Deep reinforcement learning for dynamic strategy investment in financial markets

What is Deep Reinforcement learning ?

Deep Reinforcement Learning (DRL) is a cutting-edge technique that combines the decision-making framework of reinforcement learning (RL) with the powerful pattern recognition capabilities of deep learning (DL). In DRL,same as a traditional RL agent ,an agent learns optimal actions by interacting with an environment, receiving feedback in the form of rewards or penalties, and aiming to maximize cumulative rewards.

How its different from traditional RL agents

REINFORCEMENT LEARNING (RL) :

- *Typically uses hand-crafted features or a simple representation of the environment states.*
- *Often relies on lookup tables (e.g., Q-tables in Q-learning) to store and compute the value or policy for each state-action pair. This approach works only for small state spaces.*

DEEP REINFORCEMENT LEARNING (DRL):

- *Uses deep neural networks to automatically learn complex features from high-dimensional data, such as raw images, audio, or intricate state spaces.*
- *Replaces lookup tables with neural networks that approximate value functions, policies, or Q-functions, enabling the agent to handle continuous or massive state-action spaces.*

BEFORE MOVING ON

LET'S LOOK ON TO SOME BASIC/IMPORTANT CONCEPTS OF DRL

POLICY:

It is the agent's strategy for choosing actions based on its current state in the environment. It defines how the agent behaves to maximize its reward over time.

UPDATION

REWARD:

It is a signal that tells the agent how good or bad its action was in a particular state. It's the feedback the agent gets from the environment after taking an action, guiding it toward achieving its goal

DIFFERENCE

Q-VALUE

It represents the expected return (reward) the agent can obtain by taking a certain action from a given state and following the optimal policy thereafter.

ACTION

ENVIRONMENT :

The world in which the agent operates. It provides feedback based on the agent's actions, typically in the form of rewards.

TRANSITION

Reward is the immediate feedback received after taking an action, while quality refers to the expected long-term reward the agent will get from a state-action pair.

THE PROTAGONIST

FINANCIAL STRATEGY REINFORCEMENT LEARNING (FRSL) :-

- *It is a novel framework leveraging DRL to dynamically select and execute the most appropriate quantitative strategy from a diverse set based on real-time market conditions.*

- *This approach departs from conventional methods that depend on a fixed strategy, instead modeling the strategy selection process as a Markov Decision Process (MDP), which means ,making decisions, considering the current state, possible actions, rewards, and the transition dynamics.*

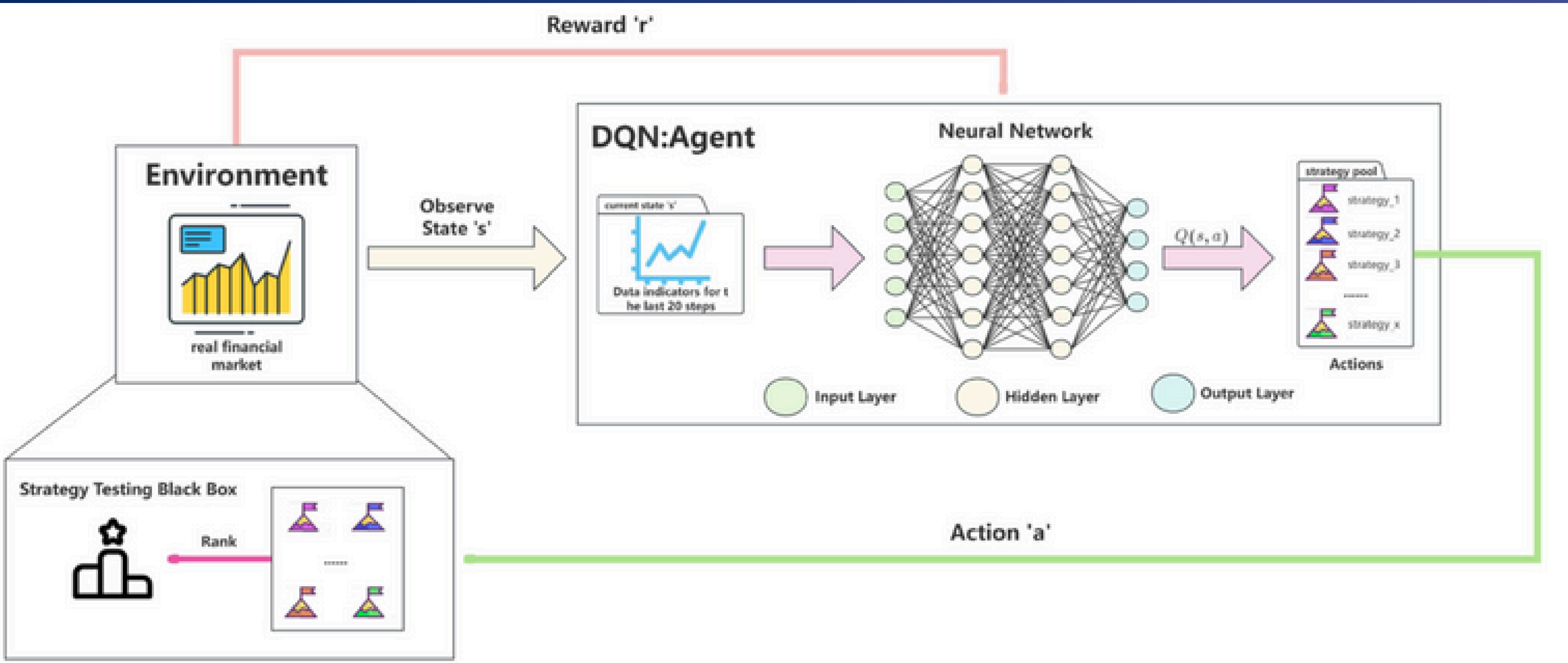
- *FSRL offers a deep reinforcement learning-based trading model with dynamic strategy-switching capabilities, incorporating a standardized financial market environment for realistic market assessments transforming quantitative trading from a multi-factor approach to a multi-strategy paradigm, offering enhanced adaptability and robustness in the face of market volatility.*

NEED FOR IT.....

The traditional models such as the Capital Asset Pricing Model (CAPM), Markowitz's Portfolio Theory, and other models have played a significant role in understanding market dynamics they exhibit notable limitations in addressing the non-linearity and complexity of financial markets. These models typically assume market equilibrium and rational investors, which do not always hold true in reality.

On the other hand, Deep Reinforcement Learning (DRL),when dealing with high-dimensional and noisy financial data,may suffer from overfitting and their learning processes can be easily disturbed by the noise in market data.. Secondly, most existing DRL methods focus on optimizing a ***single strategy****, overlooking the importance of adopting different strategies under varying market conditions.*

On the other hand FSRL's dynamic strategy-switching capability captures the ***strengths of individual strategies*** *and applies it to the market condition that well suits it and offers a robust and adaptive trading solution.*


Reward 'r'
Environment
real financial market
Observe State 's'
DQN:Agent
current state 's'
Data indicators for t he last 20 steps
Neural Network
$Q(s,a)$
strategy pool
strategy_1
strategy_2
strategy_3
strategy_x
Actions
Input Layer
Hidden Layer
Output Layer
Strategy Testing Black Box
Rank
Action 'a'

Important Functions/Classes in the model

STATE

The state function present in the financial environment class provides the state/condition of the environment and the current timestep .It provides us the OHLC DATA of the selected stock of the current step along with the values of 3 technical indicators : RSI , MACD , BOLLINGER BANDS

- Relative Strength Index (RSI) : For measuring the strength of the upcoming trend in the market by comparing recent gains and losses.
- Moving Average Convergence Divergence (MACD) : It Identifies whether a market is in an uptrend or downtrend.
- Bollinger Bands (BB) : It combines a moving average with standard deviations to create a dynamic range around the price, helping traders assess potential breakout or reversal points.

Open	High	Low	Close	Adj Close	Volume	CODE	RSI	macd	BBU_20_2.0	BBM_20_2.0	BBL_20_2.0

REWARD

*The primary objective is to rank the strategies based on key financial metrics such as the **Sharpe Ratio** (SR), **Maximum Drawdown** (MD), **Total Return** (TR), **Annualized Return** (AR), and **Annualized Volatility** (AV).*

Each strategy is scored based on the weighted combination of the financial metrics using the weights w1, w2, w3, w4, w5, which are defined as:

$$Score = w_1 \cdot SR + w_2 \cdot MD + w_3 \cdot TR + w_4 \cdot AR + w_5 \cdot AV$$

Strategies are then ranked based on their respective scores from highest to lowest.

REPLAY MEMORY

It stores past experiences or transitions encountered by the agent during interactions with the environment. Each transition is typically represented as a tuple containing :

```
replay_memory([state, action, reward, next_state, done])
```

Consecutive experiences might be highly correlated, which can lead to inefficient learning. By randomly sampling transitions from the replay memory, the agent learns from diverse experiences, improving the generalization of the learned policy.Replay memory helps the agent learn even when rewards are sparse or delayed.

The sampled transitions are used to update the Q-values in the model using the Bellman equation. This helps improve the agent's decision-making over time.

Q-VALUE

The action-value function ***Qπ (st, at)*** *is used to predict the expected cumulative reward starting from state st and taking action at under policy π at time step t. The* ***Bellman equation,*** *incorporating the discount factor γ , is used to recursively compute the expected rewards:*

$$Q_\pi(s_t, a_t) = \mathbb{E}_{s_{t+1}}\left[R(s_t, a_t, s_{t+1})\right] + \gamma \mathbb{E}_{a_{t+1}\sim\pi(s_{t+1})}\left[Q_\pi(s_{t+1}, a_{t+1})\right]$$

In the proposed model we have 2 deep neural network : The training model and the Target mode l representing Q value functions of the algorithm.

- *Training model is updated after every time step based on the reward fom the environment .*
- *Target model is updated after every fixed number of intervals and its parameters are set equal to training model*

Using two networks makes the training process smoother and helping the agent learn better over time without getting stuck in oscillations or errors of noisy financial market .

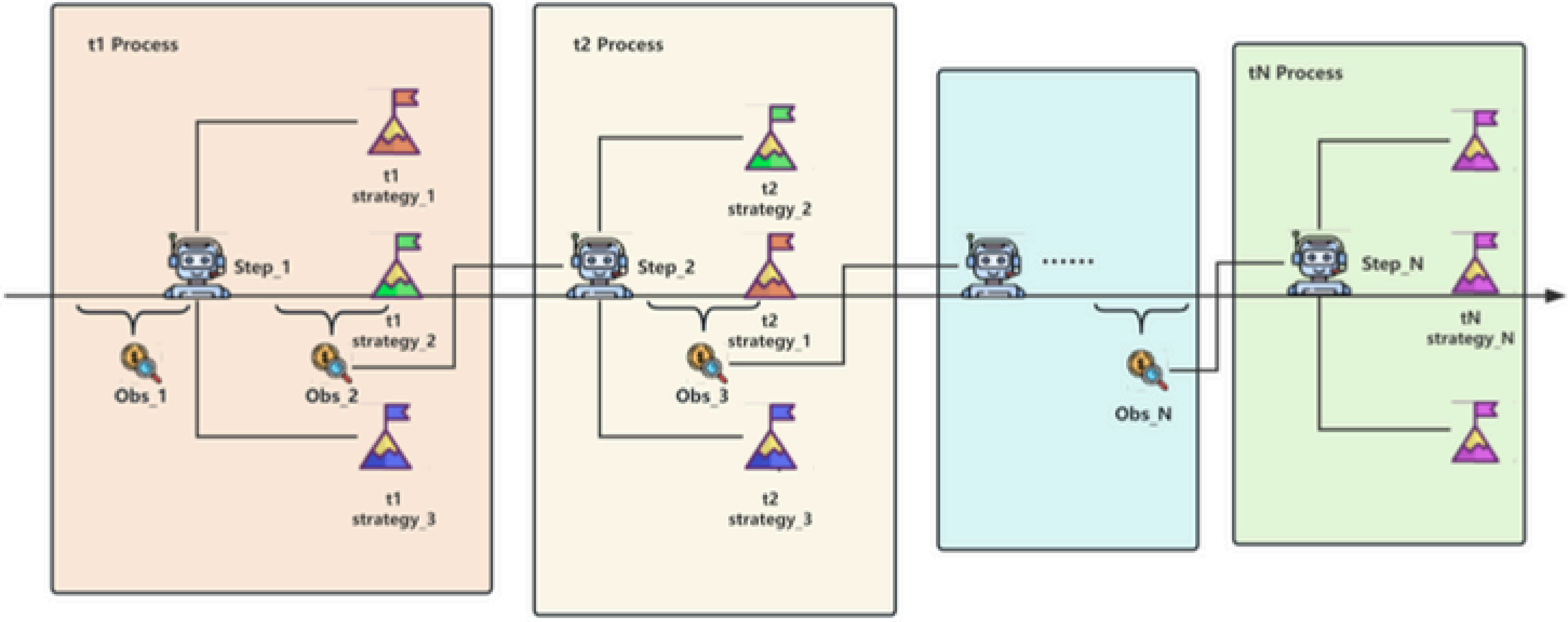
Finite transaction state machine

Finite Transaction State Machine (FTSM) to evaluate the effectiveness of different trading strategies chosen by the agent. Here is the step by step process on how it works :

Finite Transaction State Machine (FTSM) to evaluate the effectiveness of different trading strategies chosen by the agent. Here is the step by step process on how it works :

- *The agent gets observation from the environment for the current step and it predicts the best strategy based in it*
- *The FTSM model passes this strategy in the backtest environment based on the maximum number of trades and maximum steps and evaluates the performance of this strategy in comparison to others ,*
- *The model then executes its predicted strategy and recieves a reward based on the ranking of the strategies.*

FTSM-Finite transaction state machine



Obs:Observaion

Time Line:

NET WORKFLOW OF THE MODEL

Initially started taking action randomly for exploration and gradually decreased the value of epsilon for exploitation,i.e, to make the neural networks predict based on its previous state and reward combinations when an episode was close to it end .

The predicted strategy is passed on to the environment and returns the reward of the strategy chosen at that market state .

These all memories are also getting stored in the replay memory simultaneously .

After needed amount of past data with the reward for each step is calculated the neural network training begins . The training model predicts the quality values for each time step and the target model for its +1 timestep. The target q value predicted act as target for training model to achieve and on the way optimise itself

After a fixed number of steps have been completed then the target model weights are set equal to trading model weigths. Here FTSM Model is used

Now the next state is observed and the training network predicts the most suitable strategy according to it and the cycle

CONTINUES.....

RESULTS

AAPL STOCK

MAX DURING TRAINING

PORTFOLIO VALUES	START	END
FRSL	2500	4205
PPO	2500	3364
DQN	2500	4145

TESTING

PORTFOLIO VALUES	START	END
FRSL	2500	3097
PPO	2500	2834
DQN	2500	2719

Conclusion

As mentioned in the reserach paper , frsl model outperforms or equally performs popular DRL algorithms in stock trading as is a great technique using the dynamic switching strategy .

Future development

Will try to work on the model selecting the amount of stock to trade in a single step depending upon the risk factor in selecting a strategy at the current timestep .

next slide →


Thank You!