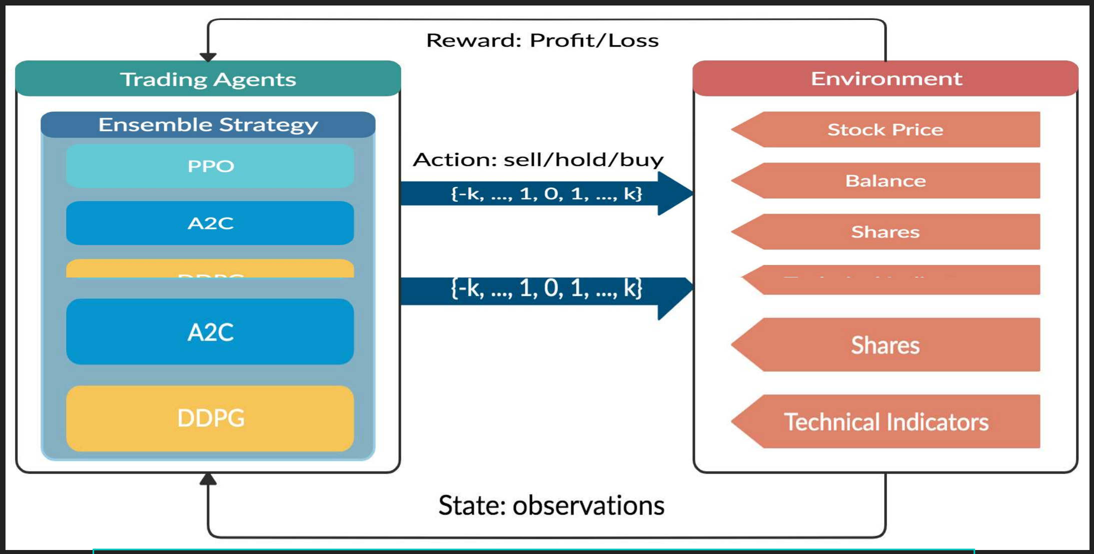
Deep Reinforcement Learning for Automated Stock Trading Ensemble Strategy

-Vishal Juneja

Introduction

The ensemble strategy inherits and integrates the best features of the three algorithms, thereby robustly adjusting to different market situations and using a load on demand technique for processing large data to avoid large memory consumption.



Trading agents and environments interacts with each other using *Action, State* and *Reward*

Markov Decision Process for Stock Trading

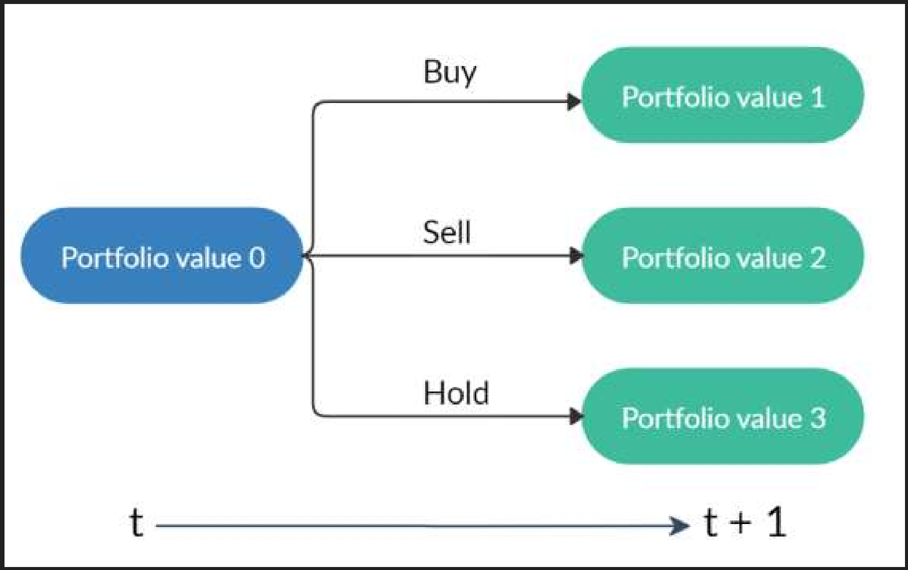
- State $s = [p, h, b]$: Vector

(Stock price, Shares, Remaining Balance)

- Action a : Vector for taking actions as

(Selling, Buying, Holding

- Reward $\mathrm{r} = (s, a, s')$: Direct reward for action from state $s \rightarrow s'$
- Policy $\pi(s)$: Probability distribution of a at s
- Q-Value $Q_\pi(s, a)$: Expected reward of action a at s following policy π



Taking action will change Portfolio

Return Maximization as Trading Goal

Stocking Constraints

- Market Liquidity: assuming that stock market will not be affected by our reinforcement trading agent
- Non Negative Balance $b \geq 0$:
- Transaction cost c_t: Transaction costs are incurred for each trade
- Risk Aversion for Market Crash:

  Turbulence index as $turbulence_t$

Maximising Reward Function

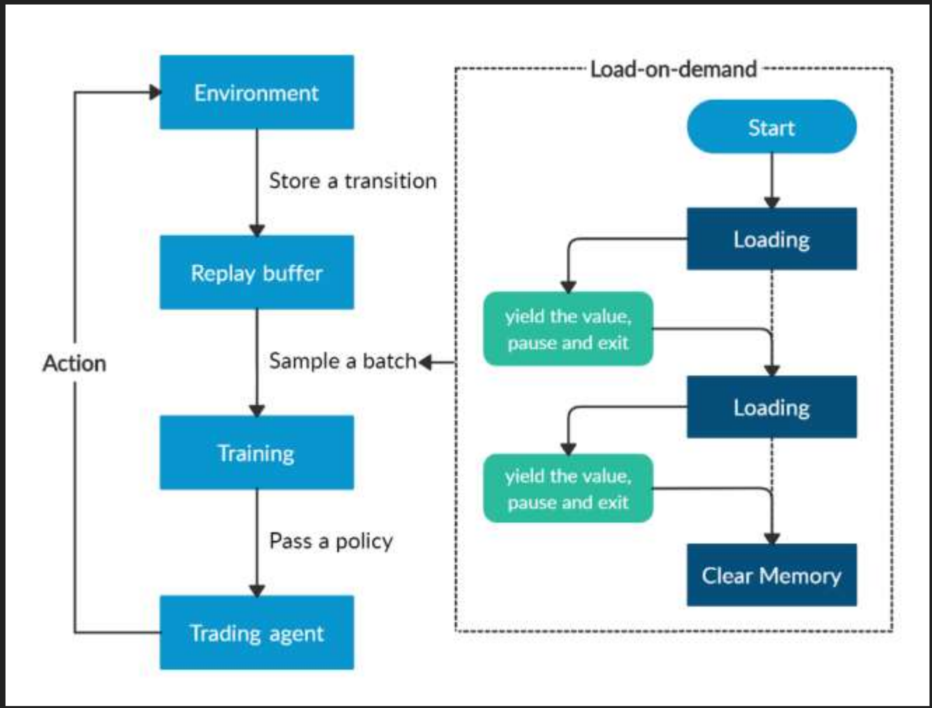
- $r = Potfolio_{t+1} - portfolio_t - c_t$

Sell all during Market Crash

- $turbulence_t$ > threshold value

Stock Trading management

- *Environment for Multiple Stocks:* A continuous action space to model the trading of multiple stocks it is assumed that the portfolio has 30 stocks in total
  1. State Space: This space is defined on components as balance, stock price, no. of stocks, MACD, RCI, CCI, ADX.
  2. Action Space: This space present is defined on the basis of number of shares k $(k < h_{max})$ to perform action of buying, selling or holding
- *Memory Management*: The load-on-demand technique does not store all results in memory, rather, it generates them on demand due to which the memory usage us reduced



Deep Learning Algorithms

Advantage Actor Critic (A2C)

- It is a typical actor-critic algorithm which utilizes an advantage function to reduce the variance of the policy gradient.
- It is a great model for stock trading because of its stability.

Proximal Policy Optimization (PPO)

- It updates and ensure that the new policy will not be too different from the previous one.
- Chosen for stock trading because it is stable, fast, and simpler to implement and tune.

Deep Deterministic Policy Gradient (DDPG)

- It encourage maximum investment return and combines the frameworks of both Q-learning and policy gradient.
- It is effective at handling continuous action space, and so it is appropriate for stock trading.

Ensemble Strategy

Step 1

- Growing window of *n* months to retrain our three agents concurrently, for the paper *n*=3

Step 2

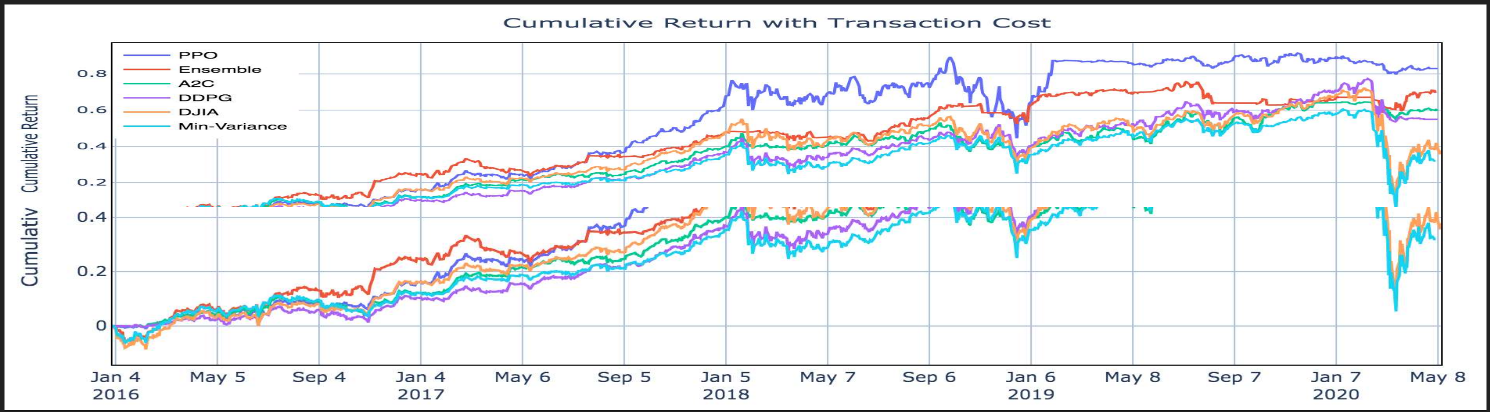
- Calculating Sharpe Ratio
- $Sharpe\ ratio = \frac{\overline{r_p} - r_f}{\sigma_p}$

$\bar{r}_p$- portfolio return, r_f- risk free return, σ_p- portfolio standard deviation

Step 3

- After the best agent is picked, it is used to predict and trade for the next quarter.
- This maximizes the returns adjusted to the increasing risk

Performance Evaluation Plots



PERFORMANCE EVALUATION COMPARISON.

(2016/01/04-2020/05/08)	Ensemble (Ours)	PPO	A2C	DDPG	Min-Variance	DJIA
Cumulative Return	70.4%	83.0%	60.0%	54.8%	31.7%	38.6%
Annual Return	13.0%	15.0%	11.4%	10.5%	6.5%	7.8%
Annual Volatility	9.7%	13.6%	10.4%	12.3%	17.8%	20.1%
Sharpe Ratio	1.30	1.10	1.12	0.87	0.45	0.47
Max Drawdown	-9.7%	-23.7%	-10.2%	-14.8%	-34.3%	-37.1%

Performance Evaluation Conclusions

- A2C agent is more adaptive to risk. It has the lowest annual volatility 10.4% and max drawdown -10.2%
- PPO agent is good at following trend and acts well in generating more returns, it has the highest annual return 15.0% and cumulative return 83.0%
- DDPG performs similar but not as good as PPO, it can be used as a complementary strategy to PPO, but its returns are not as satisfactory as other two.
- By incorporating the turbulence index, the agents are able to cut losses and successfully survive the stock market crash in March 2020.

SHARPE RATIOS OVER TIME.

Trading Quarter	PPO	A2C	DDPG	Picked Model
2016/01-2016/03	**0.06**	0.03	0.05	PPO
2016/04-2016/06	0.31	0.53	**0.61**	DDPG
2016/07-2016/09	-0.02	0.01	**0.05**	DDPG
2016/10-2016/12	**0.11**	0.01	0.09	PPO
2017/01-2017/03	**0.53**	0.44	0.13	PPO
2017/04-2017/06	0.29	**0.44**	0.12	A2C
2017/07-2017/09	**0.4**	0.32	0.15	PPO
2017/10-2017/12	-0.05	-0.04	**0.12**	DDPG
2018/01-2018/03	**0.71**	0.63	0.62	PPO
2018/04-2018/06	-0.08	-0.02	**-0.01**	DDPG
2018/07-2018/09	-0.17	**0.21**	-0.03	A2C
2018/10-2018/12	0.30	**0.48**	0.39	A2C
2019/01-2019/03	-0.26	-0.25	**-0.18**	DDPG
2019/04-2019/06	**0.38**	0.29	0.25	PPO
2019/07-2019/09	**0.53**	0.47	0.52	PPO
2019/10-2019/12	-0.22	**0.11**	-0.22	A2C
2020/01-2020/03	-0.36	**-0.13**	-0.22	A2C
2020/04-2020/05	-0.42	**-0.15**	-0.58	A2C

Results for Ensemble Strategy

- The ensemble strategy achieves a Sharpe ratio 1.30, which is much higher than the Sharpe ratio the two baselines, 0.47 for DJIA, and 0.45 for the min-variance portfolio allocation
- The ensemble strategy also outperforms A2C with a Sharpe ratio of 1.12, PPO with a Sharpe ratio of 1.10, and DDPG with a Sharpe ratio of 0.87, respectively
- Ensemble strategy outperforms the three individual algorithms, balancing risk and return under transaction costs, which makes it auto adjustable to choose for the specific market condition.

References

- Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy
- A2C Algorithm
- Proximity Policy Optimization Algorithm
- Deep Deterministic Policy Gradient algorithm