# Using Advantage Actor Critic (A2C) to build a trading model capturing seasonal behaviour in a basket of correlated stocks

Thomas Wong

# Speaker

**Thomas Wong**

Imperial College London PhD in Mathematics (2019 - now)

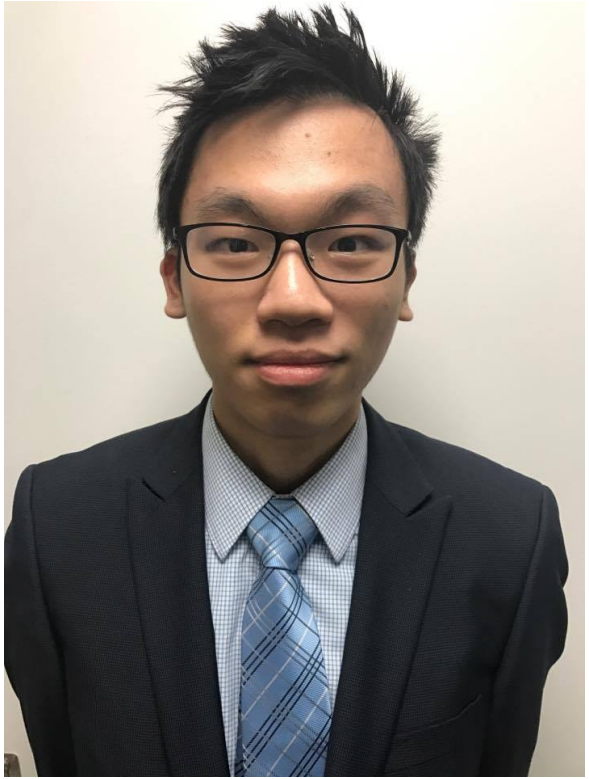
Imperial College London MSc in Bioinformatics and Theoretical Systems Biology

Imperial College London BSc in Mathematics

Summer Intern at Bloomberg (Global Data) and Goldman Sachs (Engineering)

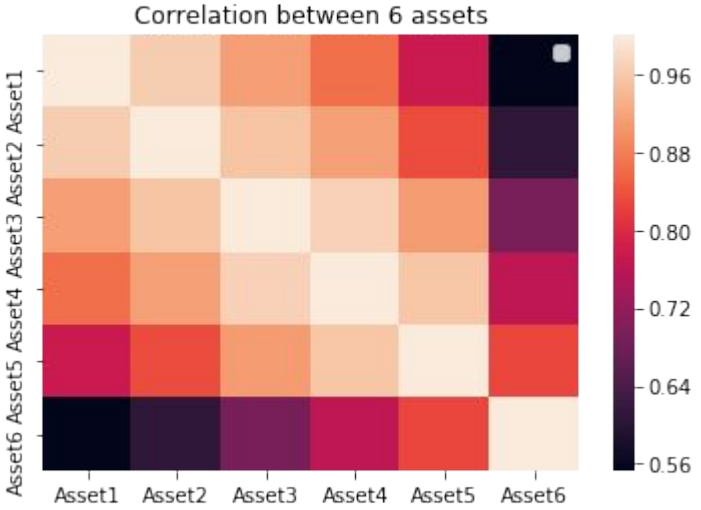
Head of Technology at Imperial Algorithmic Trading Society

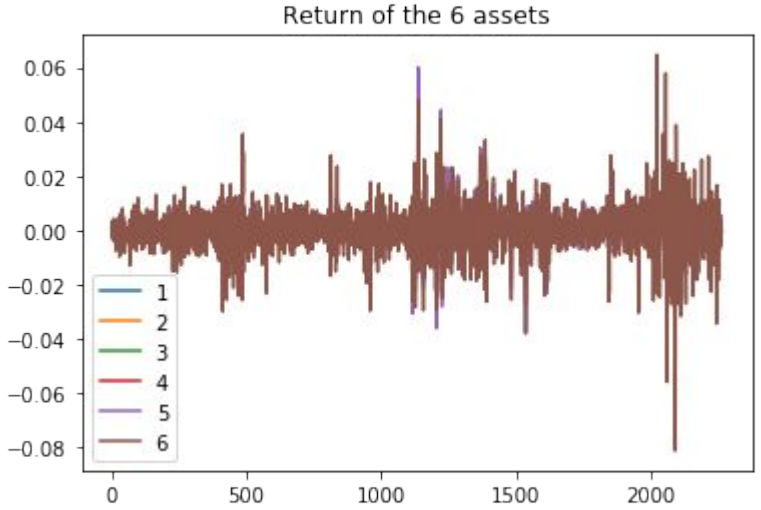
LinkedIn: https://www.linkedin.com/in/thomas-w-27b463110/



# Problem

Historical data given for around 2250 trading days of 6 stocks that are highly correlated. There are 3 liquidity events each month on average.



# Aim

The aim is to formulate a trading strategy such that

- Trading is only allowed in the liquidity windows
- Asset exposure is limited by historical volatility, for example in the volatile market between 800 to 1200 days, maximum allowed position is lower than average
- The overall strategy optimise annual return + 0.5* max drawdown
- The above optimisation penalises drawdown further than sharpe ratio

# Buy and hold strategy

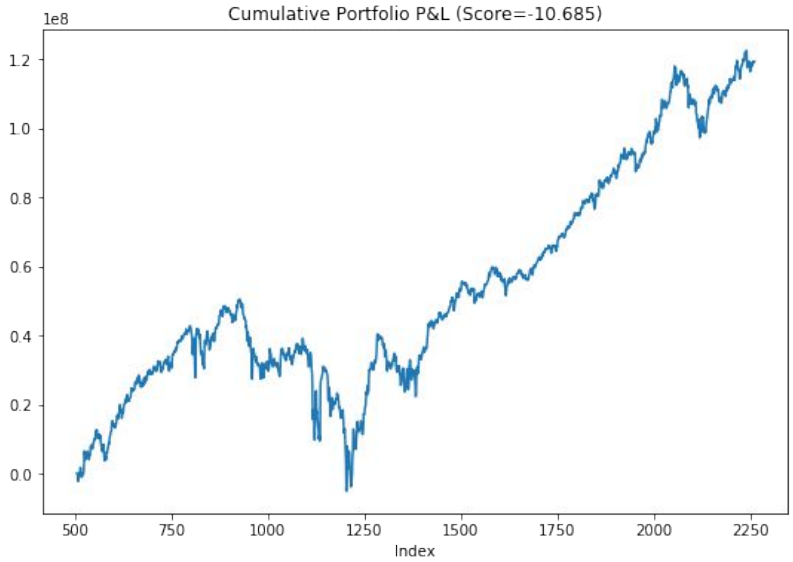



Figure 1: PnL curve of buy and hold strategy. Market environment is different in the first and second half of the data given.

A simple buy and hold strategy suffers from the drawdown in different periods.

Our strategy should be able to learn the turning points of the market, such as when the market changes from a upward trend to downward etc.

Rebalancing is allowed for every 10 days on average, our strategy needs to learn trends that can last for 2 weeks to a month

# Quantitative trading strategies

Statistical Arbitrage: (Generally) Market neutral strategy aims to utilise mispricing in a basket of assets that shows statistical relationships

- ❖ Not used here as we are only allow to trade at a low frequency
- ❖ Low frequency trade cannot capture the mean reversal behaviour between our assets

Trend following: Leverage sustained upsides and downside in the market

- ➢ In a sideway market or if market trends changes frequently, this strategy loses money.
- ➢ Historical data shows there are long term trends that we can capture

# Reinforcement learning

Reinforcement learning has been used widely in robotics control and games to solve various tasks with a clear definition of environment and reward.

Applying reinforcement learning in finance which is a stochastic environment with no clear definition of reward is a new challenge

Reinforcement learning works better when combined with other approaches

Deep learning: Use of Neural networks to extract features and make predictions

Online learning: Continuously train and update model as new data comes in

# Reinforcement learning in Education

Kumon, a famous private education company in Japan has used the concept for reinforcement learning for ages before DeepMind is founded.

Students are given a set of homework to do and they are required to repeat doing the same set of homework until they can answer majority of questions

Instructors provide feedback by grading the homework

# Does this method work?

Yes

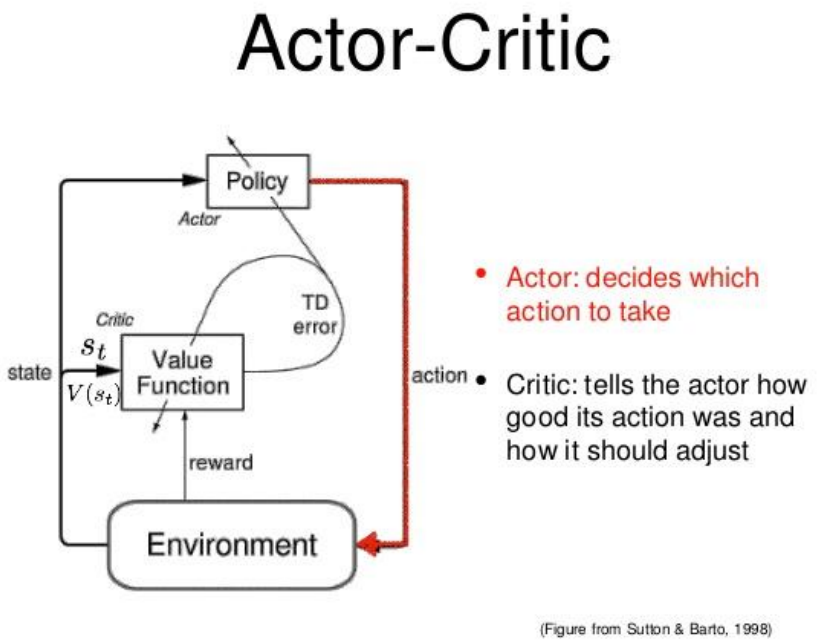
- Students who have little knowledge in mathematics shown great improvement in a short time (No prior knowledge or model assumptions)
- Standardised approaches allows instructors to monitor student performance easily

No

- Can students do well in other similar tasks (Transfer learning)
- Do students really learn the concepts of mathematics or they they just learn some ad-hoc rules in problem solving (Blackbox)

# Definition

Advantage Actor Critic (A2C) is an on-policy gradient method that uses two neural networks, actor and critic to learn the policy. The actor is the neural network which represents the policy while the critic is the neural network to learn the value function.



# Implementation

## Feature Extraction

Lookback period is set to 252 as the number of trading days in a year.

31 features including returns, levels, sensitivity and exposure are used

Features are normalised and smoothen by a MA window of 5

## Model Training

An implementation of A2C agent provided by stable-baselines is used

Different NN architecture of the actor and critic network are used

Reward is defined as the improvement of the score at a given (state,action) pair

## Model validation

Tensorboard is used to monitor the training process

Best models are saved during the training process. Training is stopped after periods of no improvement or 48 hours

Validation is performed on the whole dataset for the saved models.

# Feature Extraction

Technical indicators

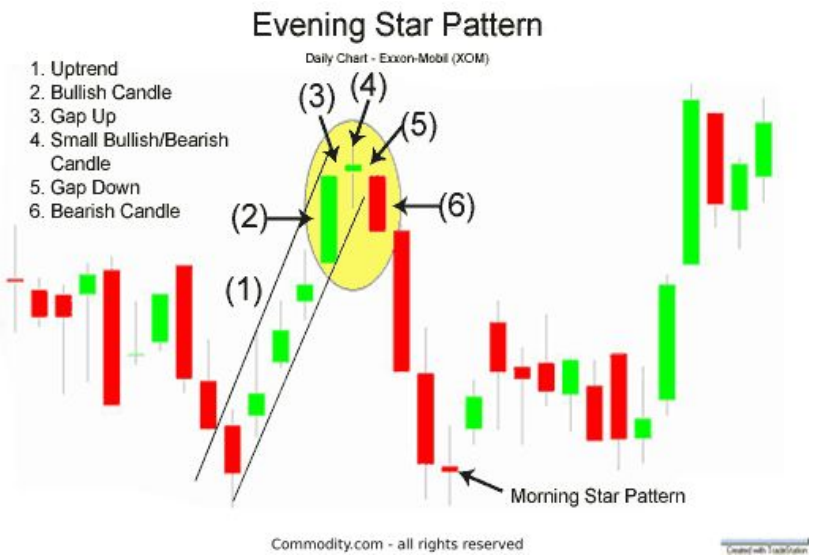
- ❖ Time Series based methods from experience of many traders
- ❖ Not used here because we do not have OHLCV data given

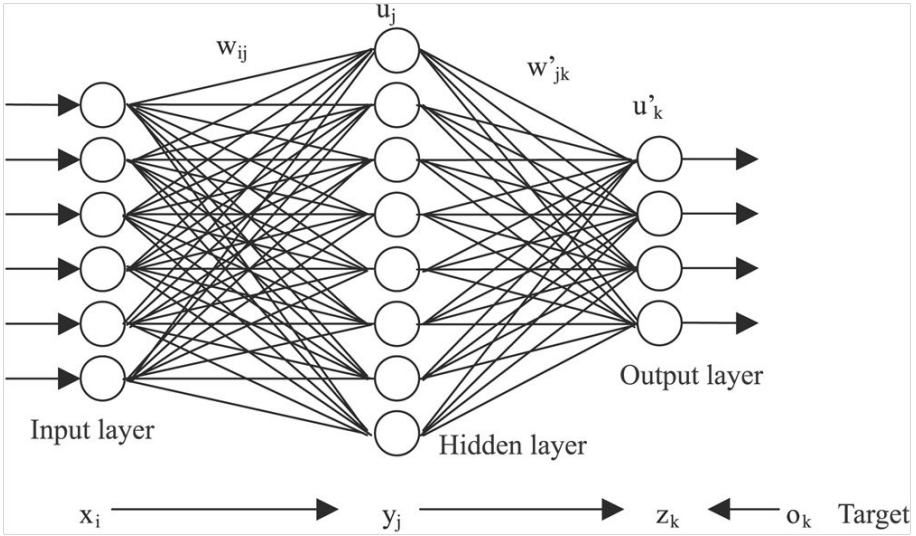
Neural network:

- No modelling assumptions or human bias
- Unsupervised method to extract features from historical data
- RNN and CNN has shown to be successful in extracting features in image processing

# Feature Extraction

Neural network can extract price movement patterns like candlestick charts



# Feature Extraction

Lookback period: 252 trading days (a year)

Features: 31 features normalised and smoothen by MA window of 5

Features used: Index and day of the month, can_trade (boolean variable), tradable_events (1x3 vector), **reward of the agent,** returns, levels, sensitivity and exposure of the 6 assets.

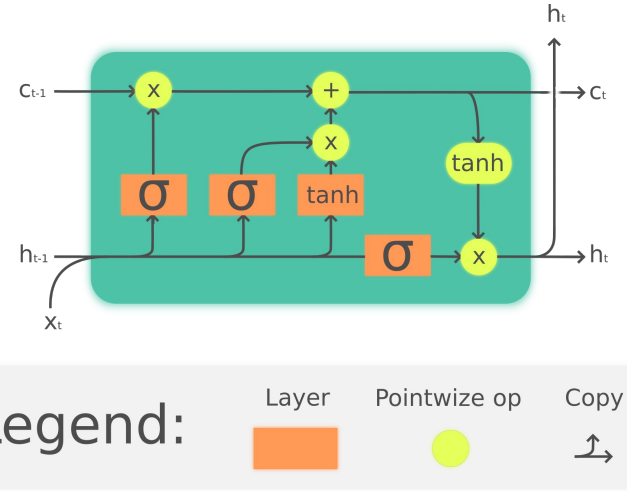
Keeping a history of previous reward of the agent allows the agent to learn whether it should change its current policies

# Model Training : Neural Network Design

LSTM is a type of RNN that has been used in different time series models. Compared to a fully connected layer, LSTM has fewer parameters to train and is able to capture dependence in data.

Example of an LSTM unit.

An LSTM unit consists of a cell (memory part) and three gates, input gate, output gate and forget gate
LSTM avoids the vanishing gradient problem in traditional RNN



# Model 1 Design

Actor and critic network shared the same architecture

Input layer: 252*31 features
Dense layer: 256 units, tanh
LSTM layer:  14 units, tanh
Dense layer: 256 units, tanh
Output layer: 6 units

First layer is to extract latent features equivalent to technical indicators
LSTM layer is to extract long term temporal patterns
Final layer is to assign weightings to the temporal patterns for output asset weights

# Model 2 Design

Actor and critic network shared the same architecture except for the last one

Input layer: 252*31 features
Dense layer: 128 units, tanh
LSTM layer:  28 units, tanh
Dense layer: 64 units, tanh
Dense layer: 16 units, tanh, separately for actor and critic network
Output layer: 6 units

# Results

I have trained two models with similar performance. Model 1 is considered as my winning entry.

| | In-sample score | Worst In-sample score | Out of sample score |
|---|---|---|---|
| Model 1 | 63.03 | 60.76 | 51.8 |
| Model 2 | 60.65 | 49.71 | 50.5 |

In-sample score: performance on days 600-2259
Worst In-sample score: worst performance on days 700-2259
Out of sample score: performance on the held-out dataset

# Model 1

| | |
|---|---|
| Strategy Score: | 63.031 |
| Cumulative P&L: | 507,013,273 |
| Max P&L Drawdown: | -19,293,222 |
| Transaction costs: | -12,108,428 |
| Cumulative Return: | 507.01% |
| Max Return Drawdown: | -19.29% |
| Annual Return: | 72.68% |
| Annual Return Vol: | 20.62% |
| Sharpe Ratio: | 3.53 |
| Traded Days: | 547 |
| Total Turnover: | 89,950 million |

Backtest results from trading model 1 from 500 to 2259 trading day.

Note that most of the drawdown is caused by transaction costs

# Model 1

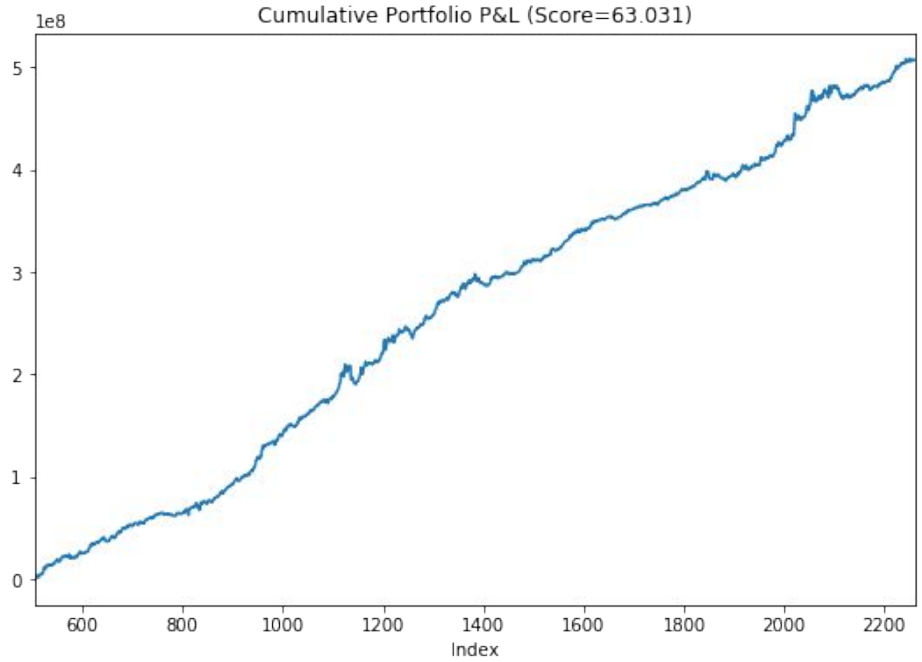



Figure 2: PnL Curve of trading model 1

It is able to capture long term market trends. Notable drawdowns occur when the market is volatile around 1100 and 2050

# Model 2

| | |
|---|---|
| Strategy Score: | 60.654 |
| Cumulative P&L: | 485,475,223 |
| Max P&L Drawdown: | -17,871,691 |
| Transaction costs: | -9,826,271 |
| Cumulative Return: | 485.48% |
| Max Return Drawdown: | -17.87% |
| Annual Return: | 69.59% |
| Annual Return Vol: | 20.72% |
| Sharpe Ratio: | 3.36 |
| Traded Days: | 547 |
| Total Turnover: | 43,432 million |

Backtest results from trading model 2 from 500 to 2259 trading day.

It has a lower turnover than model 1 but a slightly higher maximal drawdown.

This demonstrates there is a trade off between keeping and changing the current exposure, keeping the exposure would take more market directional risk while changing the exposure incurs transaction costs.

# Model 2

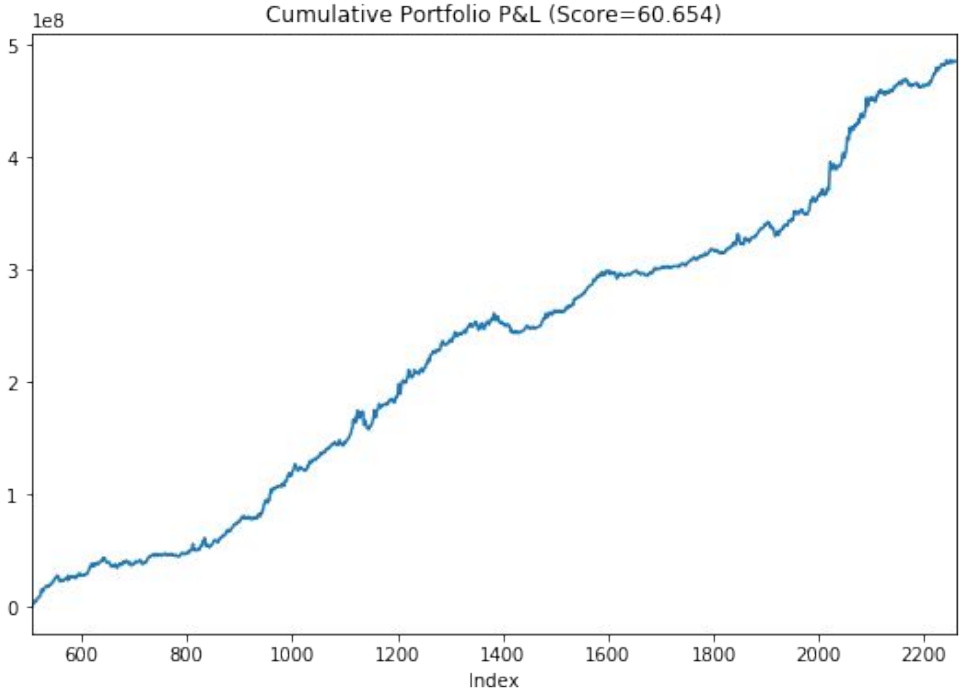
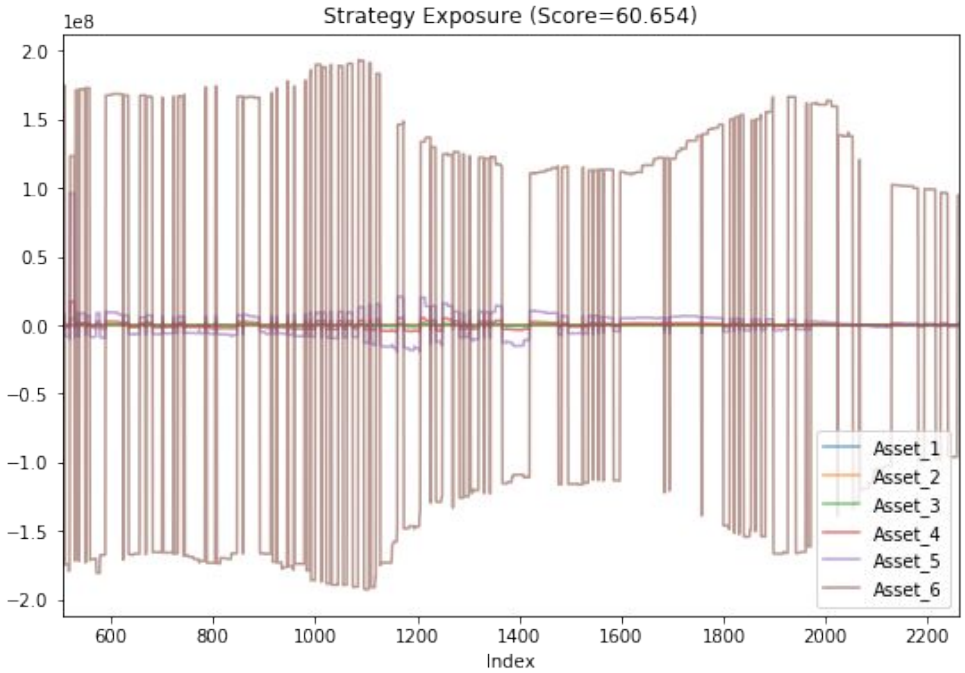



Figure 3: PnL Curve of trading model 2

The PnL curve is very similar to model 1. Both models are very similar by design and trade decisions made are similar.

# Results




Asset exposure alternates between long and short with volatility adjusted.

Asset exposures are highly correlated

Volatility scaling can be used to improve return (Overweighting / Underweighting risky assets)

# Further Work

Hyperparameter search

- Discount factor of reward is set to 0.95 for my models. Choosing a suitable discount factor can improve convergence
- Different neural network designs: use genetic algorithm to search for the optimal combination of dense and LSTM layers for the actor and critic networks

Online reinforcement learning

- Retrain the model with most recent price data allows model to adapt to changes in market environment

# Algorithm

## Algorithm for A2C [1]

Hyperparmeters defined B: Batchsize, $V_\phi^*$ critic neural network, $\pi_\theta$ actor neural network, $\alpha$ critic loss scaling, SGD optimizer

Intialise weights $\theta, \phi$

In each training step,

1. A roll-out of batch size B using the current policy $\pi(\theta)$
2. For each transition $T$ from the roll-out compute advantage estimation $A^\pi(T) = r' + \gamma V_\phi^\pi(s') - V_\phi^\pi$
3. Compute target $y(T) = r' + \gamma V_\phi^\pi(s')$
4. Compute critic loss Loss $= \frac{1}{B} \sum_T (y(T) - V_\phi^\pi)^2$
5. Compute critic gradients $\nabla^c = \frac{\partial Loss}{\partial \phi}$
6. Compute actor gradients $\nabla^a = \frac{1}{B} \sum_T \nabla_\theta \log \pi_\theta(a|s) A^\pi(T)$
7. Make a step of gradient descent using $\nabla^a + \alpha \nabla^c$

# Why use A2C

- Action Space: A2C supports continuous action space while DQN does not. In my model, I use a 6-dimension action space between -1 to 1 to represent the asset weights
- Policy based methods performs better than value based methods in a stochastic environment
- Using the Advantage function instead of value function lowers the variance on updates
- Asynchronous Advantage Actor Critic (A3C) a variant of A2C is scalable and runs parallel environment to update the global network

# References

[1]: S.Ivanov, Modern Deep Reinforcement Learning Algorithms arXiv:1906.10025
https://www.researchgate.net/publication/333993458_Modern_Deep_Reinforcement_Learning_Algorithms