

One Button Trader

Last Updated: October 11, 2021

Table of Contents

Table of Contents	1
Glossary	2
Introduction	2
Problem/Opportunity	3
Initial Validation	6
Product	7
How It Works	7
UI	8
Dashboard	8
Launching Your Trading AI	9
The AI Misconception	10
AI Decision-Making	11
AI Foundations	12
Refresher on Neural Networks	12
Evolutionary Strategies	13
Why evolutionary strategies?	16
AI Models	19
Astral (Filter)	20
Horizon/Ascendant (GRU)	21
Performer	22
Complex Adaptive Systems (Discussion)	23
Team	25

Glossary

AI	Artificial Intelligence
ML	Machine Learning
NN	Neural Networks
OB	One Button

Introduction

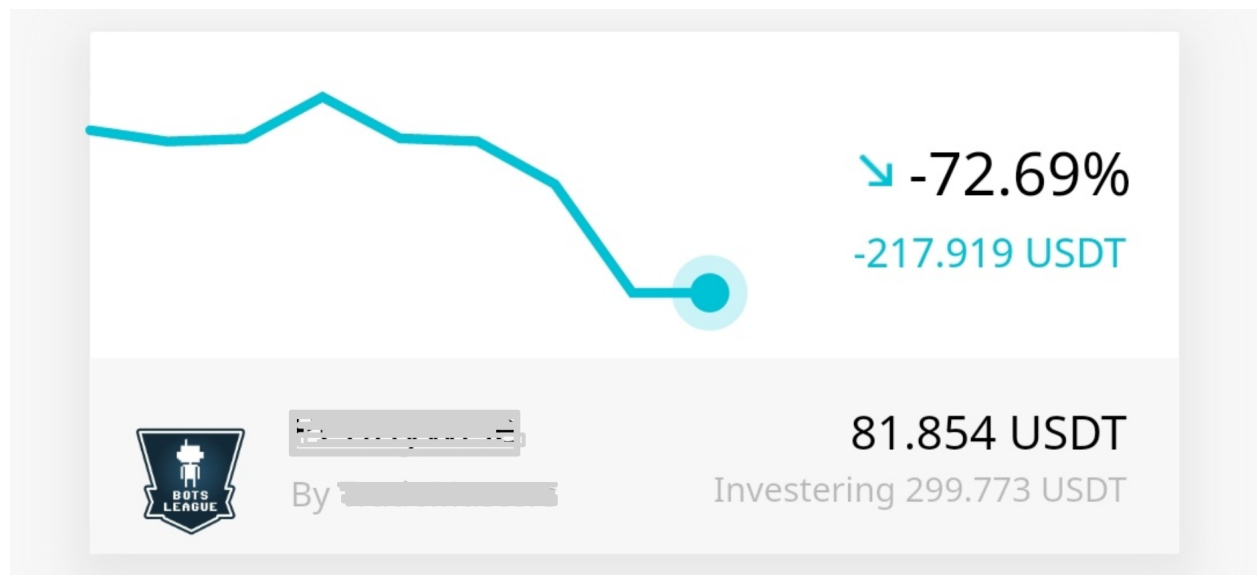
This paper aims to explain the concept behind One Button Trader, describe the current product features, and give insights into the underlying technology.

Problem/Opportunity

It is easy to lose capital when investing in cryptocurrencies.

Given the volatility of cryptocurrency markets, poorly timed acquisition by a beginner investor can lead to catastrophic losses.

Even the existing automated trading solutions (bots) do not save investors from losses. Although aimed to provide better returns than the markets, most such platforms perform poorly during rapid price drops and are generally not efficient even under normal market conditions.



One of the certified trading bots costs -72.69% of portfolio to an investor

Also, considering its often complex and unintuitive user interfaces, trading bot platforms suffer a vast churn rate and generally don't have a positive reputation in the crypto community.

The screenshot displays a trading bot configuration interface. On the left is a sidebar menu with sections: GENERAL (Basic settings, Exchange, Notifications), BUYING (Buy settings, Coins and amounts, Strategy, Trailing stop-buy), and SELLING (Sell settings, Sell strategy, Stop-loss, Trailing stop-loss, Auto close, Shorting settings, Dollar Cost Averaging). The 'Buy settings' option is highlighted. The main panel is titled 'Buy Settings' and includes a link to 'Read more'. It contains several configuration fields: 'Percentage bid' (0,45, Lower), 'Max open time buy' (40), 'Max open positions' (80), 'Max percentage open positions per coin' (1), 'Only 1 open buy order per coin' (toggle on), 'Only buy when there are positive pairs' (toggle off), 'Positive pairs timeframe' (1 day), 'Enable cooldown' (toggle on), 'Cooldown when' (After buys and sells), and 'Cooldown period' (30 Minute(s)). A result box shows 'Result: Max 1 open positions per coin'.

*Quest to set up the winning trading bot — how would you configure it?
(screenshot from one of the apps)*

Here One Button Trader enters the picture.

OB Trader helps cryptocurrency investors grow and sustain their portfolio with easy-to-use online tools backed by AI/ML technology.

It aims to solve two problems:

1. The complexity of crypto investing
2. Low returns of existing trading platforms or while holding cryptocurrencies

OB Trader was made for people who:

1. Want to invest in cryptocurrencies
2. Tired of holding
3. Tired of managing their coins and trades manually
4. Don't have much time to test/experiment with different strategies and tools
5. Looking for a **simple** yet **effective** solution to do the coin management for them

First, OB Trader is adapted even toward less advanced investors. The onboarding process is so straightforward to make sure even a beginner could start using it.

Next, at OB, you can find a variety of pre-trained AI trading strategies adapted for different markets. The single goal of OB trading AI is to outperform the underlying assets so users can get a better return on their existing crypto capital.

Finally, OB Trader works with managing risks and can save from losses. For example, we can ensure the AI is not losing money when markets move downwards by using the embedded stop-loss feature.

Both ease-of-use and performance are vital elements in the OB Trader ecosystem.





Initial Validation

OB Trader has been live since November 2020 and has brought significant value to its investors.

The primary target of OB Trader AI is to outperform the underlying market - provide better returns than simply holding cryptocurrencies.

After gathering seven months of data from 8 AI strategies, 4 of them outperformed the market on average*.

Top-4 AIs had a +1.13-7.91% dominance over the market performance on average per month.

	Bot Type	New Bots Created	Trades Made	ROI Per Bot ?	ROI Per Month ?	ROI Per Trade ?	AVG Bot vs Market ROI Per Month ↑
	Performer	6	560	+10.05%	+2.13%	+0.11%	+7.91%
	Astral v2	7	629	+25.39%	+1.18%	+0.28%	+1.75%
	Horizon	10	1740	+11.77%	+2.25%	+0.07%	+2.77%
	Clipper	49	7746	+39.57%	+6.75%	+0.25%	+1.13%

**data gathered from AI active for longer than three months*

Given the broad scope of opportunities to advance the existing AI, we can further say that these indicators can be improved.

Product

How It Works

To start using OB Trader AI, a user must have an account on one of the cryptocurrency exchanges like Binance, Kraken, or Coinbase Pro.

OB Trader doesn't take a custodial of user's assets and trades directly within an exchange wallet.

OB Trader establishes a connection with the user's exchange account via an API key, which can be generated in the settings of every exchange.

API key doesn't have withdrawal access, so users can always be safe about their funds.

After connecting an exchange, a user must select a market pair and an AI type to start trading.

Here is how the product use case looks like for a regular user:

1. Sign up at OB Trader
2. Connect an exchange
3. Select market pair
4. Specify position size
5. Select an AI and launch it

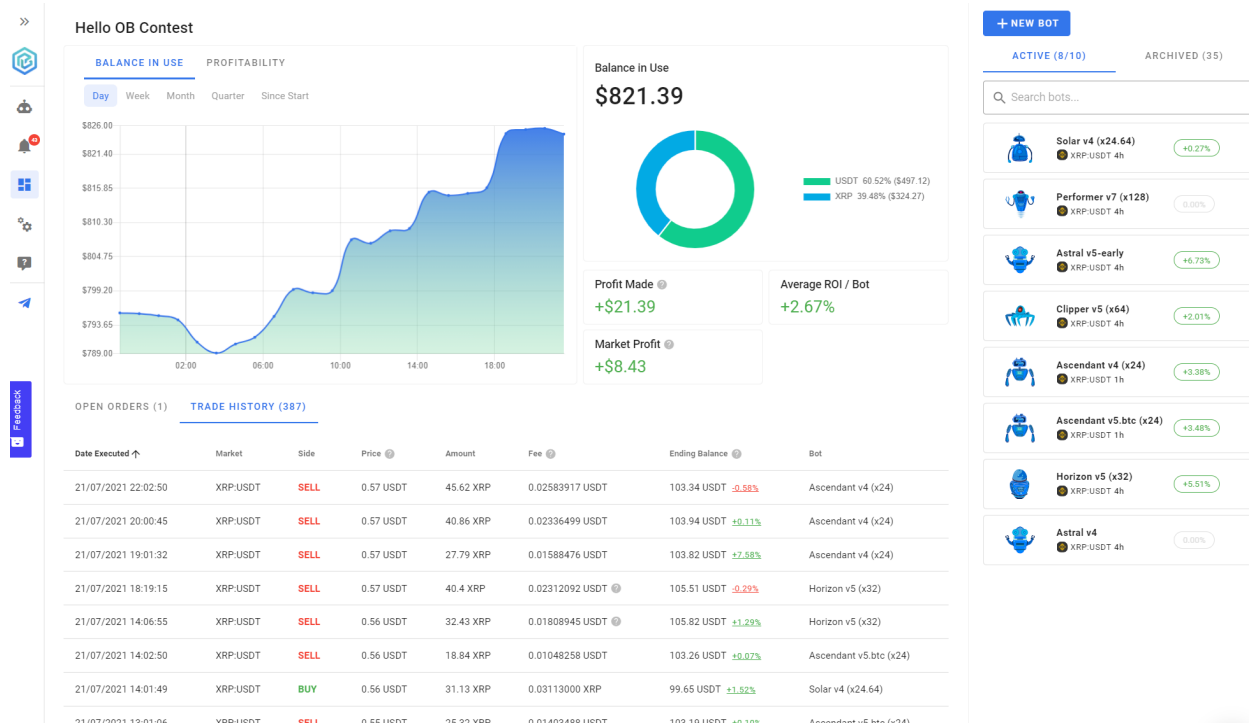
UI

OB Trader was built as a web application to ensure accessibility from different devices, locations, and operating systems.

Simplicity and transparency are the fundamental principles by which the whole OB Trader application was created. OB Trader application is made in a customer-centric way, where usability is one of the biggest priorities.

Dashboard

By looking at the OB Trader dashboard, an investor can assess the status of their portfolio, active AIs, and their performance.



OB Trader dashboard


Launching Your Trading AI

Also, the process of setting up an AI is extremely straightforward.

By including various data about AI's past performance, OB Trader aims to provide sufficient information for an investor to make a decision about which AI to start using.


Create New Bot

Your Exchange


Binance
✓

+ CONNECT ANOTHER

Choose Your Bot


Performer

ROI ⓘ


Last 24 hours **+0.91%**

Last 7 days **+10.95%**

Last 30 days **+12.32%**

AVG Monthly **+15.68%**

MORE ▾


Clipper

ROI ⓘ


Last 24 hours **+0.12%**

Last 7 days **+6.01%**

Last 30 days **+10.28%**

AVG Monthly **+7.62%**

MORE ▾


Ascendant

ROI ⓘ


Last 24 hours **+0.57%**

Last 7 days **+4.70%**

Last 30 days **+7.49%**

AVG Monthly **+4.60%**

MORE ▾


Solar

ROI ⓘ


Last 24 hours **+0.29%**

Last 7 days **+2.24%**

Last 30 days **+7.42%**

AVG Monthly **+10.44%**

MORE ▾




View All


Overall ROI


ROI For ▼ VET:USDT

Select Market

🔍 Top 3 Markets for Performer (Last 30 days)


VET:USDT
+18.52%


ETH:BUSDT
+13.30%


ETH:USDT
+13.30%

View All

The AI Misconception

There is much confusion behind the terminology of *AI* in the trading and crypto space. Some online platforms use the word *AI* when they talk about smart recommendations. You see the words “recommended strategy,” “suggested position size” — the so-called “*AI*” just creates a personalized experience for existing features and trading strategies. But it’s not the actual trading *AI*.

There is very little where *AI* is used as an underlying mechanism for trading (executing trades and placing orders). Instead, most trading bots use technical indicators (like Moving Averages, Bollinger Bands) and their different combinations to make a trading strategy. But all these strategies are based on math functions and formulas — not something you can reasonably call an *AI*.

OB Trader is the first public automated trading tool that uses *only* *AI* for decision-making.

OB Trader is an *AI* that actually trades for you.

AI Decision-Making

In a non-technical way, the AI decision-making logic can be described as the following:

OB Trader AI performs *market scans* — retrieving new market data — every 1 and 4 hours with the new candlestick on the chosen market pair. Based on the latest data and detected price patterns, AI defines if it's a BUY, SELL, or HOLD.

After that, AI calculates the *confidence threshold* — range from 0-100, which includes factors like the amount of risk and the expected change in the market. Then the AI uses the confidence threshold value to define the amount to buy or sell.

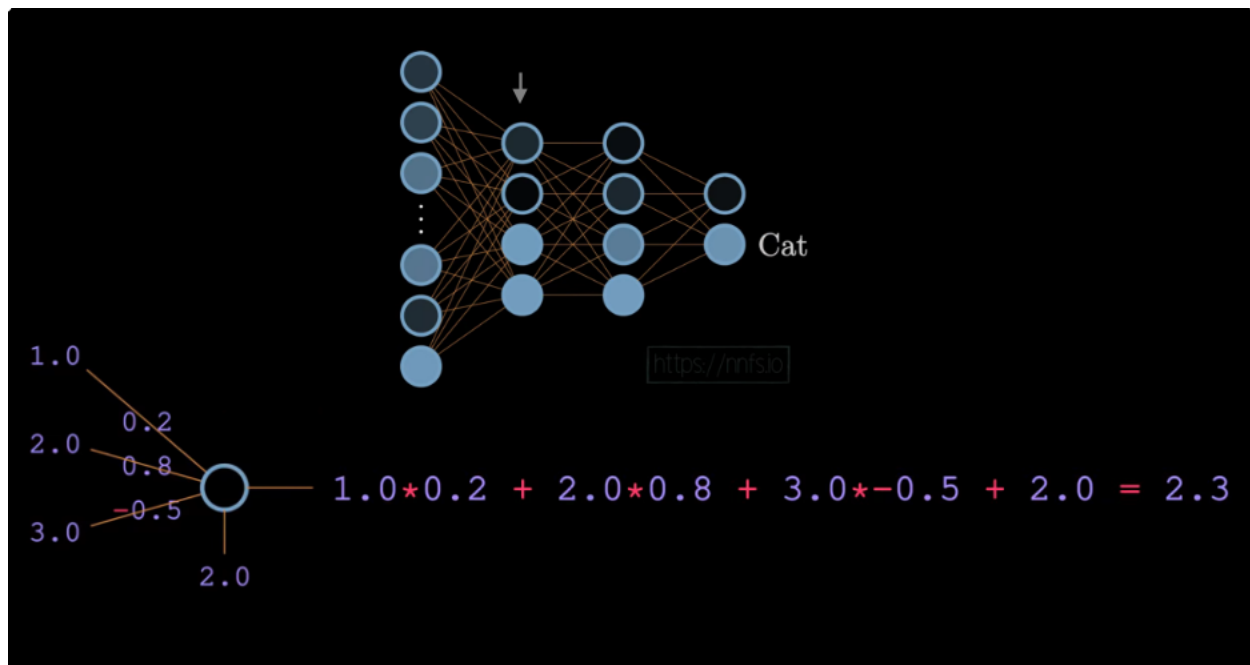
A more in-depth explanation is available in the next section.

AI Foundations

Refresher on Neural Networks

Neural networks (currently) are a form of narrow AI, which means that it's able to learn as specific tasks designed by a human. In theory, any system of non-linear (differentiable) functions could be a neural network, although generally, they adjust scalars, vectors, and tensors as parameters, in order to fulfill the task.

The upsides of neural networks is that they achieve high performance on a variety of tasks, with minimal expert knowledge required. It does have the downside of being hard to interpret by humans because of the complex non-linear relations that are computed. Additionally, it can require quite a large amount of up-front compute in order to train a neural network.



Evolutionary Strategies

For our trading bots, we use Evolutionary strategies to train them, it's a relatively simple concept if you understand Darwinian evolution although because we are using numbers (rather than physical animals) we can make some additional changes to make it more reliable for solving difficult issues.

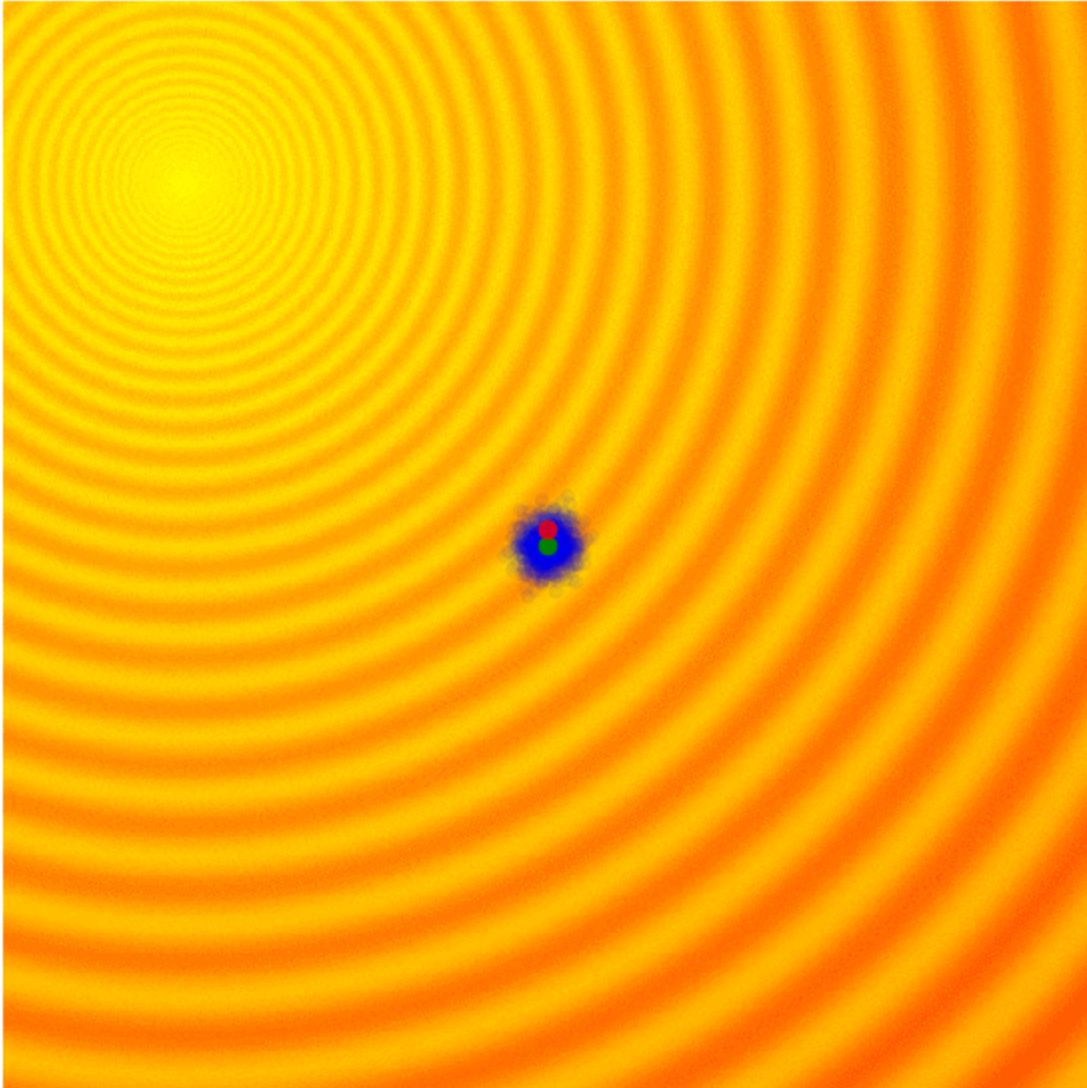
Evolutionary strategies go through 5 (relatively) simple steps:

1. Create a randomly initialized model as the Master model
2. Create N mutated models from the Master model, by applying random noise
3. Evaluate the N models in the environment
4. Adjust the main model by the weighted sum of the reward multiplied by their respective random noise.
5. Go to step 2 until satisfied.

In math terms:

$$m^n = m^{n-1} + L * \sum_{i=0}^{pop} E(m^{n-1} + J_i) * J_i$$

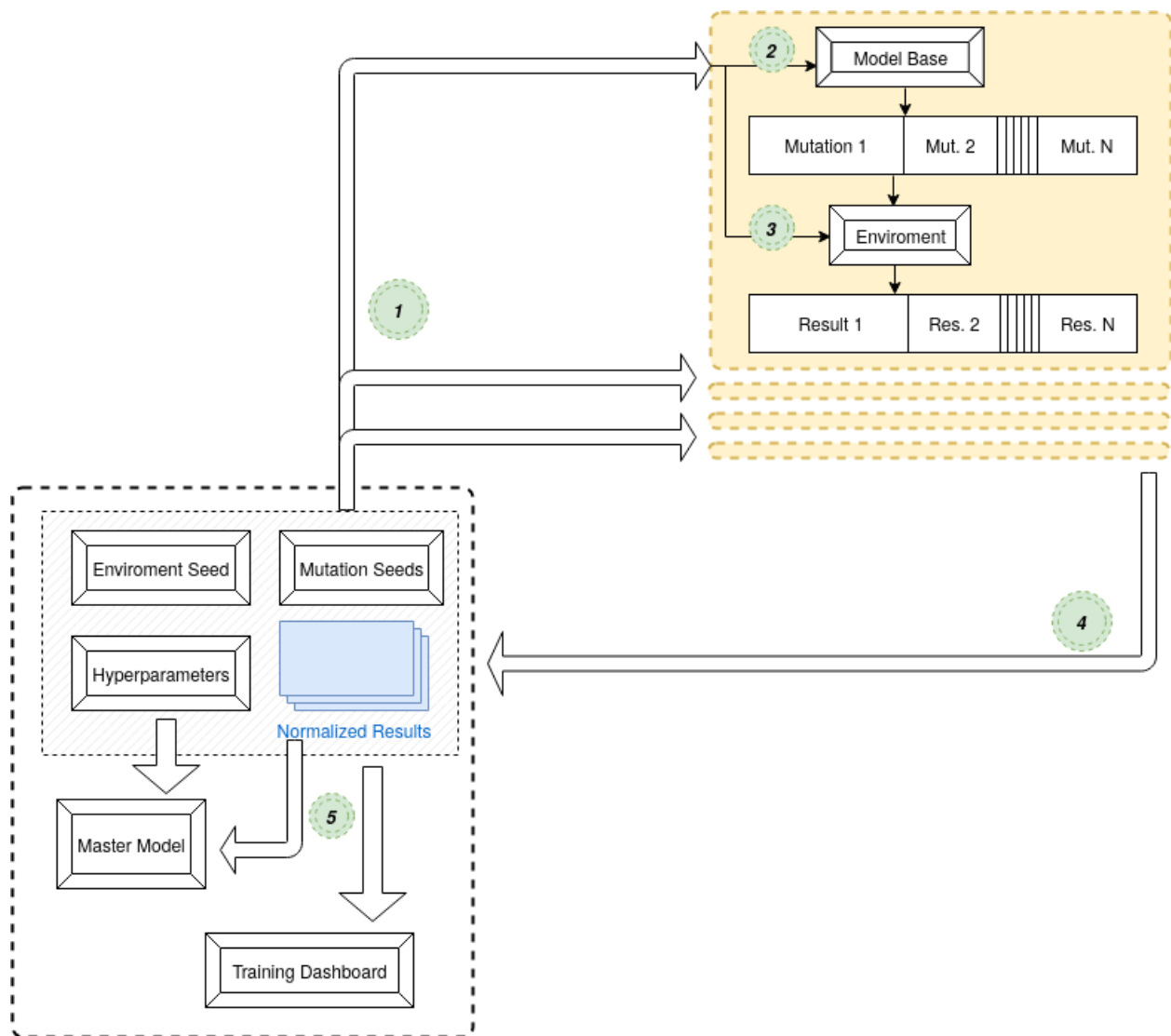
- E is the evaluation function (returns normalized reward)
- J is a deterministic noise
- m is the model at a certain time-step n
- L is the learning rate.



Demonstration of the evolutionary strategies algorithm.

The image is animated. See the animated version [here](#)

We have written our own implementation for scalability and extendability. It is available under MPL-2.0 at <https://github.com/ob-trading/portable-es>



Why evolutionary strategies?

There are a lot of reinforcement learning algorithms (PPO/TRPO/DQN/Dueling-DQN/etc), which allows for training against an environment (in our case a simulated market). The main issue with these are they either:

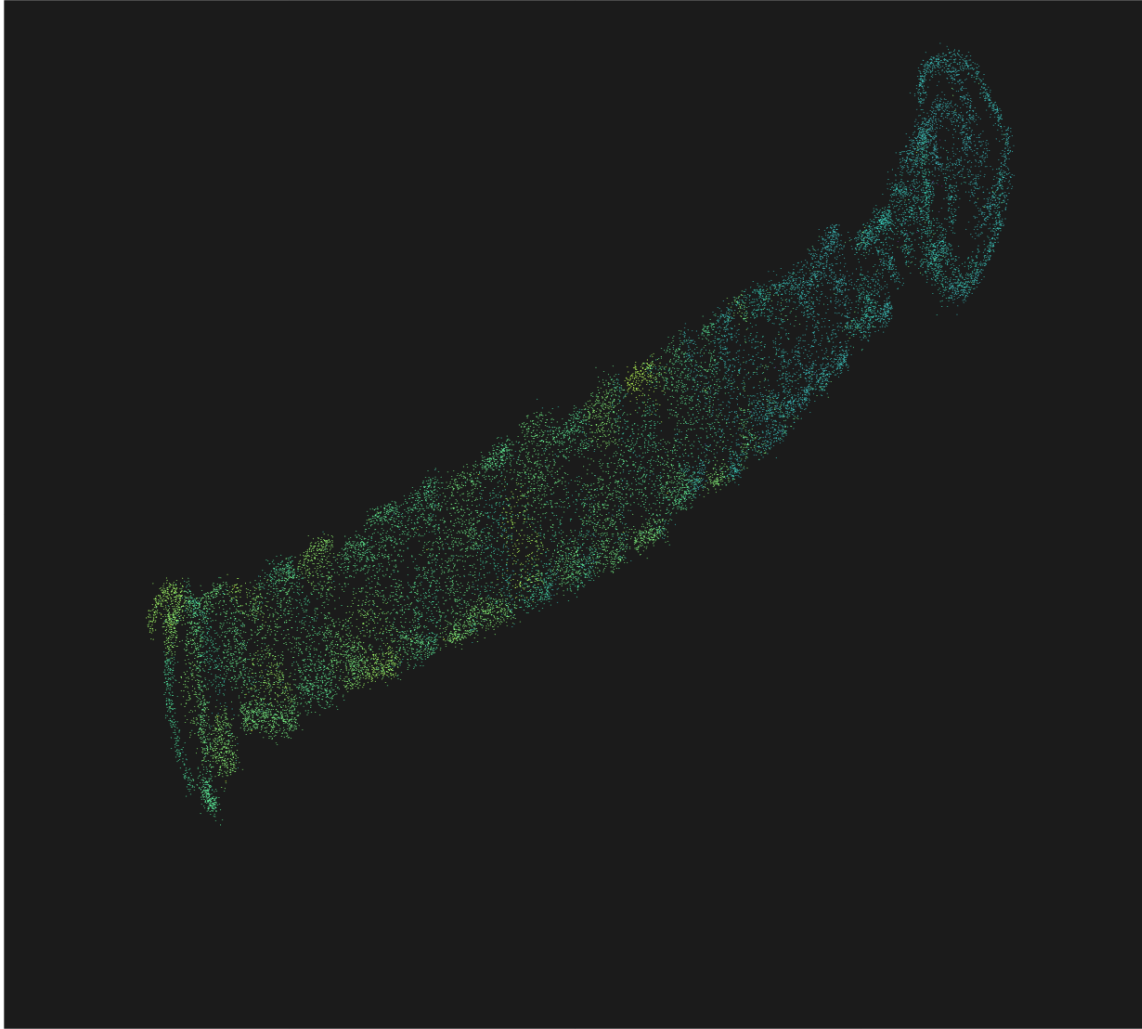
1. Expect the agent to have a meaningful effect on the environment
2. They estimate the expected future value based on the current state/action
3. They expect rewards to be given for a certain action within relatively short period of time

While in theory it's possible to make this work for trading, these algorithms are likely to mismatch our goal. Since we do not expect to have a major effect on the market when a strategy is released, and the future value cannot be accurately estimated without having knowledge of the future.

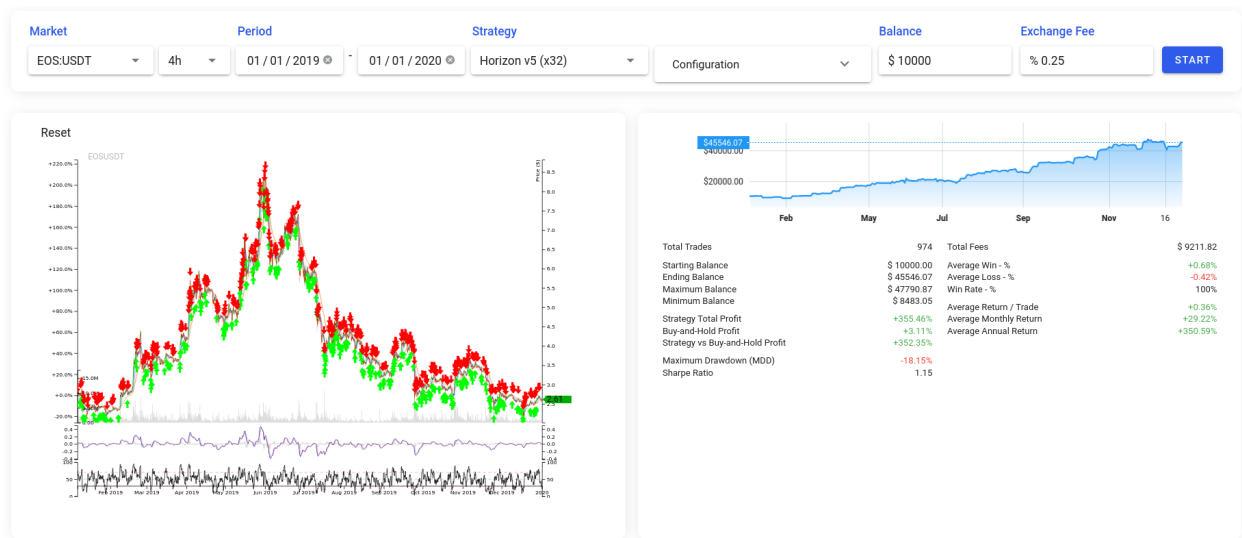
Evolutionary strategies bypasses all of these issues since the reward are a single scalar for each episode, the rewards we give can be calculated at the end of the episode so it doesn't run into issue #3 and #1.

It compares perturbed models on the same simulation to get the direction rather than having the estimate it, which fixes issue #2.

Now it's possible that one of the regular RL algorithms fixes some of these issues as well, however because of their modus operandi it would not be easy to do.

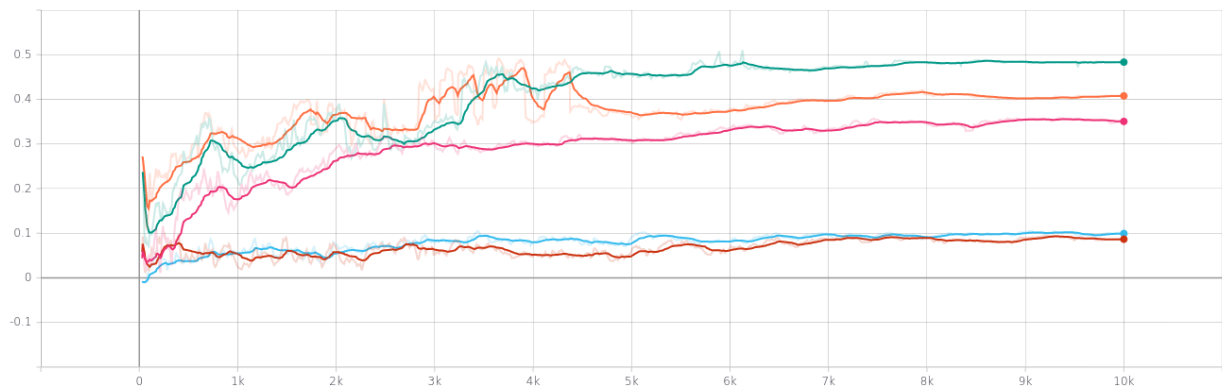


Visual representation of AI training process.



Backtesting results of the trained model

V2



Different AI models' V2 metric during training

AI Models

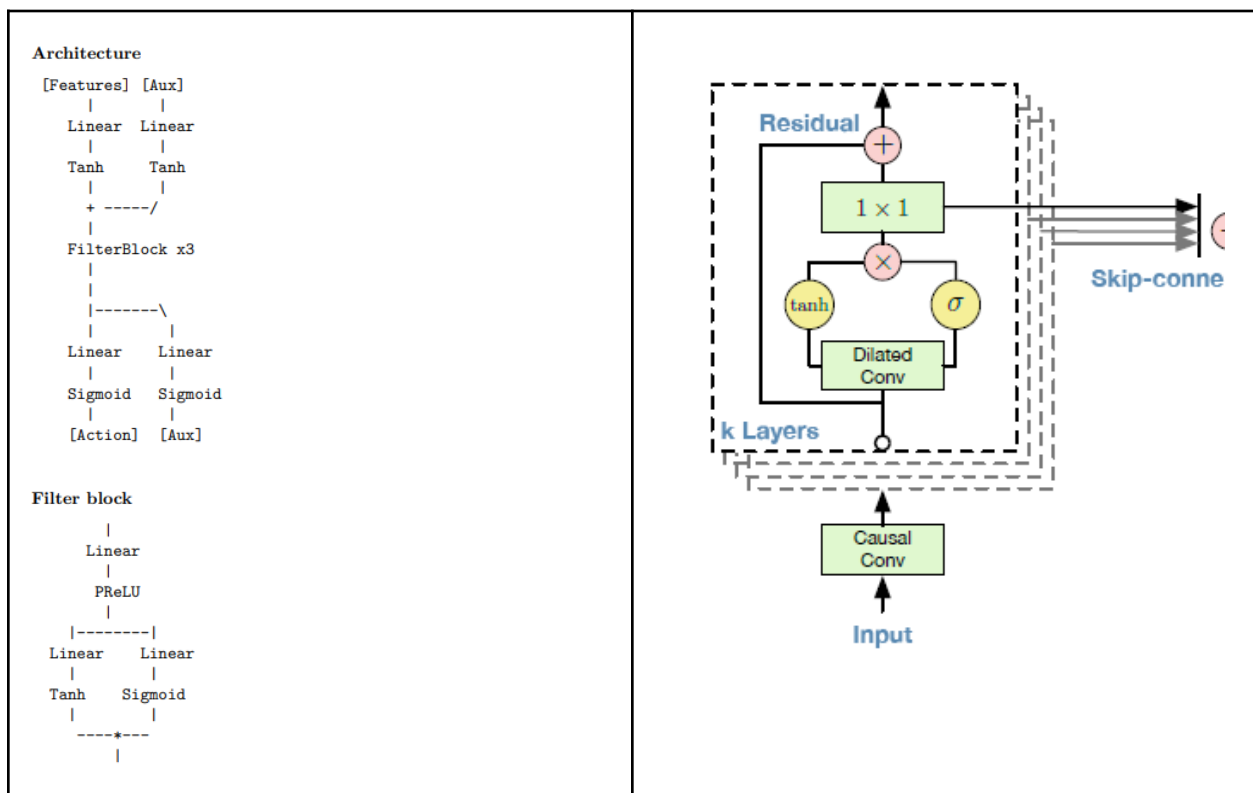
All of our current models are neural networks, and are based on battle-tested architectures.

Astral (Filter)

Astral was one of our first models based on concepts from the [WaveNet paper](#), it uses 2 parallel Linear layers in with *sigmoid* and *tanh* activations respectively to create a filter. In all of our deployed strategies, this model has 3 FilterBlocks and 2 projection layers for the input/output.

FilterBlocks have 3 feed-forward layers. Each contains a *PReLU* shared linear layer which goes into the *tanh* and *sigmoid* linear layers respectively, afterwards these are re-combined by a Hadamard product.

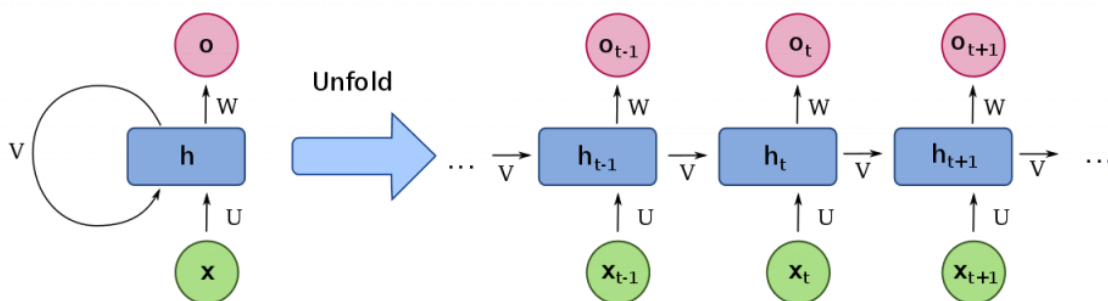
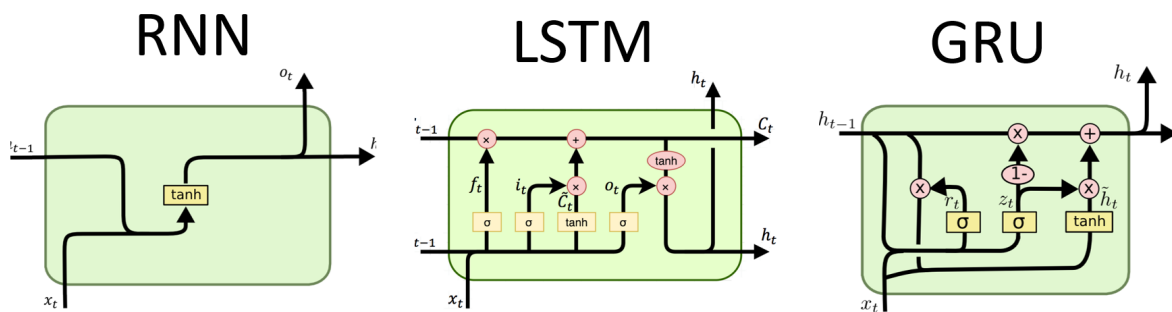
These models have mostly be replaced by the more recent Performer models, because of the fact that the Performer can in theory make more complex computations. Filter models do have the benefit of allowing for a better compute-to-memory usage ratio than Performer, however for raw trading performance this is a moot point.



Horizon/Ascendant (GRU)

The horizon and ascendant models use recurrent neural networks to take actions, specifically a multi-layer [GRU](#). They keep their internal state between actions, allowing them to re-use some computation from previous steps.

The strategies that we have deployed typically have 3 GRU layers with different hidden state and channels dimensions depending on the input data. Each market scan the model gets a full window of price data (e.g. 64 of most recent ohlc), these are used as a single step in the GRU. This allows for more efficient computation as well as creating an inductive bias for applying historical data to itself with an offset (as is used in many technical indicators).

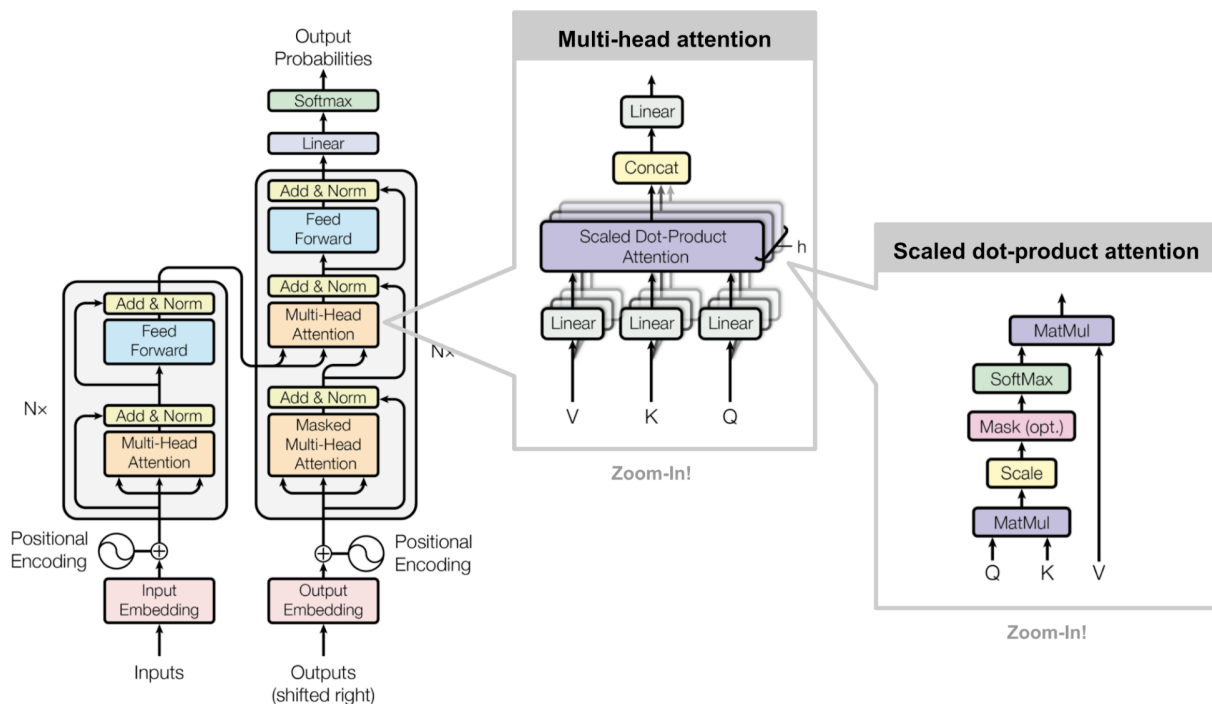


Performer

The performer model is a direct adaptation from the [Performer paper](#) which uses FAVOR+ to approximate attention used in [Transformers](#). It is otherwise equivalent to the Transformer architecture.

Transformers are currently the most popular/promising type of neural network in most fields of study (NLP, Timeseries, and even some Vision). This is due to it's extremely general nature, it can learn many different types of tasks using approximately the architecture.

The strategies we have deployed typically have 2 layers, and has learned positional embedding. It also has 2 projection layers for the input/output layers. The models are otherwise the same as described in the paper.



Complex Adaptive Systems (Discussion)

The first thing which rationalizes our choices is the fact that markets are complex adaptive systems, which means that any changes in the system *can* change dynamics long term and may influence other parts of the system in a feedback-loop. The agents in the system collectively influence the system, which in turn makes the agents adapt, which changes the system again.

Any such agent acting in the system in a static way will eventually fail to perform; if this agent's reasoning is publicly available other agents will also be able to take this into account speeding up the issue.

This effectively means that using purely technical indicators is extremely unlikely to hold performance in the future (if they ever worked at all), as any consistent causal pattern will be adapted to, especially if this pattern is publicly available.

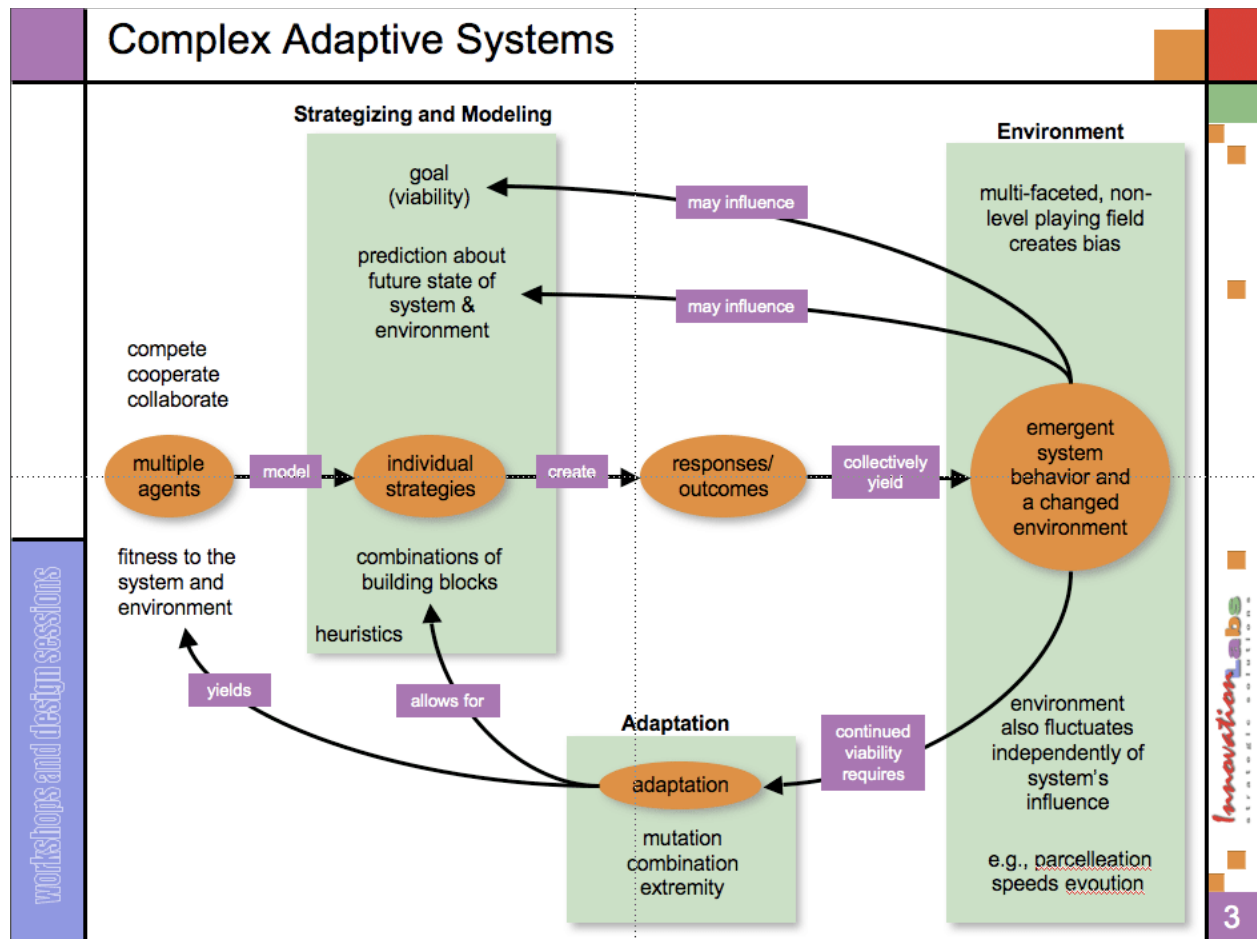
Humans can adapt; research into a company's fundamentals and environment can provide certain expected prices with certain probabilities (for example [Morgan Stanley's analysis on page 6](#))

However such research requires vast human resources and can cost a customer [up to 350,000\\$ per year](#); and this doesn't even provide you with a strategy, just the probabilities/expectations of certain events happening.

So for everyone who isn't a stock/crypto-expert billionaire, there will need to be a different solution.

For us this meant going to neural networks which directly interface with the market. In it's current state it only handles the direct environment (the market), however it is able to adapt to changes in the market and provides you with fully automated way of using the strategy. We can also regularly update our models (or even do it with online learning) to adapt to changes that the model itself doesn't account for. In the

future the inclusion of external features (such as social media, news articles, performance reports), will allow for even more internal adaptability.

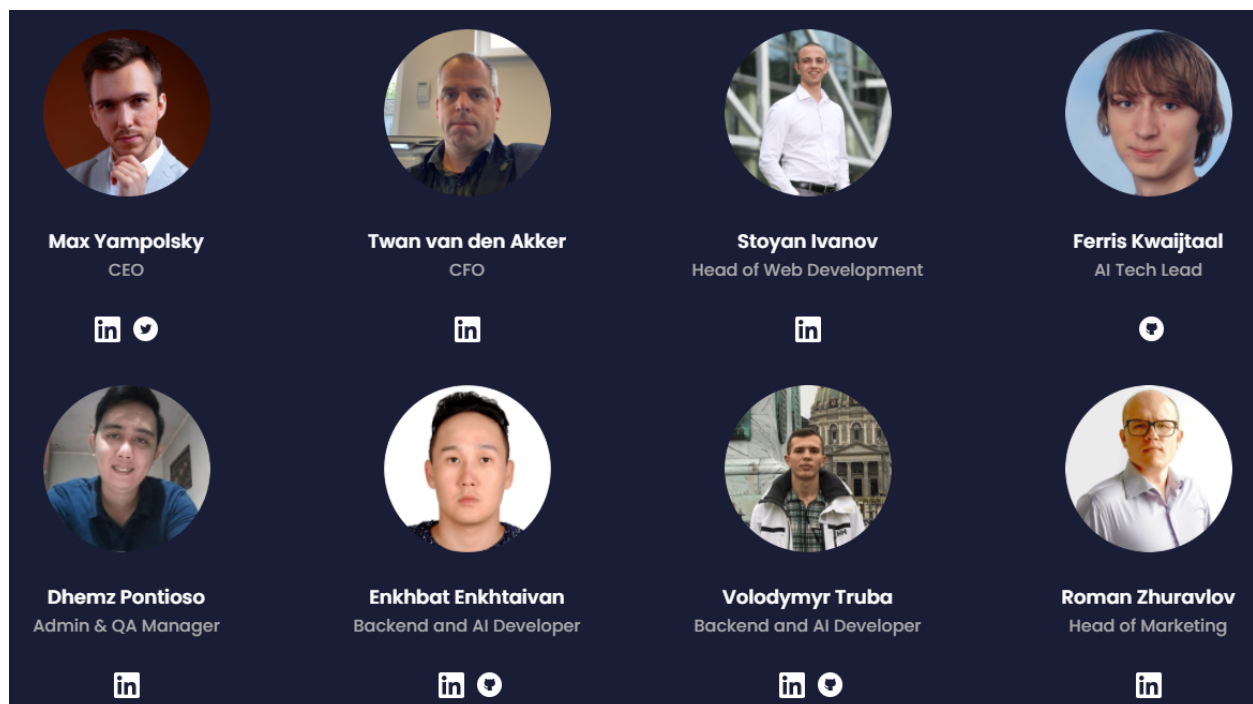


Team

Team at OB Trading consists primarily of technology experts with a heavy focus on AI/ML development and product design.

With a track record of dozens successfully implemented digital apps (web & mobile) including in the fintech space, the team has proven to be suitable for developing AI trading products at the commercial level.

The team at OB has access to vast networks of potential investors, advisors, and local communities of crypto enthusiasts in the Netherlands.



Invest

OB launches its own token. Read more about One Button Trader tokenization [here](#).

Read the tokenomics paper [here](#).

References

- <https://github.com/ob-trading/portable-es>
- <https://blog.devdroplets.com/diving-into-evolutionary-strategies/>
- <https://blog.otoro.net/2017/10/29/visual-evolution-strategies/>
- <https://lilianweng.github.io/lil-log/2019/09/05/evolution-strategies.html>