Artificial Intelligence and Trading

By Harvey Huang

Warm-up discussions

- How do you envision “artificial intelligence”?
- How do you envision “AI in trading (maybe finance in general)”?

Motivation

Motivation

Fundamental	Technical
• Economy & Industry (incl. upstream/downstream) • News. • Financials (EBIDTA, PE multiples, bad debt ratio etc.) • Management team. • The future!	• Market trends (Fed, SEC and News). • X min lines, candles, books. • Econometric/derivatives models. • Deep learning!

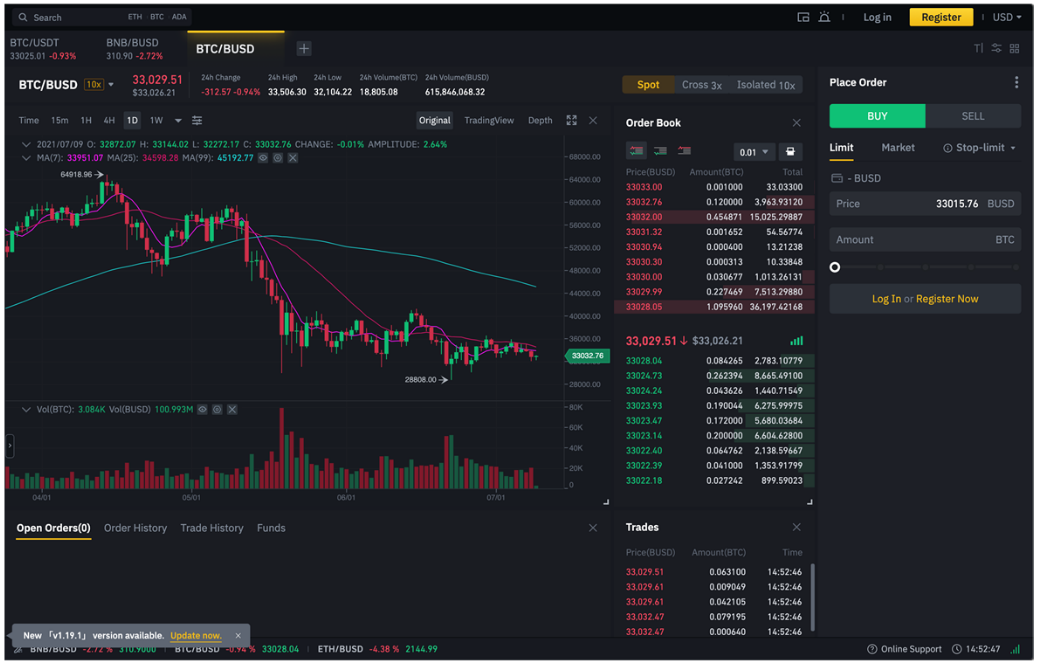
Target price: $100K / strong buy

Buy when the 5 mins line crosses the 20 mins line from below

You make some predictions … then what?

Motivation

- Conclusion: I'm going to buy BTC.
- Then you open the trading app.



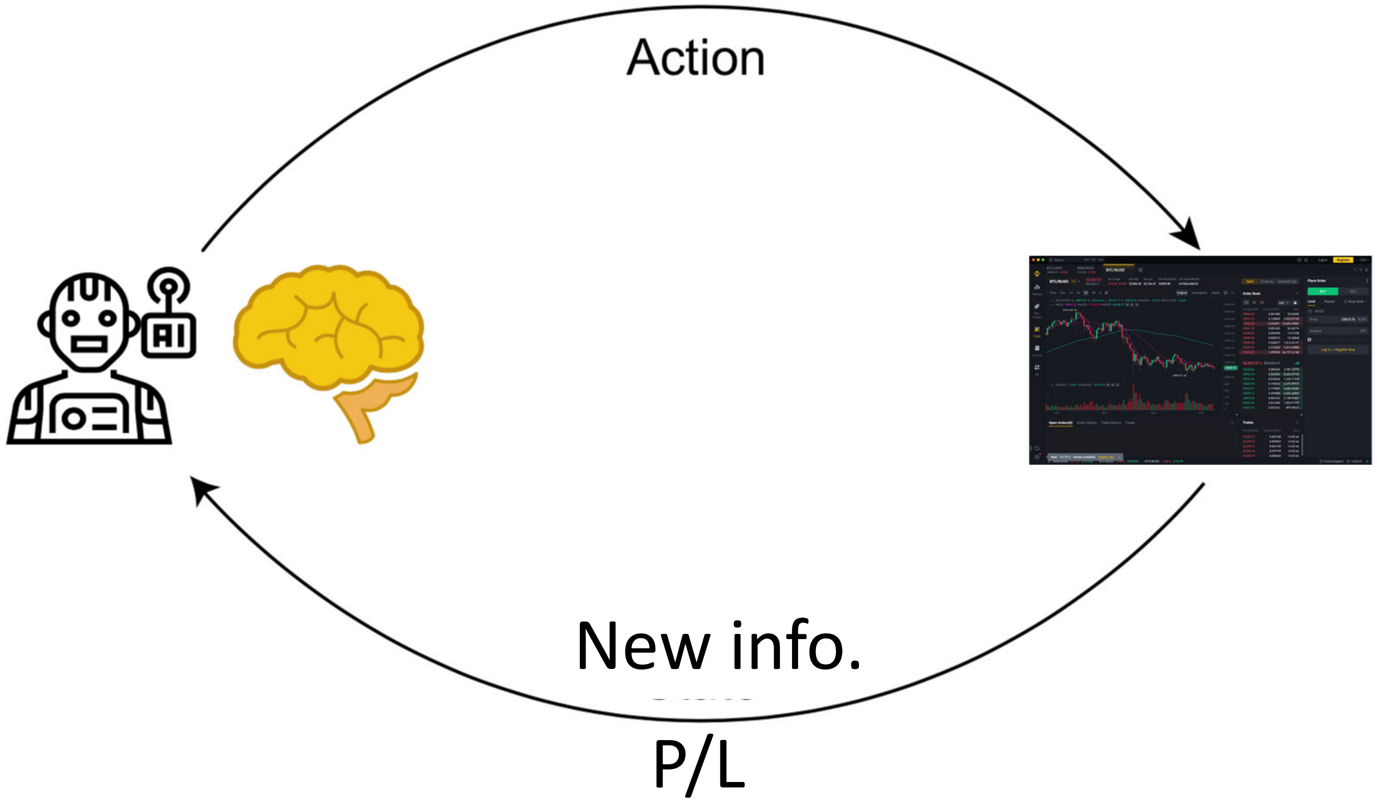
1. Market information: Price, Volume, books@depth5
2. Portfolio information: cash, assets, borrowing power.
3. Etc.

You need to **make a decision**:
=> Input how many @ what price

Motivation

- You can make predictions however you want, but ultimately an important part of trading is “action”.
  - Individuals: minutes -> daily basis.
  - Trading firms: nanoseconds -> daily basis.
- More broadly, a single trading decision can be modelled as a vector that contains the following components at minimum:
  - Which asset
  - Direction: BUY or SELL
  - Quantity: integer number
  - Order type: Limit Order (LMT) or Market Order (MKT)
  - Price (if LMT): a continuous number
- **Can we train a robot that makes predictions and submits orders together?**

Let’s view trading as a sequential decision-making process



Extension: under what condition/assumption is this framework valid?

Rule based approaches

- def monitor_twitter():
  - If Elon Mask tweets:
    - If +ve sentiment on dogecoin:
      - Buy 1 million dogecoin@market order
    - Else:
      - Sell 1 million dogecoin@market order
- Run monitor_twitter() until human intervention.
- Problem is, the logics are fixed/pre-determined by the coder.
  - Yes, they are much faster than human, but they are “dumb”.
  - Market evolves (quickly!) but the algorithms do not.
  - Not feasible to program every single scenario into a robot.
- Can we do better?

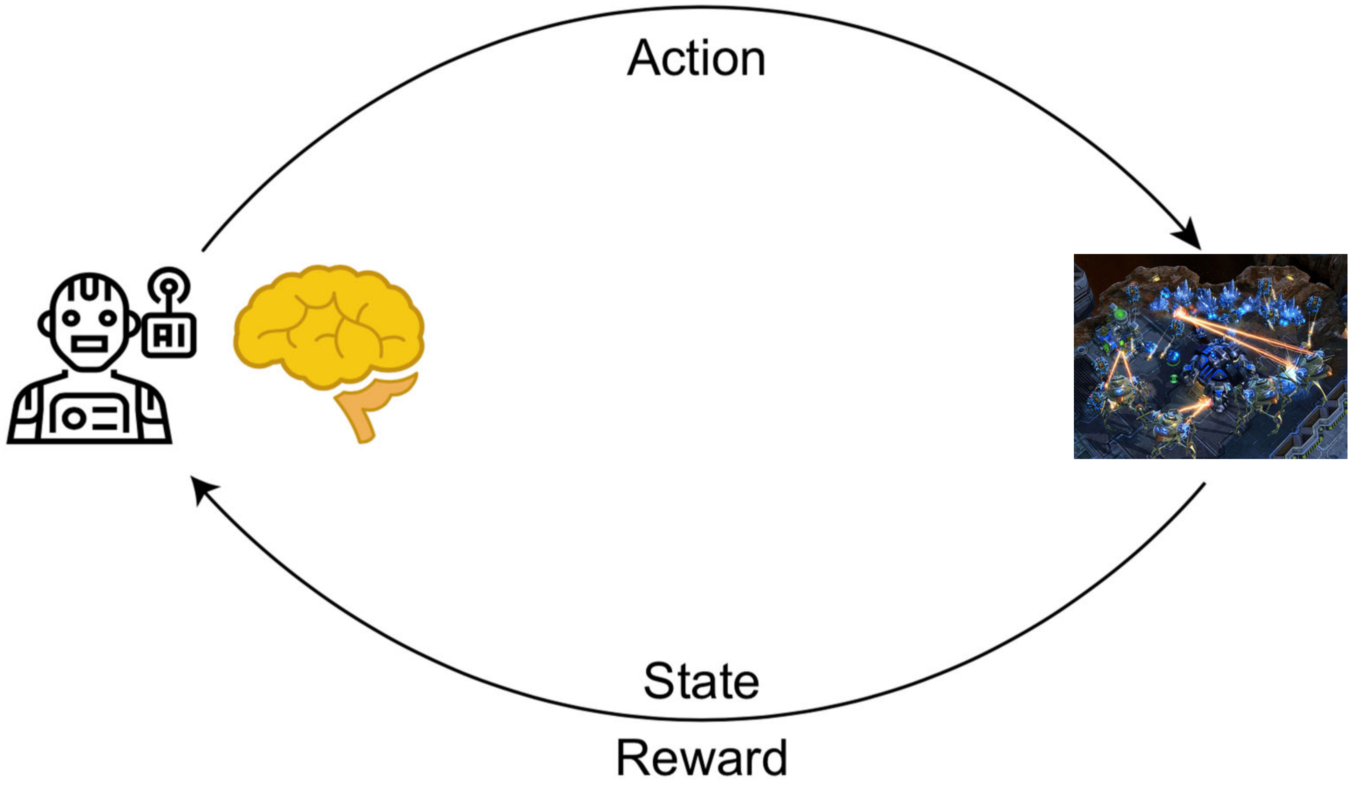
Learning and Artificial Intelligence

Recent AI research: an overview

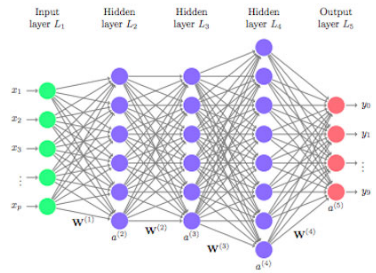
- Generic goal:
  - How do you build a system that ***learns*** to perform sequential decision-making tasks and outperform human?
- This is not trivial, where do you start?
  - Let’s ask how humans learn to play a game.
  - We learn by playing the game for many times ⇔ learn from experience.
    - +ve reward upon an action: “cool I did the right thing.”
    - Punishment (-ve reward) upon an action: “humm I think I did this step wrong.”
  - The core idea here is ***learning from experience***, both positive and negative ones.

A sequential decision process

- Essential components:
  - **Agents:** human or AI or system.
  - **States:** information received at each step.
  - **Actions:** available actions to perform.
  - **Reward:** +ve or –ve value when an agent performs an action.

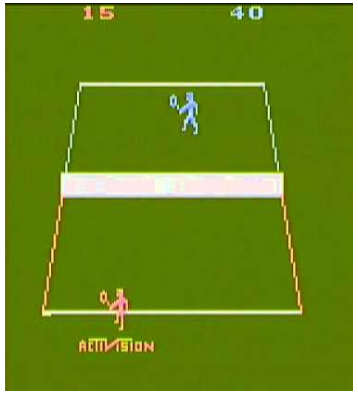


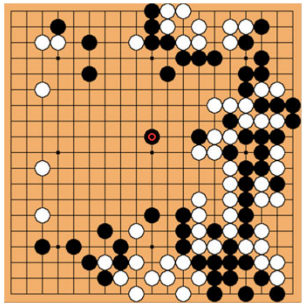
A sequential decision process



- This process can be modelled by a framework called ***reinforcement learning (RL)***.
  - Iteratively, agents learn to perform actions that return positive feedbacks, and avoid actions that return negative feedbacks.
- Neural networks offer the ability to process information/learn from a large and complicated state space.
  - Deep learning: neural networks with multiple layers.
  - Ability to process LOTS of information/data at each time step.
- Together they form a family of algorithms called ***deep reinforcement learning (DRL)*** which dominate humans in many (complicated) games.

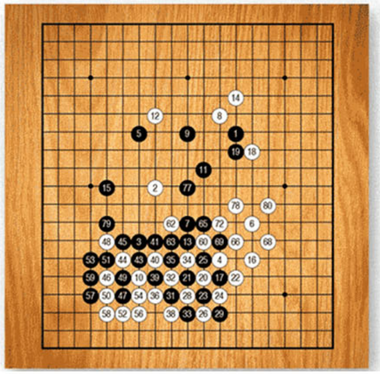
Super-human performance





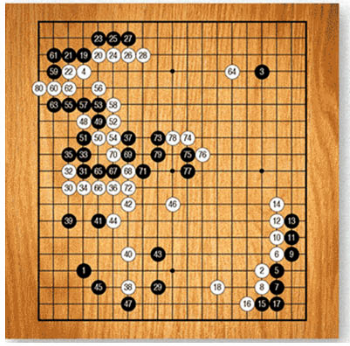
Games	Atari	Board Game	Dota / League of Legends	Starcraft II
Agents	1	2	≥ 1 (Generally 5 vs. 5)	≥ 2 (Generally 1 vs. 1)(?)
Information	Perfect information	Perfect Information	Imperfect information (fog of war)	Imperfect information (fog of war)
State space	Limited	Limited but large	Unlimited	Unlimited
Action space	Limited	Limited but large	Limited but large (one unit)	Unlimited (multiple units)
AI vs. Human	AI dominates	AI dominates	AI won in 1 vs. 1 5 vs. 5 (?)	AI dominates

Learning and strategy development



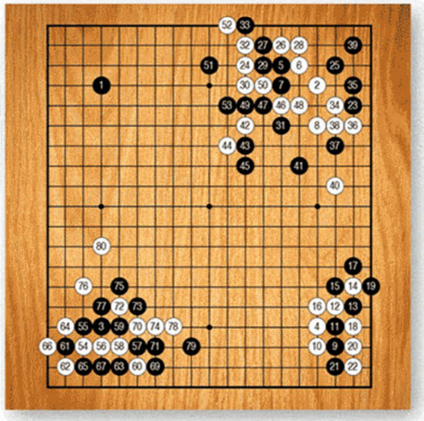
3 hours

AlphaGo Zero plays like a human beginner, forgoing long term strategy to focus on greedily capturing as many stones as possible.



19 hours

AlphaGo Zero has learnt the fundamentals of more advanced Go strategies such as life-and-death, influence and territory.



70 hours

AlphaGo Zero plays at super-human level. The game is disciplined and involves multiple challenges across the board.

AI in action (StarCraft II)

Technology/Army

Agents observe only part of the map

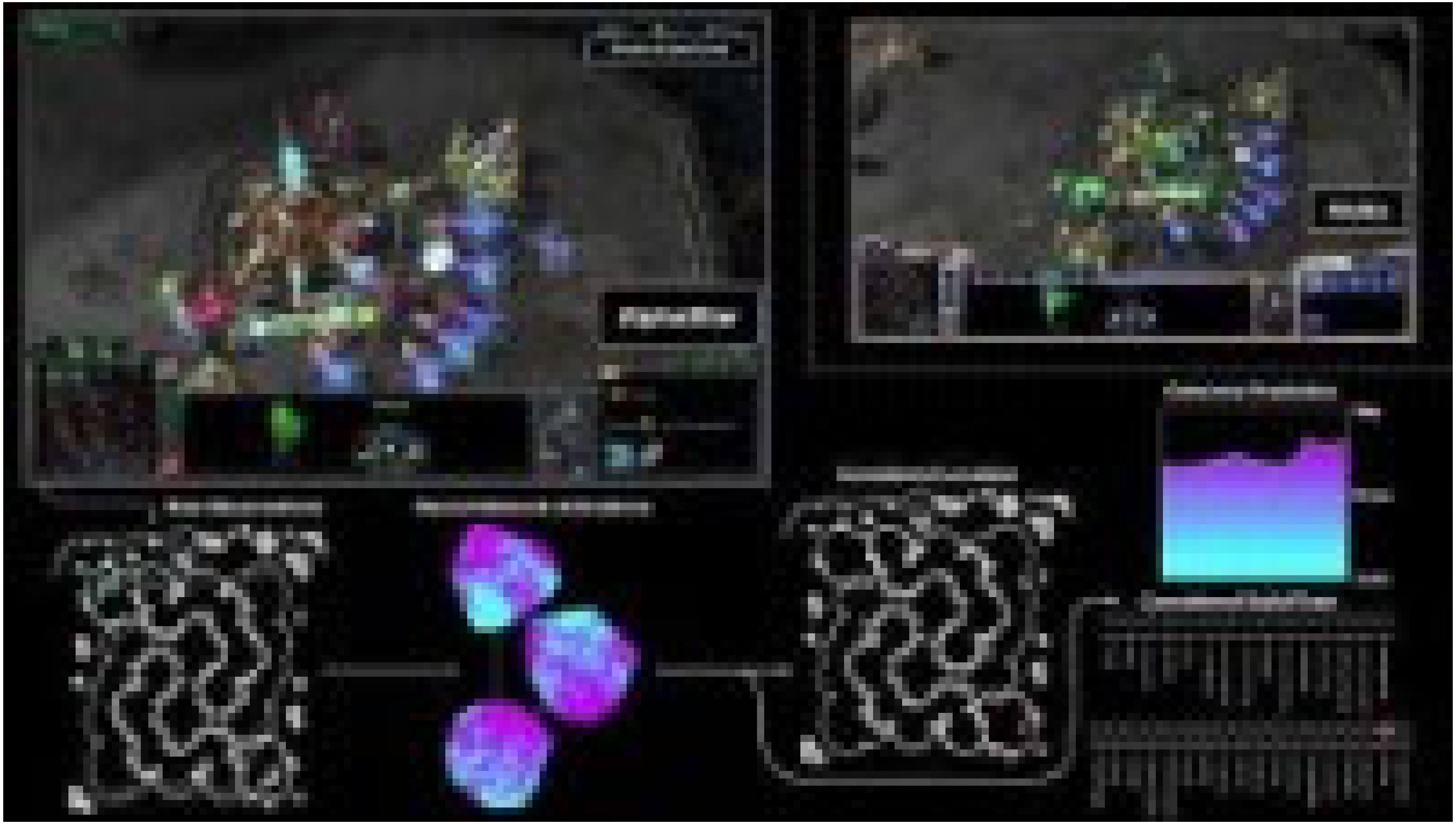
Resources

Fog of war (partial information)

Available actions for a given unit

AI in action (StarCraft II)

- Objective: to defeat the opponent(s).
- High-level optimization problem (macro-strategy), make a choice in the following action space:
  - Build forces.
  - Upgrade technology.
  - Expand your base to get more resources.
- Low-level optimization problem (micro-strategy):
  - E.g. if you want to build forces and attack.
    - Which forces?
    - How many?
    - Where do you want them to be?
    - What is your attacking strategy?

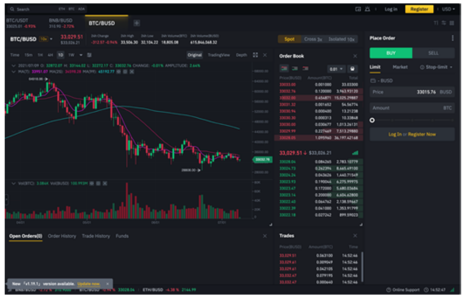


https://www.youtube.com/watch?v=HcZ48JDamyk

Trading:
A state-based control (RL) view

Trading ⇔ Gaming

States	Human/AI (RL system)	Action	Reward



Place a sell order of 5 @$99.95

Sell order executed
+1

Update training parameters (learning)

Move to location (52, 35)

Enemy spotted
+1

Update training parameters (learning)

Reinforcement learning is a framework

- RL is not necessarily about generic intelligence:
  - i.e. that you have one algorithm which works in all tasks.
  - It can be a sub-system within a much larger/broader system.
- It’s powerful and relatively flexible but it requires pre-defined state space, action space, rewards, etc.
- Question: what <u>***optimization***</u> you intend to achieve?
- Or simply what type of agent you want to build?
  - Market maker?
  - Order execution?
  - Etc.
- State space/action space/reward mechanism can be massively different.

Example 1: order execution

- A canonical trading problem: sell X number of shares in 10 mins.
  - Objective: sell X number of shares at a price as high as possible (average execution price).
  - Constraint: within 10 mins.
- In this case, one way to model it is to put not only the ask price but also time in the state space.
- If time goes towards the 10mins limit, the agents better learn to lower the sell price (cancel the existing orders and lower the position of the orders in the book) so that the ask orders are prioritized.

Example 1: order execution

Universal Trading for Order Execution with Oracle Policy Distillation

Yuchen Fang, [1*] **Kan Ren,** [2] **Weiqing Liu,** [2] **Dong Zhou,** [2]
Weinan Zhang, [1] **Jiang Bian,** [2] **Yong Yu,** [1] **Tie-Yan Liu**[2]
[1]Shanghai Jiao Tong University
[2]Microsoft Research
{arthur_fyc, wnzhang}@sjtu.edu.cn, yyu@apex.sjtu.edu.cn
{kan.ren, weiqing.liu, zhou.dong, jiang.bian, tyliu}@microsoft.com

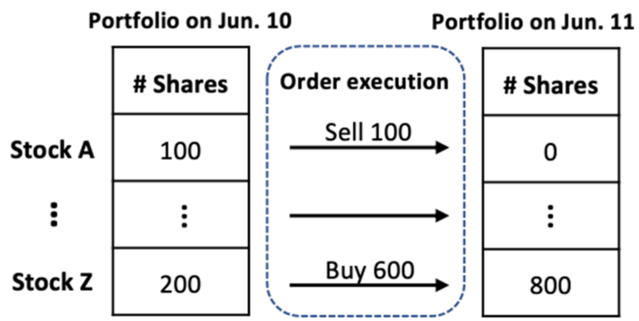


Figure 1: An example of portfolio adjustment which requires order execution.

Example 2: market making

Reinforcement Learning for Market Making in a Multi-agent Dealer Market

Sumitra Ganesh
JPMorgan AI Research

Nelson Vadori
JPMorgan AI Research

Mengda Xu
JPMorgan AI Research

Hua Zheng
JPMorgan Quantitative Research

Prashant Reddy
JPMorgan AI Research

Manuela Veloso
JPMorgan AI Research

An RL-based market maker

- States:
  - Trades executed in the previous time step.
  - Current inventory size.
  - Mid-point price (mid best bid-ask).
  - Bid and ask curve: limit order book@depth n.
  - Market share: proportion of your market volume in total market volume in the previous time step.
- Actions:
  - Pricing: the bid/ask price (or distance to the midpoint) you want to place in the book.
  - Hedging: % fraction of inventory to hedge.
- Reward:
  - Spread P&L.
  - Hedge cost: the cost to offload % of the inventory.
  - Inventory P&L.

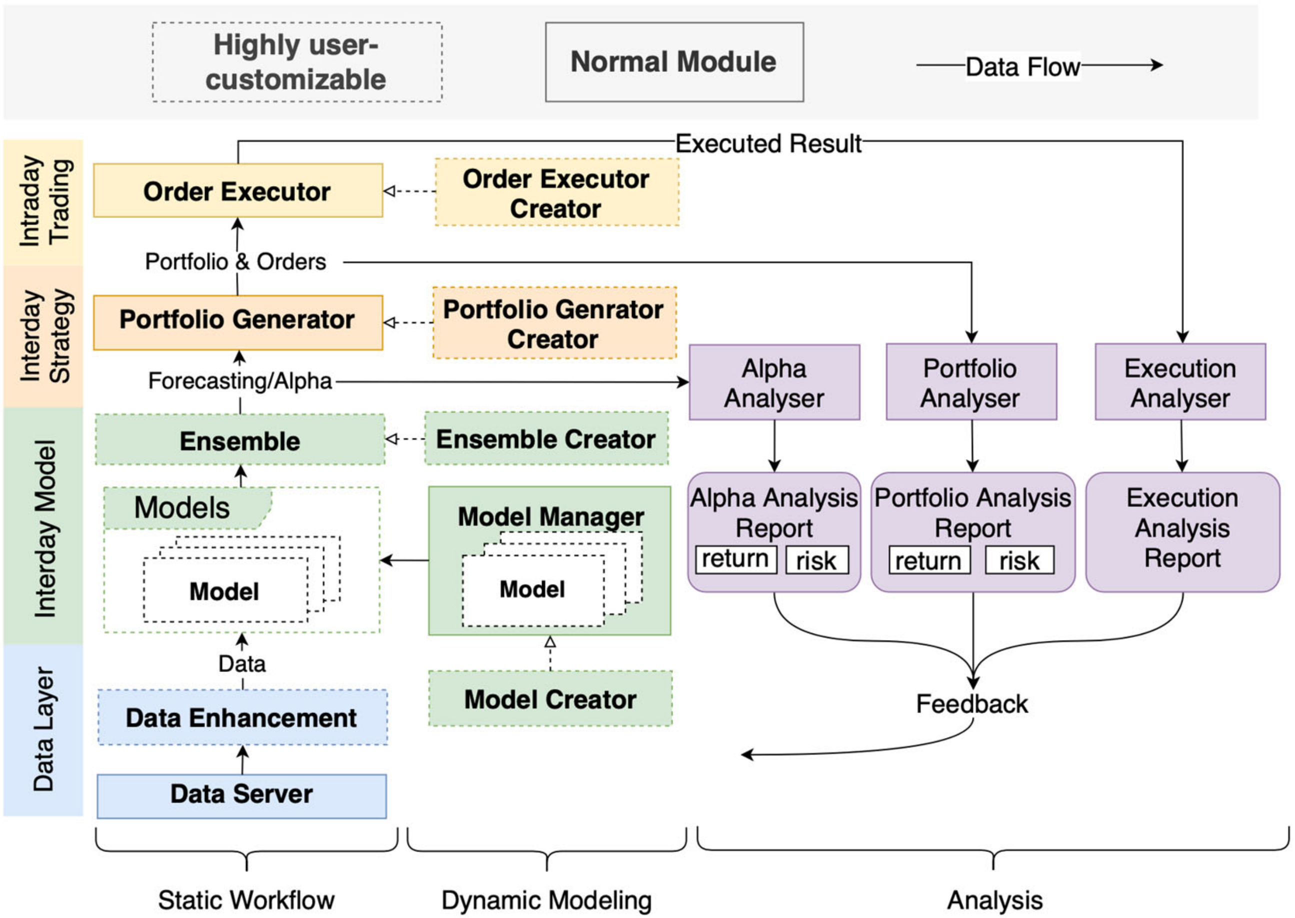
Example 3: A Financial Engineering Pipeline (1)

Qlib : An AI-oriented Quantitative Investment Platform

Xiao Yang, **Weiqing Liu**, **Dong Zhou**, **Jiang Bian** and **Tie-Yan Liu**
Microsoft Research
{Xiao.Yang, Weiqing.Liu, Zhou.Dong, Jiang.Bian, Tie-Yan.Liu}@microsoft.com

Example 3: A Financial Engineering Pipeline (1)

https://github.com/microsoft/qlib



Example 3: A Financial Engineering Pipeline (2)

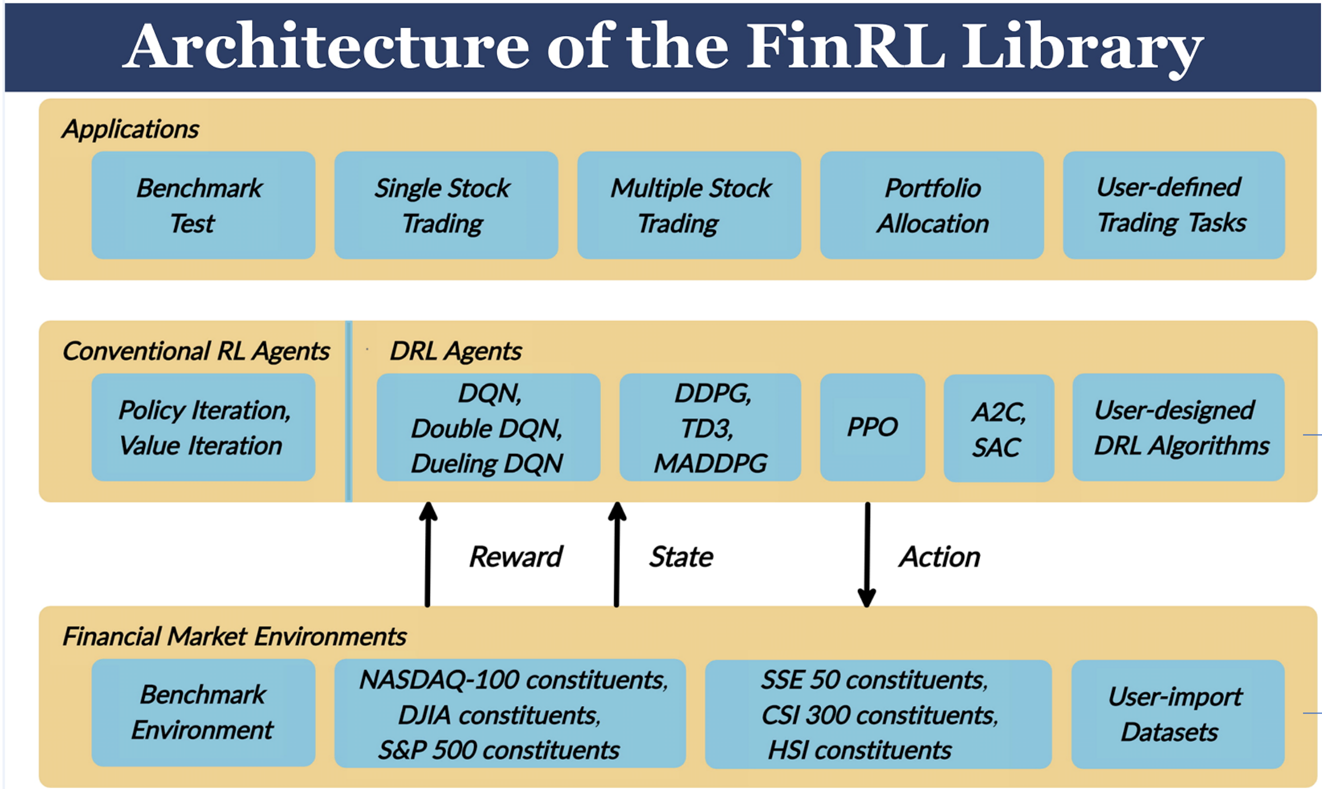
FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance

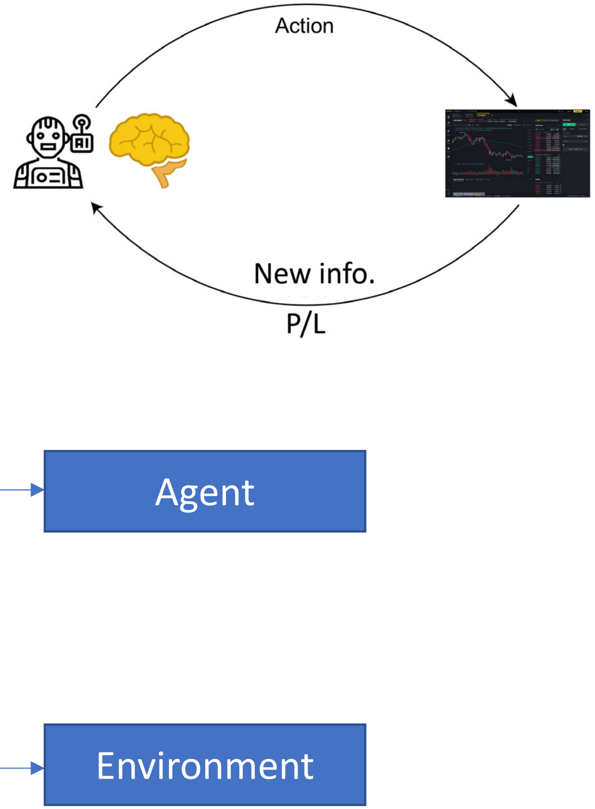
Xiao-Yang Liu[1*], Hongyang Yang[2,3*], Qian Chen[4,2], Runjia Zhang[3], Liuqing Yang[3], Bowen Xiao[5], Christina Dan Wang[6]
[1]Electrical Engineering, [2]Department of Statistics, [3]Computer Science, Columbia University,
[3]AI4Finance LLC., USA, [4]Ion Media Networks, USA,
[5]Department of Computing, Imperial College, [6]New York University (Shanghai)
Emails: {XL2427, HY2500, QC2231, LY2335}@columbia.edu,
info@ai4finance.net, bowen.xiao20@imperial.ac.uk, christina.wang@nyu.edu

Section 3.2.1 defines ***states, actions, reward functions***.

Example 3: A Financial Engineering Pipeline (2)

- https://github.com/AI4Finance-Foundation/FinRL





Some thoughts

- Be cautious!
  - These are (financial) engineering views of the trading problems.
  - Mainly focus on the following:
    - Modelling of agents' risk appetite (mostly risk-aversion)
    - Multi-agent collaboration and competition (game-theory)
    - ***Optimization*** (e.g. portfolio optimization)
    - Backtesting
  - Be careful about the assumptions they impose/results they claim…

There’s more….

- A lot of the complexity of trading environments today comes from the ***automation in different markets.***
  - Information available to you (states)? -> perfect vs. imperfect information
  - What do you do with the information? -> your strategy/actions
  - What do your counter parties do with the information? -> game
  - How do other market participants handle the information? - > multi-agent game
  - Concurrency? --> Market doesn’t wait for you to finalize a decision.
- But one thing for sure -> it’s much more than the price prediction.

Reference

- Algorithmic trading and machine learning by Michael Kearns:
  - https://www.youtube.com/watch?v=XNZ7o3621FY
- Reinforcement learning and market making:
  - https://arxiv.org/abs/1911.05892
- BorealisAI:
  - https://www.borealisai.com/en/
- AlphaGo Zero:
  - https://deepmind.com/blog/article/alphago-zero-starting-scratch
- Rossetta-Analytics and deep reinforcement learning
  - https://www.pionline.com/industry-voices/commentary-embracing-advanced-ai-achieve-consistent-returns
- FinRL
  - https://github.com/AI4Finance-Foundation/FinRL