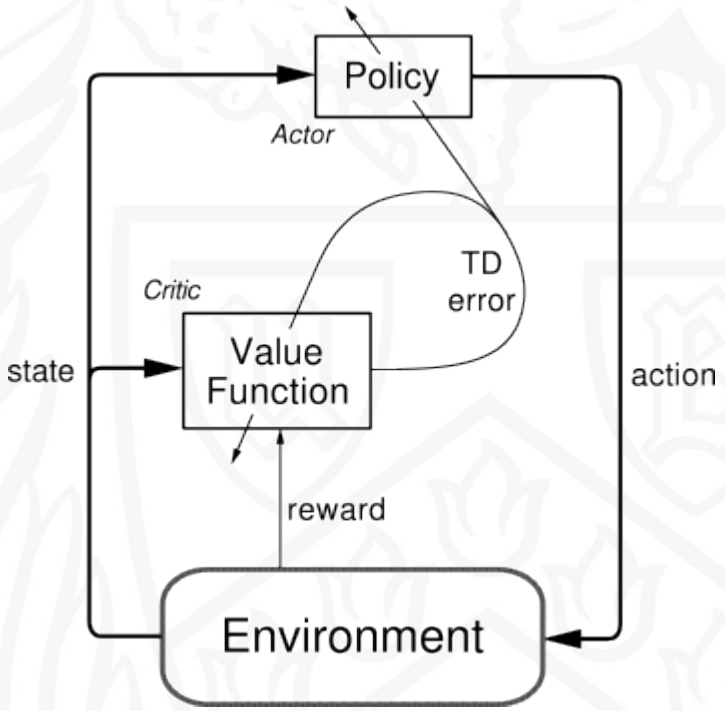
The What

Time-series analysis

- learn time-dependent patterns across multiples models
- to interpret a phenomenon, identifying the components of a trend, cyclicity, seasonality and to predict its future values
- tools for classification, clustering, forecasting, and anomaly detection depend upon real-world business applications

Actor-critic method

- Policy structure is the actor and it selects actions
- Critic estimates Value function
- A critic must learn & critique the current policy of the actor; TD error is used to critique
- Actor & critic learns simultaneously

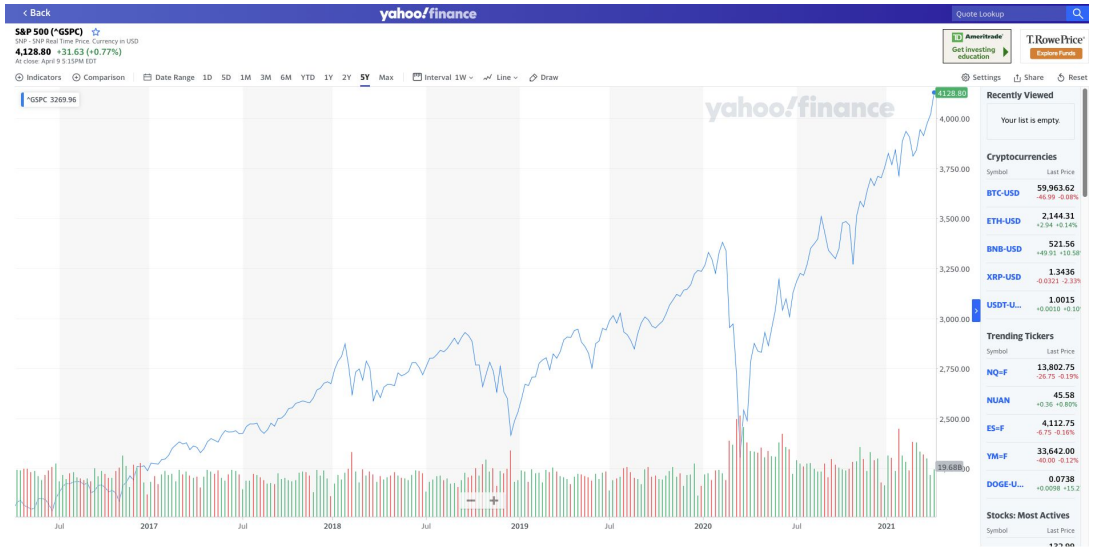


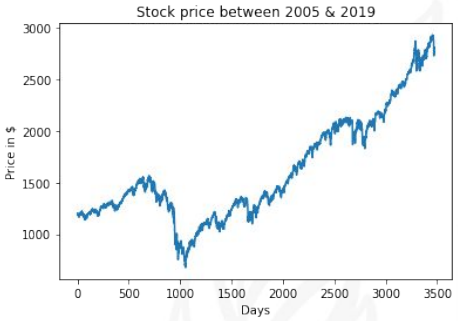
The Why

- Pandemic encouraged people to have a second stream of income
- Not all understand the workings behind stock market
- Plain curiosity!

The Data

Standard & Poor 500 (S&P 500)



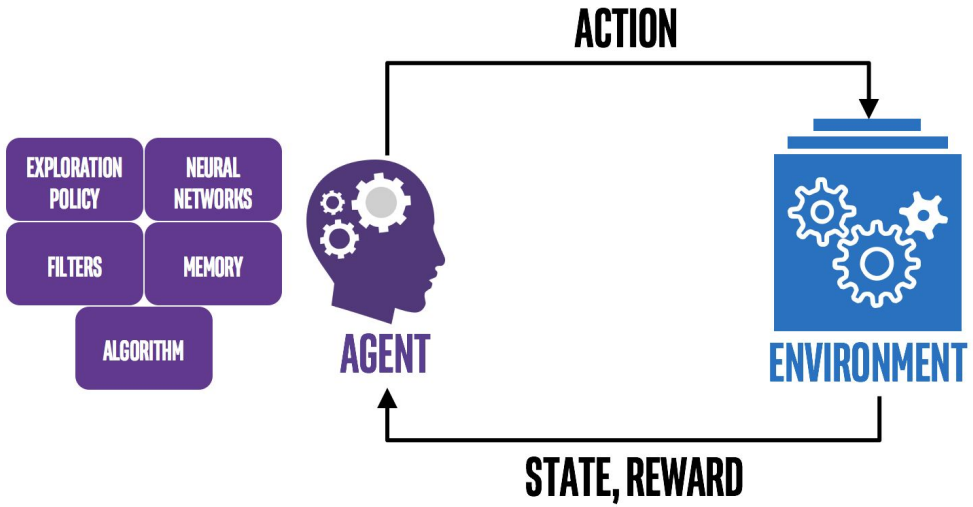


Training split
(85% of historical data)

Testing split
(15% of historical data)

Source: https://finance.yahoo.com/quote/%5EGSPC/history/

The Gamification



S&P 500 - Jan 05 to Mar 21.csv | Preview 'S&P 500 - Jan 05 to Mar 21.csv'

26-Mar-21	3,917.12	3,978.19	3,917.12	3,974.54	3,974.54	5,467,850,00
25-Mar-21	3879.34	3919.54	3853.5	3909.52	3909.52	4940800000
24-Mar-21	3919.93	3942.08	3889.07	3889.14	3889.14	4766990000
23-Mar-21	3937.6	3949.13	3901.57	3910.52	3910.52	4645340000
22-Mar-21	3916.48	3955.31	3914.16	3940.59	3940.59	4311380000
19-Mar-21	3913.14	3930.12	3886.75	3913.1	3913.1	7725050000
18-Mar-21	3953.5	3969.62	3910.86	3915.46	3915.46	4043170000
17-Mar-21	3949.57	3983.87	3935.74	3974.12	3974.12	4541620000
16-Mar-21	3973.59	3981.04	3953.44	3962.71	3962.71	4604870000
15-Mar-21	3942.96	3970.08	3923.54	3968.94	3968.94	4882190000
12-Mar-21	3924.52	3944.99	3915.21	3943.34	3943.34	4469240000
11-Mar-21	3915.54	3960.27	3915.54	3939.34	3939.34	5300010000
10-Mar-21	3891.99	3917.35	3885.73	3898.81	3898.81	5827250000

Environment: 15 year CSV data | **State:** Stock price window - Continuous | **Action:** Buy, Wait, & Sell - Discrete

Reward: Profit

Source:Reinforcement Learning, Part 1: A Brief Introduction | by dan lee | AI³ | Theory, Practice, Business | Medium

The Algorithm

Algorithm 1 Soft Actor-Critic

Input: initial policy parameters θ, Q-function parameters ϕ_1, ϕ_2, empty replay buffer $\mathcal{D}$
Set target parameters equal to main parameters $\phi_{\text{targ},1} \leftarrow \phi_1$, $\phi_{\text{targ},2} \leftarrow \phi_2$
repeat
  Observe state s and select action $a \sim \pi_\theta(\cdot|s)$
  Execute a in the environment
  Observe next state s', reward r, and done signal d to indicate whether s' is terminal
  Store (s, a, r, s', d) in replay buffer $\mathcal{D}$
  If s' is terminal, reset environment state.
  if it's time to update **then**
    for j in range(however many updates) **do**
      Randomly sample a batch of transitions, $B = \{(s, a, r, s', d)\}$ from $\mathcal{D}$
      Compute targets for the Q functions:

$$y(r, s', d) = r + \gamma(1 - d)\left(\min_{i=1,2} Q_{\phi_{\text{targ},i}}(s', \tilde{a}') - \alpha \log \pi_\theta(\tilde{a}'|s')\right), \quad \tilde{a}' \sim \pi_\theta(\cdot|s')$$

      Update Q-functions by one step of gradient descent using

$$\nabla_{\phi_i} \frac{1}{|B|} \sum_{(s,a,r,s',d) \in B} \left(Q_{\phi_i}(s, a) - y(r, s', d)\right)^2 \qquad \text{for } i = 1, 2$$

      Update policy by one step of gradient ascent using

$$\nabla_\theta \frac{1}{|B|} \sum_{s \in B} \left(\min_{i=1,2} Q_{\phi_i}(s, \tilde{a}_\theta(s)) - \alpha \log \pi_\theta\left(\tilde{a}_\theta(s)|\, s\right)\right),$$

      where $\tilde{a}_\theta(s)$ is a sample from $\pi_\theta(\cdot|s)$ which is differentiable wrt θ via the reparametrization trick.
      Update target networks with

$$\phi_{\text{targ},i} \leftarrow \rho \phi_{\text{targ},i} + (1 - \rho)\phi_i \qquad \text{for } i = 1, 2$$

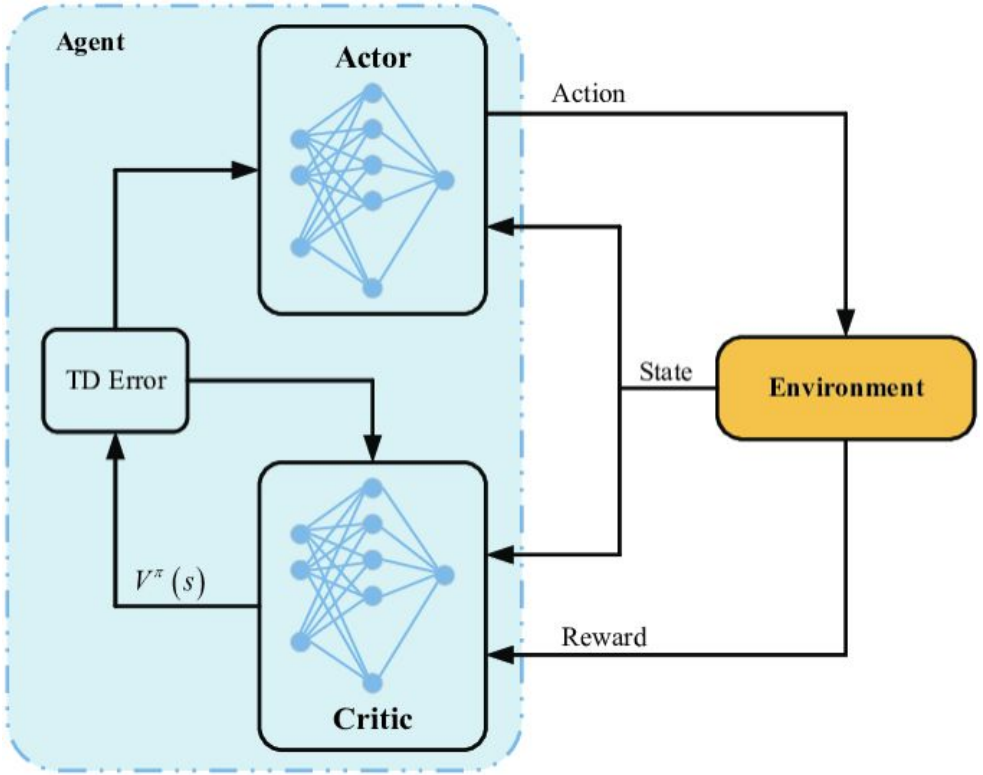
    end for
  end if
until convergence

SAC - Soft Actor Critic

- Works in discrete action space (Unlike DDPG & TD3)
- Concurrently learns a policy and two Q-functions
- Off-policy method (Prefers exploitation rather than Exploration)

Source: https://spinningup.openai.com/en/latest/algorithms/sac.html#documentation

Code structure



Key components

- Data preparation function (Environment)
- Actor & Critic network
- SAC Agent

Observations

- It's hard, but interesting
- Season dependent data not representative of real-life operations
- My shortcomings

Future scope

- Comparison with other AC DRL methods
- Adding more constraints (like news article - word vectorization) for added pattern in data

Thank you