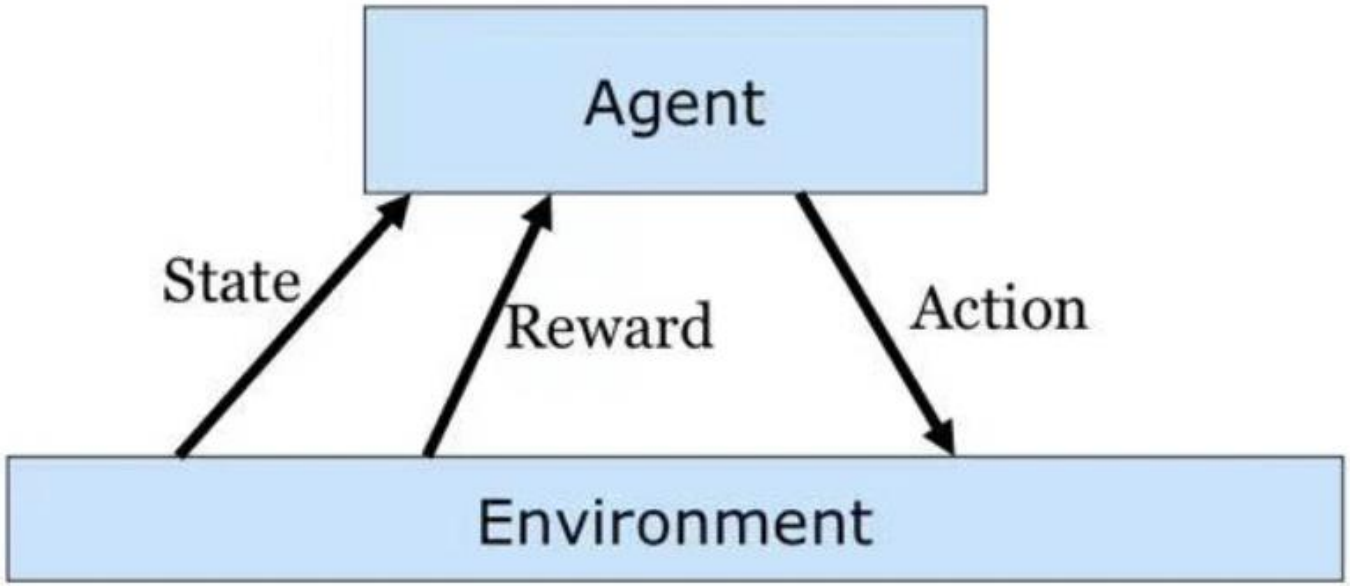
Requirements Engineering for Deep Reinforcement Learning in a Finance-Trading Application

Based on a stock-trading system

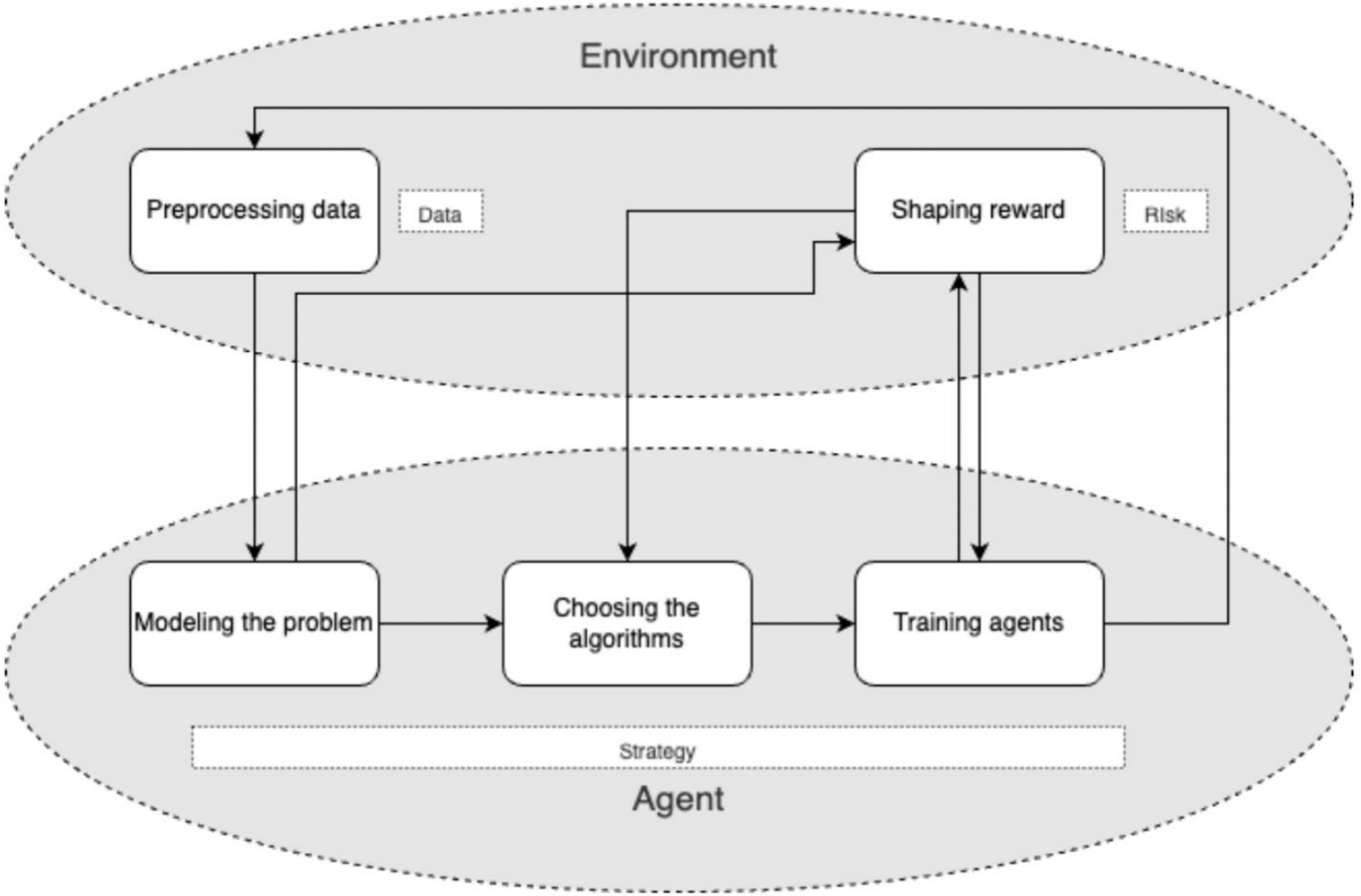
By Lufan Wang

What's reinforcement learning?



Goal: Learn to choose actions that maximize rewards

How could we use that to solve our problem?



Why is requirement engineering (RE) for reinforcement learning important?

- A little or even no awareness of a well-designed performance measures to state the good requirements can lead to a poor or even unsuccessful performance of the applications when facing new cases.
- Therefore, working with requirement engineering (RE) is important for building robust agents.

Five different steps

- Preprocessing data
- Modelling the problem
- Shaping reward
- Training agents
- Choosing the algorithms

Preprocessing data

- The most common data type in the finance: [time, open, high, low, close, volume].

- Requirements:
- Requirement 1: Store the price into the list only if the converted timestep of that price is 60*60*24 times larger than the latest price timestep in the list.
- Requirement 2: Stop storing prices only if the length of the list reaches 2000.

Modelling the problem

- Since it is a stock trading system, so we will have three actions: buy, sell or hold.

- Requirements:
- Requirement 3: Choose to buy the shares only if the action number generated by the agent is >0.
- Requirement 4: Choose to sell the shares only if the action number generated by the agent is <0.
- Requirement 5: Choose to do nothing only if the action number generated by the agent is =0.

Problems:

- The agent will return you the action number they think will you give you "highest reward". But since we are in a stock-trading system, there are more constraints that need to be considered here.

- For example:
- If the cash amount = 0, are we able to buy any shares?
- If the shares' amount we have = 0, are we able to sell any shares?

Updated the subrequirements:

- Requirement 3: Choose to buy the minimum integer amount of shares between the number a where a=(cash amount)/(stock price) and the absolute value of the action number only if the action number generated by the agent is >0.
- Requirement 4: Choose to sell the minimum integer amount of shares between the amount of shares we have and the absolute value of the action number only if the action number generated by the agent <0
- Requirement 5: Choose to do nothing only if the action number is =0

Shaping reward

- Profit = cash amount + stock price * shares amount held - initial amount

- Requirement here:
- Requirement 6: Reward used in the learning machine should be the output of the profit equation where profit is equal to the cash amount plus the stock price times the shares amount held minus the initial amount.

Training agents

- How to train our agents?
- When should we stop training?
- The requirements for training agents will be a combination of above two questions' answers.

First, when should we stop training?

- Possible subrequirements:
- Start a new training process only if the step is equal to 0
- Add one in the step counter only if we do a buy/sell/hold action
- Stop the training process only if the current step is bigger than the limit number.

Second, how to train our agents?

- Possible subrequirements:
- The action output is bigger than 0 only if the next time step price is higher than the current.
- The action output is smaller 0 only if the next time step price is lower than the current.
- The action output is equal to 0 only if the next time step price is equal to the current.

Update subrequirements:

- Requirement 7: Start a new training process only if the step is equal to 0
- Requirement 8: The action output is bigger than 0 only if the next time step price is higher than the current.
- Requirement 9: The action output is smaller 0 only if the next time step price is lower than the current.
- Requirement 10: The action output is equal to 0 only if the next time step price is equal to the current.
- Requirement 11: Add one in the step counter only if we produces an action output
- Requirement 12: Stop the training process only if the current step is bigger than the limit number.

Choosing the algorithm

- Monte Carlo Method: generate 1000 price paths
- VaR & CVaR: Worst 5% & Mean of these worst 5%

- Calculate CVaR for one agent:
- Running the trained agent on the 1000 prices path will produce a list R of size 1000 where it stores each path's reward number at the stop time of the training process
- Get the related return: related return list= 1+ R/initial cash amount
- CVaR = mean of the smallest 5% number in the related return list

Requirements:

- Requirement 13: Use Monte Carlo Method to generate 1000 prices paths with length of 3000
- Requirement 14: Calculate the CVaR value only based on this running agents' results of this 1000 paths
- Requirement 15: Choose the algorithm only if its agent has the biggest CVaR value.

Requirements:

- Requirement 1: Store the price into the list only if the converted timestep of that price is 60*60*24 times larger than the latest price timestep in the list.
- Requirement 2: Stop storing prices only if the length of the list reaches 2000.
- Requirement 3: Choose to buy the shares only if the action number generated by the agent is >0.
- Requirement 4: Choose to sell the shares only if the action number generated by the agent is <0.
- Requirement 5: Choose to do nothing only if the action number generated by the agent is =0.
- Requirement 6: Reward used in the learning machine should be the output of the profit equation where profit is equal to the initial cash amount minus the cash amount minus the stock price times the shares amount held.
- Requirement 7: Start a new training process only if the step is equal to 0
- Requirement 8: The action output is bigger than 0 only if the next time step price is higher than the current.
- Requirement 9: The action output is smaller 0 only if the next time step price is lower than the current.
- Requirement 10: The action output is equal to 0 only if the next time step price is equal to the current.
- Requirement 11: Add one in the step counter only if we produces an action output
- Requirement 12: Stop the training process only if the current step is bigger than the limit number.
- Requirement 13: Use Monte Carlo Method to generate 1000 prices paths with length of 3000
- Requirement 14: Calculate the CVaR value only based on this running agents' results of this 1000 paths
- Requirement 15: Choose the algorithm only if its agent has the biggest CVaR value.

Future works

Q&A