ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING IN FINANCE

From nowadays applications to future possibilities

Mirko Polato, PhD – mpolato@math.unipd.it
16 Dicembre 2020

TABLE OF CONTENTS

01

ML & AI IN FINANCE

Nowadays applications

ML USE CASES IN FINANCE

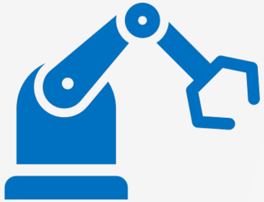
PROCESS AUTOMATION

SECURITY

ROBO-ADVISORY

ALGORITHMIC TRADING

PROCESS AUTOMATION



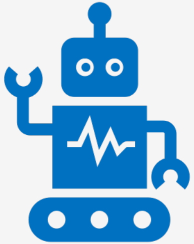
- **Chatbots:** for basic assistance for the users
- **Call-center automation**
- **Back office operational optimisation**: automates routine tasks with ML efficiency. Employees can then be used for higher-level tasks

SECURITY

- **Fraud detection**: algorithms examine in **real time** each action a cardholder takes and assess if an attempted activity is characteristic of that particular user

- **Network security**: to spot and isolate cyber threats

ROBO-ADVISORY

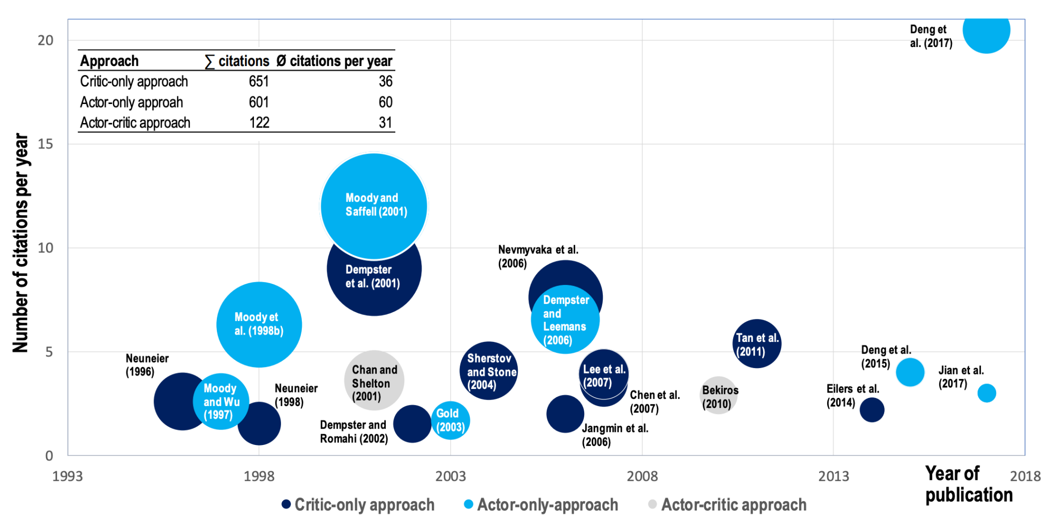


- **Portfolio management**: ML algorithms used for allocating, managing and optimizing clients' assets

- **Recommendation of financial products**: recommend personalized insurance plans to a particular user

ALGORTHMIC TRADING

High-Frequency Trading: according to statistics, nearly **73%** of the everyday trading is executed by machines.



Approach	∑ citations	Ø citations per year
Critic-only approach	651	36
Actor-only approah	601	60
Actor-critic approach	122	31

Reinforcement Learning papers on algorithmic trading

Fischer, Thomas G., 2018. "Reinforcement learning in financial markets - a survey". FAU Discussion Papers in Economics 12/2018.

02

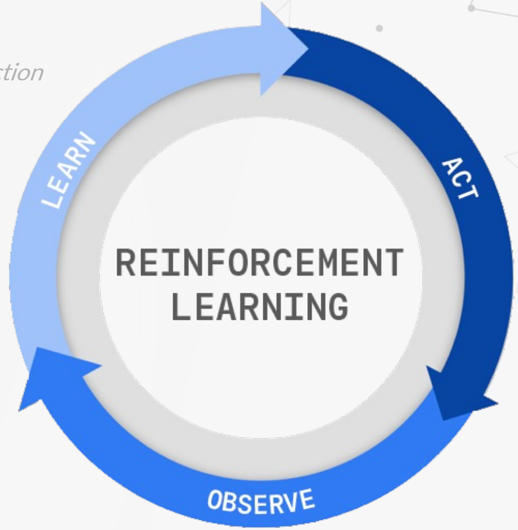
(DEEP) REINFORCEMENT LEARNING

A gentle introduction

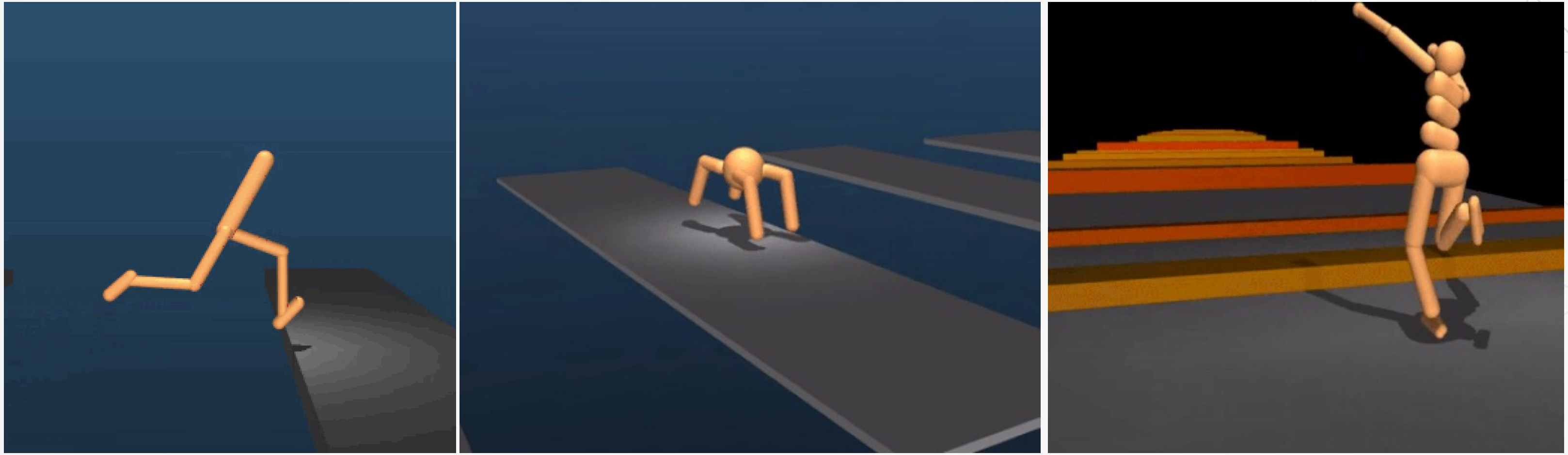
REINFORCEMENT LEARNING

> *Reinforcement learning is learning what to do—how to map situations to actions—so as to maximize a numerical reward signal.*
>
> Sutton & Barto. *Reinforcement learning: An introduction*

- No supervision, only a **reward** signal
- **Feedback is delayed**
- **Time matters**: sequential and non i.i.d. data
- The **agent's actions can affect the state**/environment

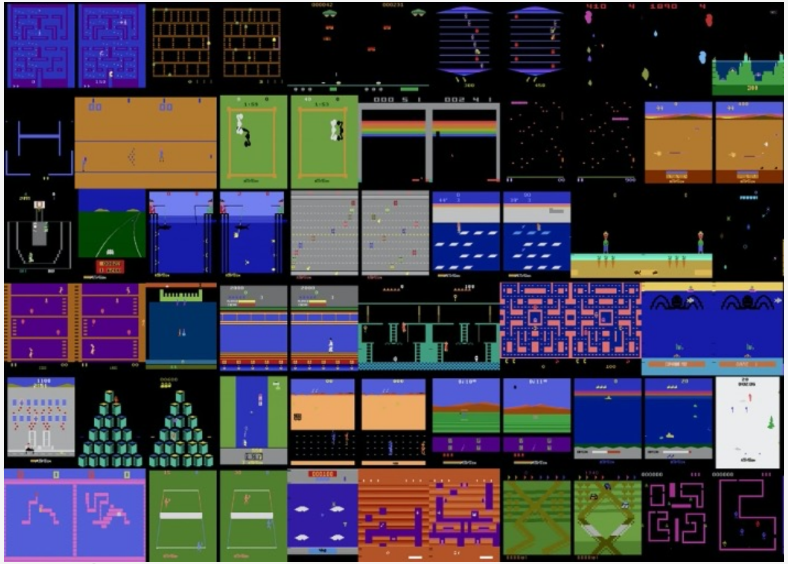


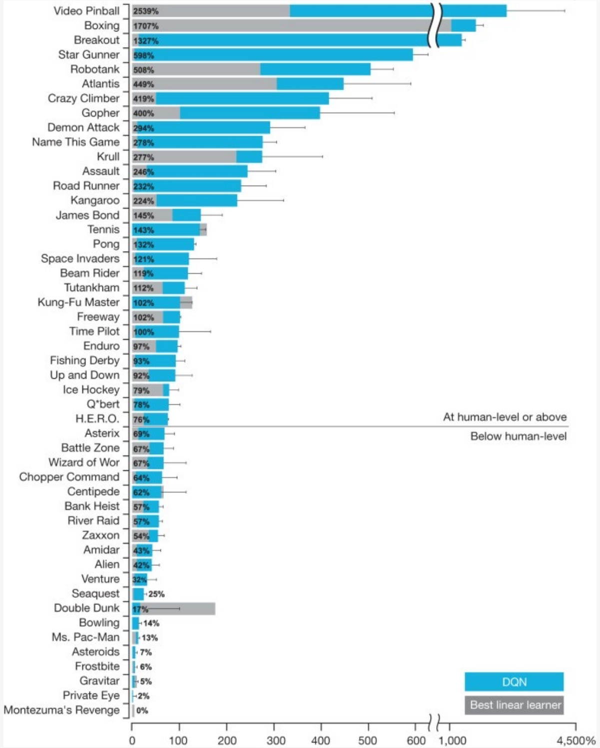
LEARNING TO WALK



Haarnoja, T., Zhou, A., Ha, S., Tan, J., Tucker, G., & Levine, S. (2019). Learning to Walk via Deep Reinforcement Learning. ArXiv, abs/1812.11103.

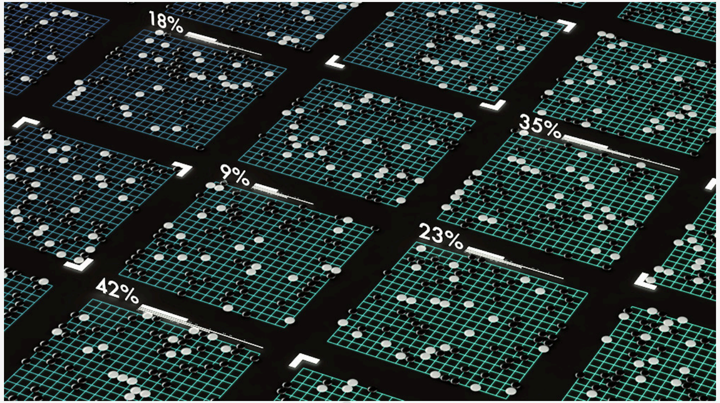
SUPER-HUMAN LEVEL IN ATARI GAMES





Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S. & Hassabis, D. (2015), 'Human-level control through deep reinforcement learning', Nature 518 (7540), 529--533.

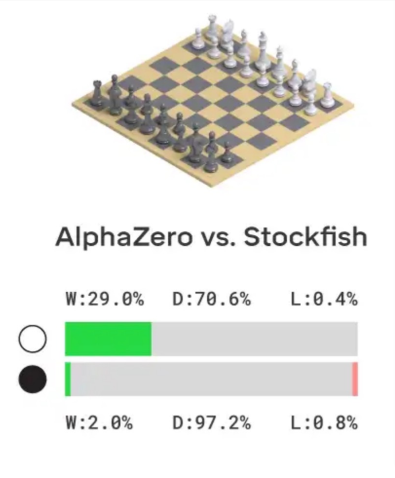
SUPER-HUMAN LEVEL IN GO

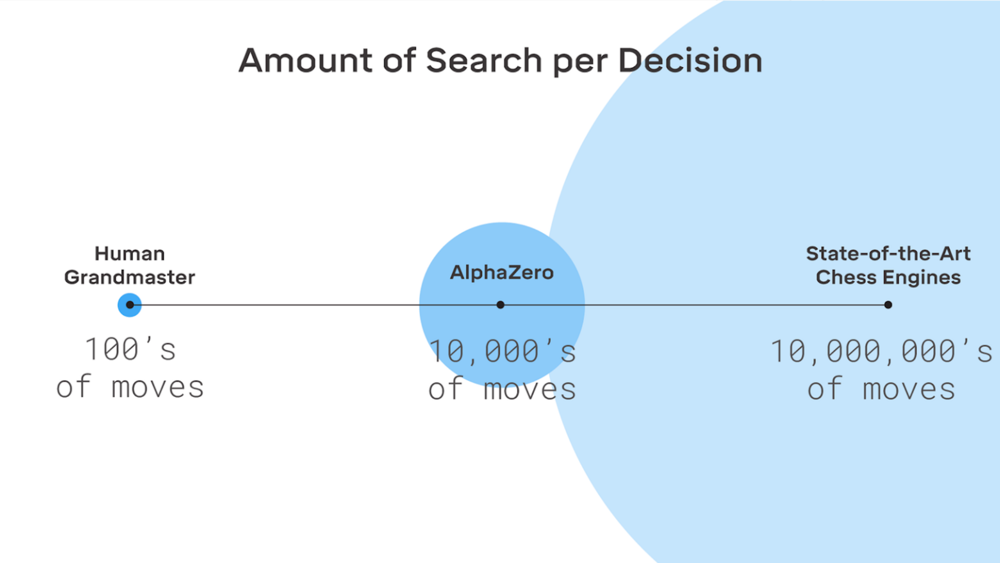


source: https://www.quantamagazine.org/is-alphago-really-such-a-big-deal-20160329/

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; Dieleman, S.; Grewe, D.; Nham, J.; Kalchbrenner, N.; Sutskever, I.; Lillicrap, T.; Leach, M.; Kavukcuoglu, K.; Graepel, T. & Hassabis, D. (2016), 'Mastering the Game of Go with Deep Neural Networks and Tree Search', Nature 529 (7587), 484--489.

SUPER-HUMAN AND STATE-OF-THE-ART LEVEL IN CHESS

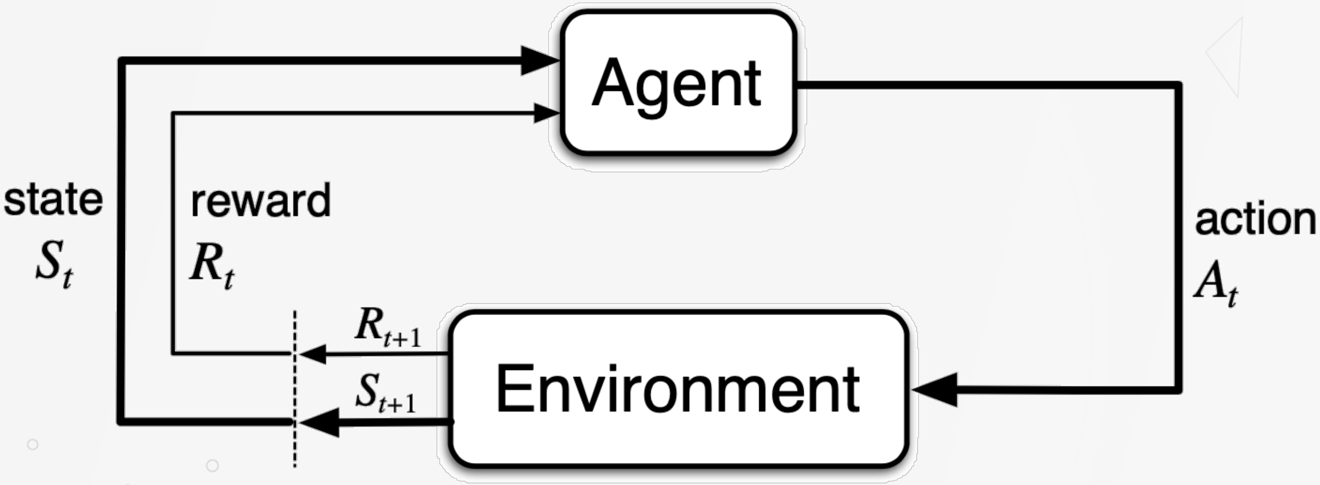




Source: https://deepmind.com/blog/article/alphazero-shedding-new-light-grand-games-chess-shogi-and-go

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., & Hassabis, D. (2017). Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. ArXiv, abs/1712.01815.

THE AGENT–ENVIRONMENT INTERFACE



Sutton, R.S. & Barto, A.G., 2018. *Reinforcement learning: An introduction*, MIT press.

REWARD

- A reward R_t is a **scalar feedback** signal
- Indicates how well agent is doing at step t
- The agent aims to **maximize its cumulative reward**

REWARD HYPOTHESIS

All goals can be described by the maximization of the expected cumulative reward

SEQUENTIAL DECISION MAKING

GOAL

Select (the sequence of) actions to maximize the total future reward.

- Actions may have **long term consequences**
- **Reward may be delayed**
- It may be better to sacrifice immediate reward to gain more **long-term reward**
    - Exploration vs exploitation trade-off
- For example: a financial investment (may take months to mature)

HISTORY AND STATE

HISTORY

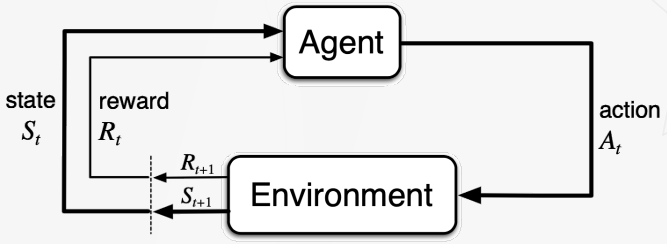
The history is the sequence of observations, actions, and rewards

$$H_t = O_1, R_1, A_1, \dots, A_{t-1}, O_{t-1}, R_t$$

STATE

The state is a function of the history

$$S_t = f(H_t)$$



- Environment state: the environment's private representation (may not be visible)
- Agent state: is the agent's internal representation, i.e., the information used by reinforcement learning algorithms

MARKOV PROPERTY

"The future is independent of the past given the present"

A state S_t is Markov if and only if

$$\mathbb{P}[S_{t+1}|S_t] = \mathbb{P}[S_{t+1}|S_1, \dots, S_t]$$

state transition probability

$\mathcal{P}_{S_t S_{t+1}}$

- The state captures all relevant information from the history
- The state is a **sufficient statistic of the future**

COMPONENTS OF AN RL AGENT (1)

POLICY (π)

- The **agent's behaviour**
- Deterministic policy:
  $$A_t = \pi(S_t)$$
- Stochastic policy:
  $$\pi(a|s) = \mathbb{P}[A_t = a \,|S_t = s]$$

MODEL

- **Predicts what the environment will do next**
- $\mathcal{P}$ predicts the next state
  $$\mathcal{P}_{ss'}^{a} = \mathbb{P}[S_{t+1} = s'|S_t = s, A_t = a]$$
- $\mathcal{R}$ predicts the immediate reward
  $$\mathcal{R}_s^a = \mathbb{E}[R_{t+1}|S_t = s, A_t = a]$$

COMPONENTS OF AN RL AGENT (2)

VALUE FUNCTION

- **Prediction of future reward**
- Used to evaluate the goodness/badness of states
- State-value function $v_\pi(s) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\right]$
- Action-value function $q_\pi(s,a) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right]$

Expected return

- $\gamma \in [0,1]$ is the discount factor

MARKOV DECISION PROCESS

(FINITE, FULLY OBSERVABLE) MDP

A **Markov Decision Process** is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$:

- $\mathcal{S}$ is a finite set of states
- $\mathcal{A}$ is a finite set of actions
- $\mathcal{P}$ is a state transition probability matrix

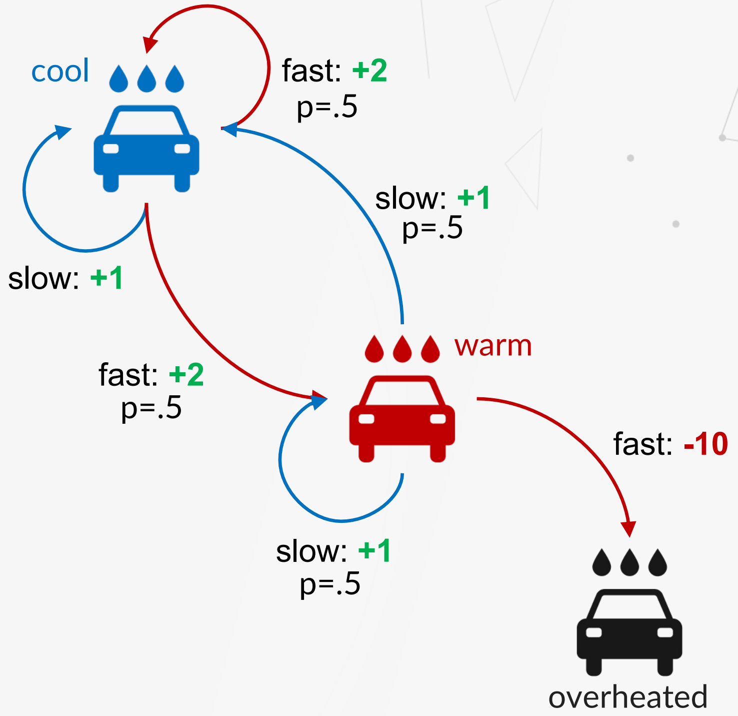
$$\mathcal{P}_{ss'}^{a} = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$$

- $\mathcal{R}$ is a reward function, $\mathcal{R}_s^a = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$
- $\gamma \in [0,1]$ is a discount factor

MDP EXAMPLE: RACING

GOAL: A robot car (agent) wants to travel **far** and **quick**.

- $\mathcal{S}$ = {cool, warm, overheated}
- $\mathcal{A}$ = {slow, fast}
- $$\mathcal{P} = \begin{array}{cc} \begin{array}{r} \text{cool} \\ \text{warm} \end{array} & \begin{bmatrix} (1,.5) & (0,.5) & (0,0) \\ (.5,0) & (.5,0) & (0,1) \end{bmatrix} \\ & \begin{array}{ccc} \text{cool} & \text{warm} & \text{over} \end{array} \end{array}$$
- $$\mathcal{R} = \begin{array}{cc} \begin{array}{r} \text{cool} \\ \text{warm} \end{array} & \begin{bmatrix} +1 & +2 \\ +1 & -10 \end{bmatrix} \\ & \begin{array}{cc} \text{slow} & \text{fast} \end{array} \end{array}$$



SOLVING AN MPD

An MDP is "solved" when we know the optimal value function

GOAL: finding the optimal state-value function

$$v_*(s) = \max_\pi v_\pi(s)$$

GOAL: finding the optimal action-value function

$$q_*(s,a) = \max_\pi q_\pi(s,a) = \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = a, A_t = a]$$

GOAL: finding the optimal policy

An optimal policy can be found by maximizing over $q_*(s,a)$

$$\pi_*(a|s) = \begin{cases} 1, & a = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, q_*(s,a) \\ 0, & otherwise \end{cases}$$

LEARNING VIA "TRIAL & ERROR"

- The learning is performed by improving v_π and q_π using a **trial-and-error** approach
- The agent tries to perform the task by taking an action using its current policy (initially random)
- Based on the obtained rewards $\boldsymbol{v_\pi}$ **and** $\boldsymbol{q_\pi}$ **are updated,** and hence the policy $\boldsymbol{\pi}$
- This process is **repeated several (e.g., millions) times**
- Randomization, i.e., not always choose the best action according to π, is usually used to guarantee **exploration** (e.g., ε-greedy policy)

TABULAR-BASED METHODS

- When MDPs are small v_π and q_π can be stored in a table!
- Popular tabular methods:
  - Monte-Carlo methods (simulation-based)
  - Temporal difference learning (TD-Learning)
  - **Q-Learning / R-Learning**
  - **SARSA**

HIGHLY INEFFICIENT

For real-world MDPs tabular approaches are **not feasible**!

APPROXIMATION-BASED METHODS

- Estimate value function with function approximation ➜ no need to store the table!

$$v_\pi(s) \approx \hat{v}(s, \theta)$$
$$q_\pi(s, a) \approx \hat{q}(s, a, \theta)$$

- **Generalize** to unseen states

- Usually **(Deep) Neural Networks** are chosen as function approximator

- Popular approximation based RL methods:
  - Deep Q-Network (DQN) / **Double DQN**
  - **A2C**
  - **Policy Gradient (PG)**
  - Rainbow

DQN

- DQN uses an **ε-greedy** policy
- Store transitions (s_t, a_t, r_{t+1}, s_{t+1}) in a replay memory *D*
- Sample random mini-batch of transitions (s, a, r, s′) from *D*
- Compute Q-learning targets w.r.t. old, fixed parameters θ'
- Optimize MSE between Q-network and Q-learning targets

current network

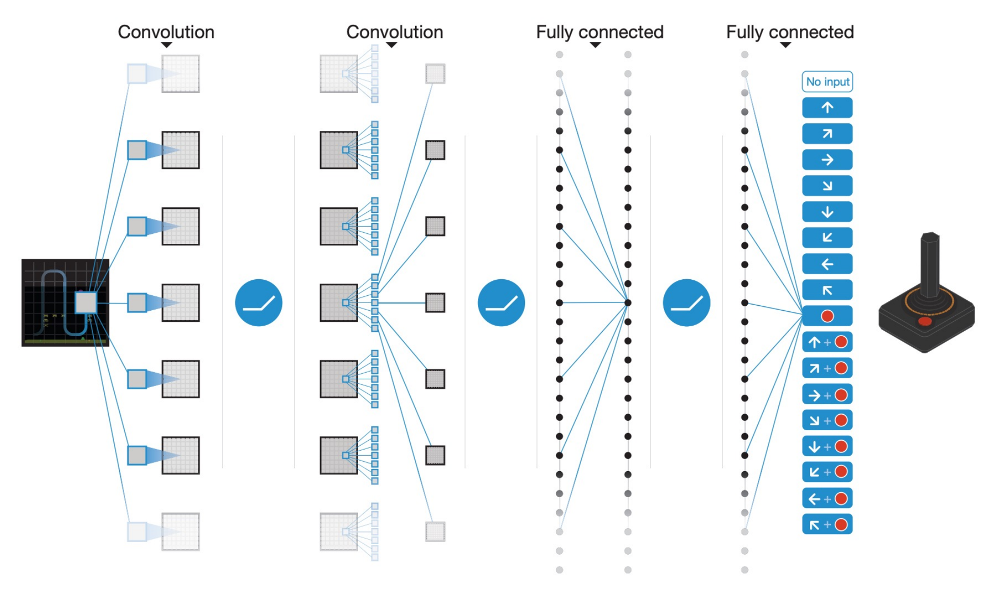
$$\mathcal{L}(\theta) = \mathbb{E}_{s,a,r,s'\sim D}\left[\left(r + \gamma \max_{a'} Q(s', a'; \theta') - Q(s, a; \theta)\right)^2\right]$$

target (old) network

- Use a variant of stochastic gradient descent (SGD)

DQN IN ATARI

- End-to-end learning of values *q(s, a)* from pixels
- **Input** state *s* is a stack of raw pixels from last 4 frames
- **Reward** is the change in score for that step



Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S. & Hassabis, D. (2015), 'Human-level control through deep reinforcement learning', Nature 518 (7540), 529--533.

03

REINFORCEMENT LEARNING IN FINANCE

Examples of RL methods in financial applications

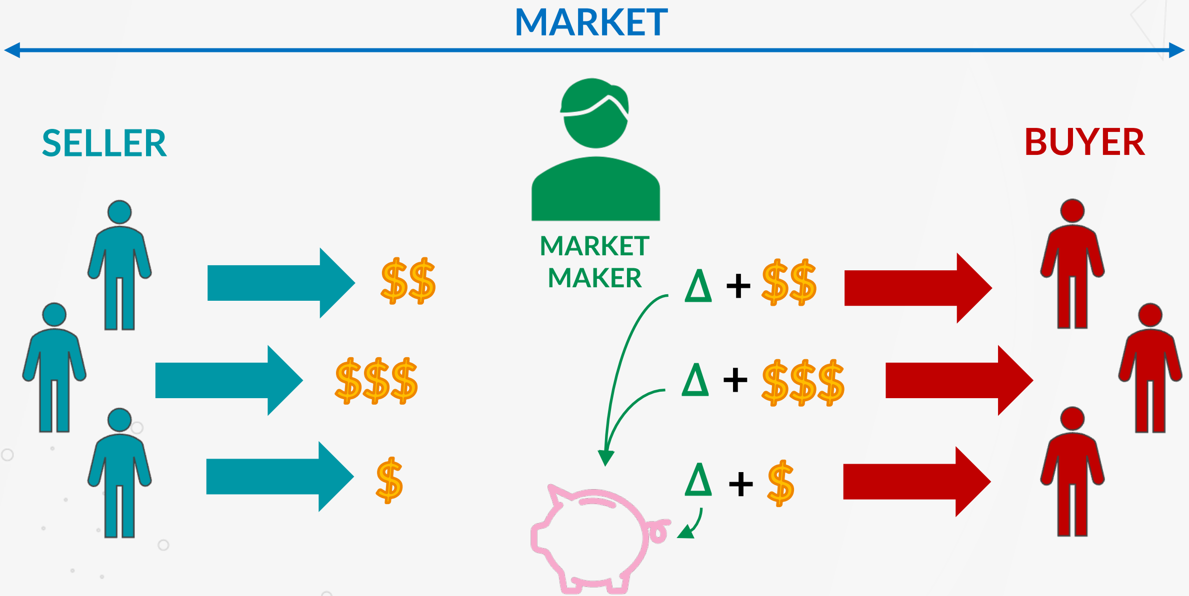
3.1 MARKET MAKING VIA REINFORCEMENT LEARNING

Thomas Spooner, John Fearnley, Rahul Savani, and Andreas Koukorinis. 2018. Market Making via Reinforcement Learning. In Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '18). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 434–442.

MARKET MAKER

> *Traders who profit from facilitating exchange in a particular asset and exploit their skills in executing trades*

Cartea, Jaimungal, & Penalva. Algorithmic and High-Frequency Trading.



PROFIT & RISKS OF A MARKET MAKER

PROFIT

- Spread (Δ)
- Favorable market: increasing of the value of the owned financial instruments

RISKS

- Non-zero inventory: bought financial instruments are never sold, or *viceversa*
- Unfavorable market: decreasing of the value of the owned financial instruments

REWARD FUNCTION

PnL REWARD: the money lost/gained through executions of the orders relative to the mid-price

agent's quoted spread + inventory increment – dampening factor

agent's price

$$R_t = X_t^a \cdot [p_t^a - m_t] + X_t^b \cdot [m_t - p_t^b] + I_t \Delta m_t - \eta D(I_t)$$

volume matched (executed) against the agent's orders since t−1 in the order books

mid-price

ACTION SPACE

Action ID	0	1	2	3	4	5	6	7	8
Ask (θ_a)	1	2	3	4	5	1	3	2	5
Bid (θ_b)	1	2	3	4	5	3	1	5	2
Action 9	MO with $\text{Size}_m = -\text{Inv}(t_i)$								

clear its inventory using a Market Order

Agent's pricing strategy: $p_t^{a,b} = m_t + \frac{1}{2}\theta_t^{a,b} s_t$

STATE SPACE

AGENT-STATE

- **Inv(t_i)**: the amount of stock currently owned or owed by the agent
- Effective values of the control parameters, **$\theta_{a,b}$**, after going forward in the simulation

MARKET-STATE/ENVIRONMENT-STATE

- Market (bid/ask) spread
- Mid-price move (Δm)
- LOB imbalance
- Volatility
- RSI

EXPERIMENTAL SETTING

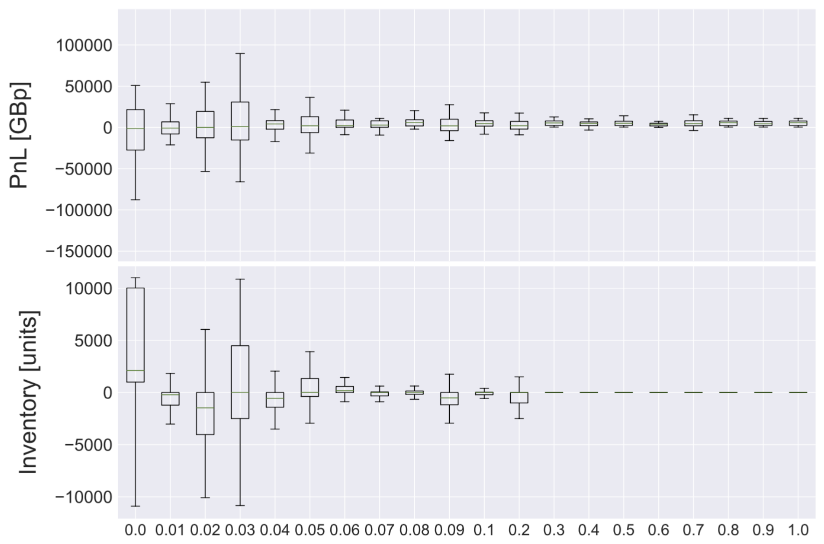
- **Simulated data** of a financial market via direct reconstruction of the limit order book from historical data (January – August 2010) of 10 securities from 4 different sectors

- Tested RL models:
    - Q-learning
    - SARSA
    - R-learning
    - Variants of the previous approaches
    - Consolidated agent: SARSA + ad-hoc state representation

RESULTS – Performance of the baselines

	QL	SARSA
CRDI.MI	**8.14 ± 21.75**	4.25 ± 42.76
GASI.MI	−4.06 ± 48.36	**9.05 ± 37.81**
GSK.L	4.00 ± 89.44	**13.45 ± 29.91**
HSBA.L	**−12.65 ± 124.26**	−12.45 ± 155.31
ING.AS	−67.40 ± 261.91	**−11.01 ± 343.28**
LGEN.L	**5.13 ± 36.38**	2.53 ± 37.24
LSE.L	4.40 ± 16.39	**5.94 ± 18.55**
NOK1V.HE	**−7.65 ± 34.70**	−10.08 ± 52.10
SAN.MC	−4.98 ± 144.47	**39.59 ± 255.68**
VOD.L	**15.70 ± 43.55**	6.65 ± 37.26

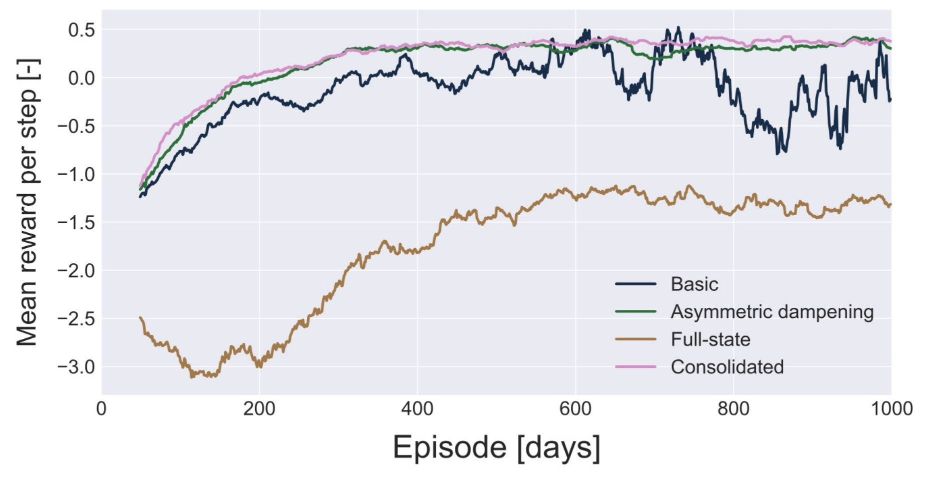
With non-dampened PnL reward

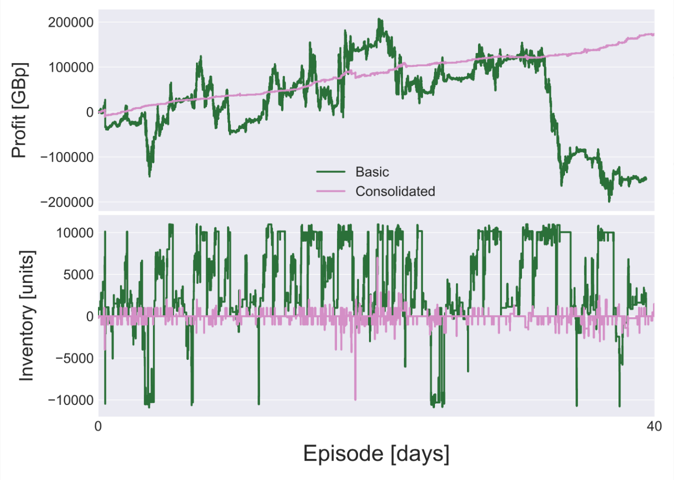


dampening factor η

RESULTS – Variants and Consolidated agent

	CRDI.MI	GASI.MI	GSK.L	HSBA.L	ING.AS	LGEN.L	LSE.L	NOK1V.HE	SAN.MC	VOD.L
Double Q-learning	−5.04 ± 83.90	5.46 ± 59.03	6.22 ± 59.17	5.59 ± 159.38	58.75 ± 394.15	2.26 ± 66.53	16.49 ± 43.10	−2.68 ± 19.35	5.65 ± 259.06	7.50 ± 42.50
Expected SARSA	0.09 ± 0.58	3.79 ± 35.64	−9.96 ± 102.85	25.20 ± 209.33	6.07 ± 432.89	2.92 ± 37.01	6.79 ± 27.46	−3.26 ± 25.60	32.28 ± 272.88	15.18 ± 84.86
R-learning	5.48 ± 25.73	−3.57 ± 54.79	12.45 ± 33.95	−22.97 ± 211.88	−244.20 ± 306.05	−3.59 ± 137.44	8.31 ± 23.50	−0.51 ± 3.22	8.31 ± 273.47	32.94 ± 109.84
Double R-learning	19.79 ± 85.46	−1.17 ± 29.49	21.07 ± 112.17	−14.80 ± 108.74	5.33 ± 209.34	−1.40 ± 55.59	6.06 ± 25.19	2.70 ± 15.40	32.21 ± 238.29	25.28 ± 92.46
On-policy R-learning	0.00 ± 0.00	4.59 ± 17.27	14.18 ± 32.30	9.56 ± 30.40	18.91 ± 84.43	−1.14 ± 40.68	5.46 ± 12.54	0.18 ± 5.52	25.14 ± 143.25	16.30 ± 32.69





3.2 DEEP REINFORCEMENT LEARNING FOR TRADING

Zihao Zhang, Stefan Zohren, Roberts Stephen, 2020. Deep Reinforcement Learning for Trading. The Journal of Financial Data Science. DOI: https://doi.org/10.3905/jfds.2020.1.030

TRADING

- Profit from buying & selling different financial instruments
- Deals with probability never certainty
- Trading vs Investing: holding period

Goal of a trader (*)
Maximize some expected utility (U) of final wealth

$$\mathbb{E}[U(W_T)] = \mathbb{E}\left[U\left(W_0 + \sum_{t=1}^{T} \delta W_t\right)\right]$$

Linear U

Goal of RL
Maximize the expected return G, i.e., the expected discounted cumulative rewards

$$\mathbb{E}[G] = \mathbb{E}\left[\sum_{k=t+1}^{T} \gamma^{k-t-1} R_k\right]$$

(*) Modern portfolio theory:
- Arrow, K. J. “The Theory of Risk Aversion.” In Essays in the Theory of Risk-Bearing, pp. 90–120. Chicago: Markham, 1971.
- Pratt, J. W. “Risk Aversion in the Small and in the Large.” In Uncertainty in Economics, pp. 59–79. Elsevier, 1978.
- Ingersoll, J. E. Theory of Financial Decision Making, vol. 3. Lanham, MD; Rowman & Littlefield, 1987.

ACTION SPACE

DISCRETE

- **-1**: maximally short position - SELL
- **0**: no holdings - DO NOTHING
- **+1**: maximally long position – BUY
- If $a_t = a_{t+1}$: no transaction costs;
  If $a_t = -a_{t+1}$: double transaction costs.

REWARD FUNCTION

REWARD: profits representing a risk-insensitive trader

cost rate: $\beta = 10^{-4}$

price

volatility target

$$R_t = A_t \frac{\sigma_{tgt}}{\sigma_{t-1}} (p_t - p_{t-1}) - \beta \, p_{t-1} \left| \frac{\sigma_{tgt}}{\sigma_{t-1}} A_{t-1} - \frac{\sigma_{tgt}}{\sigma_{t-2}} A_{t-2} \right|$$

additive profit

ex ante volatility calculated using a weighted moving std with a 60-day window on the additive profit

STATE SPACE

- Normalized close price series
- Normalized returns over the past 1, 2, 3 and 12 months
- MACD(*) indicator which "measures" the momentum, direction and duration of the trend of the price.
- RSI indicator in [0, 100] with a look-back window of 30 days
    - ≤20: oversold
    - ≥80: overbought

(*) Baz, J., N. Granger, C. R. Harvey, N. Le Roux, and S. Rattray. “Dissecting Investment Strategies in the Cross Section and Time Series.” SSRN 2695101, 2015.

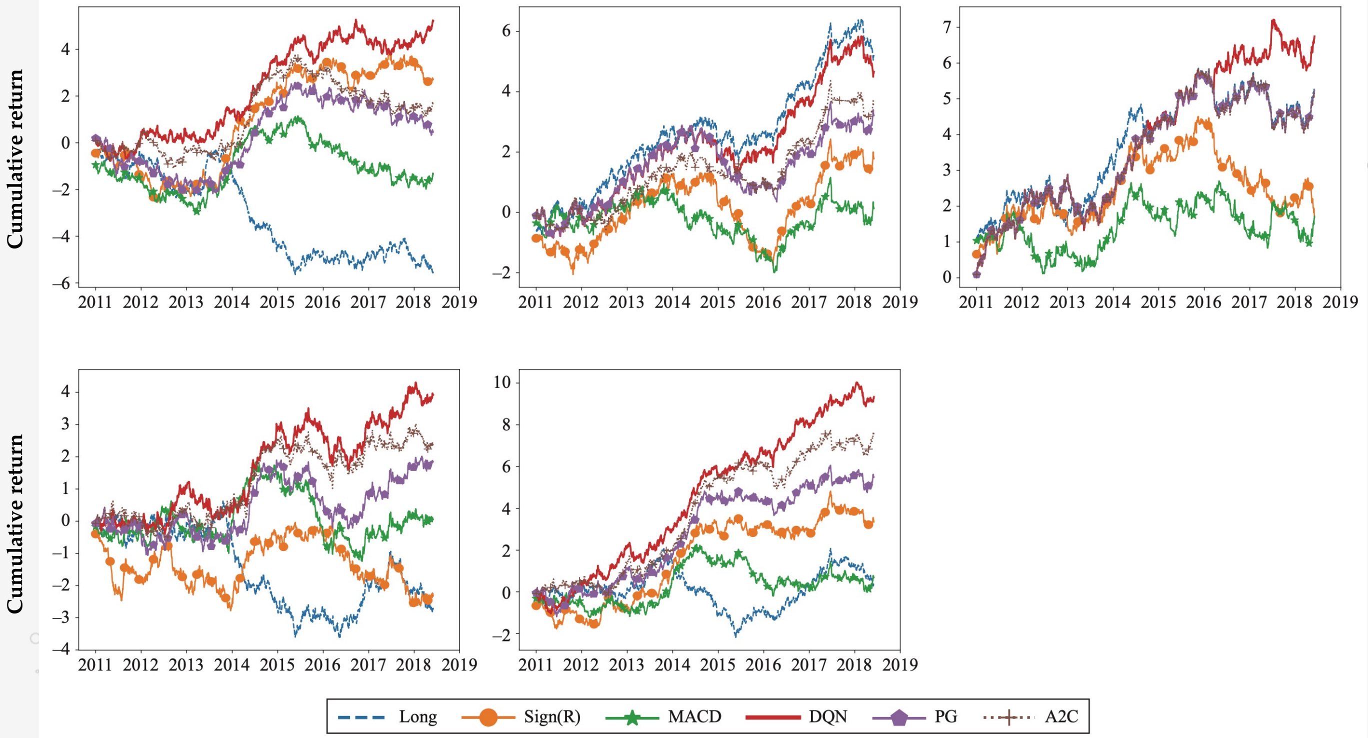
EXPERIMENTAL SETTING

- Dataset: **CLC Database** (*2019) that ranges from 2005 to 2019 and consists of a variety (4) of asset classes

- Function approximator: 2-layer **LSTM** with 64/32 units, and Leaky-RELU
- RL techniques: DQN, A2C and PG
- A separate model for each asset class is trained
- The portfolio is equally distributed over all the asset classes

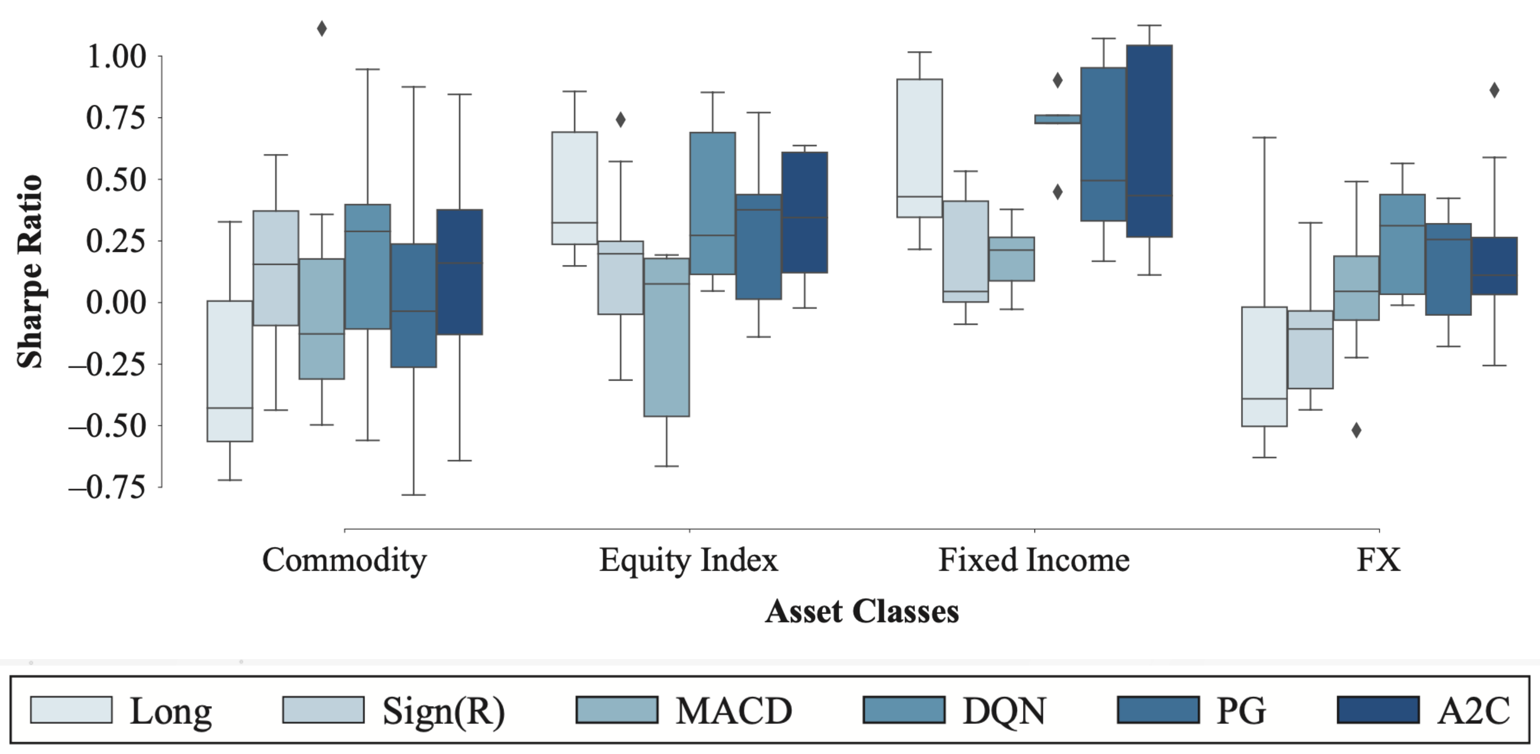
(*2019) CLC Database. Pinnacle Data Corp, 2019, https://pinnacle- data2.com/clc.html.

RESULTS (1)



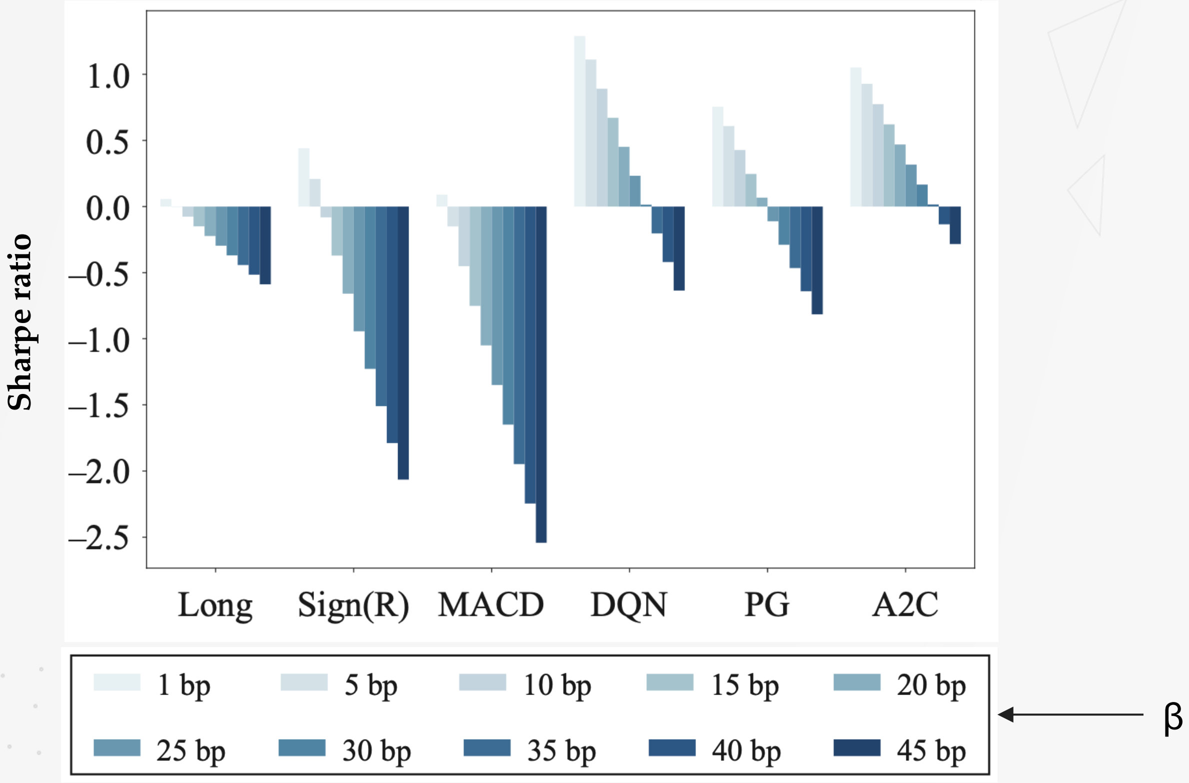
Cumulative return
Cumulative return
2011 2012 2013 2014 2015 2016 2017 2018 2019
Long
Sign(R)
MACD
DQN
PG
A2C

RESULTS (2)



Sharpe Ratio
1.00
0.75
0.50
0.25
0.00
–0.25
–0.50
–0.75
Commodity
Equity Index
Fixed Income
FX
Asset Classes
Long
Sign(R)
MACD
DQN
PG
A2C

RESULTS (3)



1.0
0.5
0.0
–0.5
–1.0
–1.5
–2.0
–2.5
Sharpe ratio
Long
Sign(R)
MACD
DQN
PG
A2C
1 bp
5 bp
10 bp
15 bp
20 bp
25 bp
30 bp
35 bp
40 bp
45 bp
β

04

FUTURE POSSIBILITIES

What's next?

% OF AI OFFERING FUNCTIONS IN BANKING

Source: EMERJ

AI SOLUTIONS TO BE CONSIDERED

Source: https://towardsdatascience.com/machine-learning-in-finance-why-what-how-d524a2357b56

ML & AI IN FINANCE IN THE FUTURE

- **Investment insights**: use of alternative data sources, from NLP of annual reports to satellite photo interpretation for predictions regarding fruitful, less risky investments

- **Compliance with legislations**: as regulatory system is frequently updated. It is crucial to bring every financial operation in line with the relevant legislation

- **Loan underwriting**: analyzing previous activity and making forecasts on possible future customer's actions and reactions, organizations can avoid potential risks and enhance operational effectiveness.

USEFUL REFERENCES

1. Sutton, R.S. & Barto, A.G., 2018. Reinforcement learning: An introduction, MIT press.

2. Thomas Spooner, John Fearnley, Rahul Savani, and Andreas Koukorinis. 2018. Market Making via Reinforcement Learning. In Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '18). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 434–442.

3. Fischer, Thomas G., 2018. "Reinforcement learning in financial markets - a survey. FAU Discussion Papers in Economics 12/2018.

4. Zihao Zhang, Stefan Zohren, Roberts Stephen, 2020. Deep Reinforcement Learning for Trading. The Journal of Financial Data Science.

5. Farzan Soleymani, Eric Paquet. Financial portfolio optimization with online deep reinforcement learning and restricted stacked autoencoder—DeepBreath. Expert Systems with Applications, Volume 156, 2020, 113456, ISSN 0957-4174.

6. Francesco Bertoluzzo, Marco Corazza. Testing Different Reinforcement Learning Configurations for Financial Trading: Introduction and Applications. Procedia Economics and Finance, Volume 3, 2012, Pages 68-77, ISSN 2212-5671.

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO MATEMATICA

dSEA

THANK YOU

QUESTIONS?

“ *The only stupid question is the one you were afraid to ask but never did.*

Richard Sutton

Mirko Polato, PhD
mpolato@math.unipd.it