

Forecasting Stock Prices from the Limit Order Book using Convolutional Neural Networks

文章摘要

此文章用了CNN模型，輸入四百五十萬筆高頻率限價單(LOD)的資料，並在不同時間範圍下，預測股票價格的變動方向。同時也將CNN模型的訓練及預測結果，跟SVM, MLP兩個模型相比，發現CNN的表現都較出色。

研究背景

現今大多數交易皆由電子系統進行，且自動化交易佔了很大的比例，這些交易包含了大量的數據。若能有效分析這些數據，就能幫助投資者發現價格變動的模式，從而提前做出決策，實現盈利或避免市場異常波動。

前情提要

- 高頻率限價單(High Frequency Limit Order Date, OLD)是指：投資者在特定價格和數量進行買賣資產的訂單。
- 其他論文很常取的數據是股價的開高低收(HOLC)。但這樣取樣方法，只會留下特定時段的關鍵價位，雖然能夠看出趨勢漲跌，但卻省略了很多的微小信息，例如交易量、市場情緒等。因此這篇論文選擇預測中間價格(mid-price).

- 對於 買單 (bid side) :
 1. 第一筆買單的價格：\$100 (價格數據點)
 2. 第一筆買單的數量：50股 (數量數據點)
 3. 第二筆買單的價格：\$99 (價格數據點)
 4. 第二筆買單的數量：30股 (數量數據點)
 5. (依此類推，直到第10筆買單)
- 對於 賣單 (ask side) :
 1. 第一筆賣單的價格：\$101 (價格數據點)
 2. 第一筆賣單的數量：40股 (數量數據點)
 3. 第二筆賣單的價格：\$102 (價格數據點)
 4. 第二筆賣單的數量：20股 (數量數據點)
 5. (依此類推，直到第10筆賣單)

為何選擇預測中間價格(mid-price)?

- 中間價格實際上並不會有成交的情況。
- 能夠作為預測價格變動方向的指標，且中間價格對於異常價格波動的影響較小。
- 反映及時價格趨勢及力量。

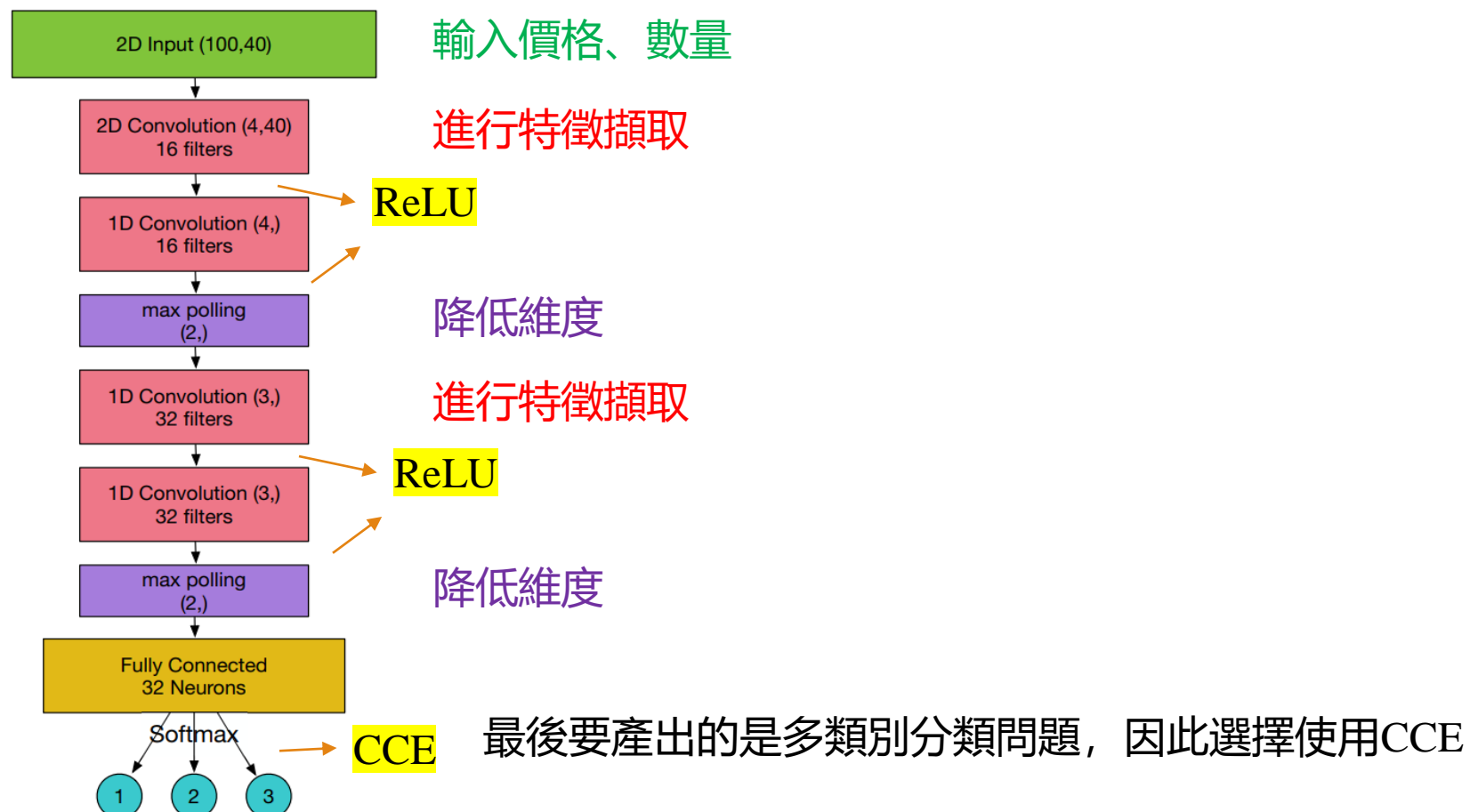
舉例說明：有一筆訂單，買方出價100，賣方出價101，則中間價格為100.5。

若在下個時間步長，中間價格提高至101，代表買方願意提高出價，市場需求增強。

研究方法

- 這篇文章使用了**那斯達克北歐股票市場**(Nasdaq Nordic Index)的2010年6/1~6/14的交易資料，總共有450萬筆限價單資料。將資料訓練集與測試集分成 7: 3。
- 輸入資料總共有100個時間步長(time step) $k=100$ ，每個時間步長又包含10個買單、10個賣單，當中又有**數量**跟**價格**兩個值。因此，輸入的資料維度是 100×40 。
- 接著將輸入的**資料標準化**，避免因為價格資訊落差過大，提升模型訓練速度、程度及泛化能力。
- 引用**ReLU**作為激活函數，並使用**類別交叉熵**作為損失函數，產出1, -1, 0這三個結果，最後使用**ADAM優化器**，達成最小化損失函數，並將誤差反向傳播更新權重。

流程展示



文章內的虛擬變數

- $P(t)$, $P(t+k)$: P_t 代表的是當前時間點(t)的中間價格, $P(t+k)$ 代表的是未來時間點($t+k$)的中間價格
 - 例如: $P(10)$ 代表當時時間點 $t=10$ 的中間價格; $P(15)$ 就是時間點 $t=15$ 的中間價格, 能夠彼此比較
- $mb(t)$, $ma(t)$: mb 表示前 k 個時間步中間價格的平均值; ma 表示未來 k 個時間步中間價格的平均值。
 - $mb(10)$ 且 $k=3$ 代表在第7, 8, 9時間步的平均價格; $ma(10)$ 且 $k=3$ 代表第11, 12, 13時間步的平均價格, 就能比較趨勢

文章內的虛擬變數

It. 透過對過去和未來平均中間價格的比較，預測未來價格將如何改變

$$l_t = \begin{cases} 1, & \text{if } m_b(t) > m_a(t) \cdot (1 + \alpha) \\ -1, & \text{if } m_b(t) < m_a(t) \cdot (1 - \alpha) \\ 0, & \text{otherwise} \end{cases}$$

α : 最小價格變化量，只有變化量超過某個閾值，才會認為市場存在趨勢變化
避免對噪音過度反應

實驗結果

TABLE I: Experimental results for different prediction horizons k

Model	Recall	Precision	F1	Cohen's κ
Prediction Horizon $k = 10$				
SVM	39.62%	44.92%	35.88%	0.068
MLP	47.81%	60.78%	48.27%	0.226
CNN	50.98%	65.54%	55.21%	0.35
Prediction Horizon $k = 20$				
SVM	45.08%	47.77%	43.20%	0.139
MLP	51.33%	65.20%	51.12%	0.255
CNN	54.79%	67.38%	59.17%	0.39
Prediction Horizon $k = 50$				
SVM	46.05 %	60.30%	49.42%	0.243
MLP	55.21%	67.14%	55.95%	0.324
CNN	55.58%	67.12%	59.44%	0.38

用來衡量模型預測結果與實際結果的一致性。

預測的It與真實數值是否一致

K= 1	完全一致
K= 0	結果與隨機猜測一樣
K= -1	一致性比隨機還低

K值越高，模型的預測更穩定準確

注意力機制(Attention)

對於不同輸入資料，分配不相同的權重，使模型能夠動態聚焦於最重要的部分，同時模型就會知道，想要產出甚麼結果，需要重點觀察那些數據。

引入注意力機制使得模型能夠靈活地識別哪些時間步的買賣單對於未來的價格趨勢更有影響力，並將這些關鍵特徵放大。

注意力機制(Attention)

時間步	價格變動幅度	引入attention前權重	引入attention後權重
1~10	10%	10%	30%
11~20	2%	10%	1%
21~30	3%	10%	3%
31~40	1%	10%	1%
41~50	0%	10%	1%
51~60	10%	10%	30%
61~70	1%	10%	1%
71~80	1%	10%	1%
81~90	2%	10%	1%
90~100	10%	10%	30%

加入注意力機制後，模型能夠更好地聚焦於那些真正影響未來價格的關鍵時段和買賣單數據，從而提升模型的F1-score, cohen's k等。

Reference

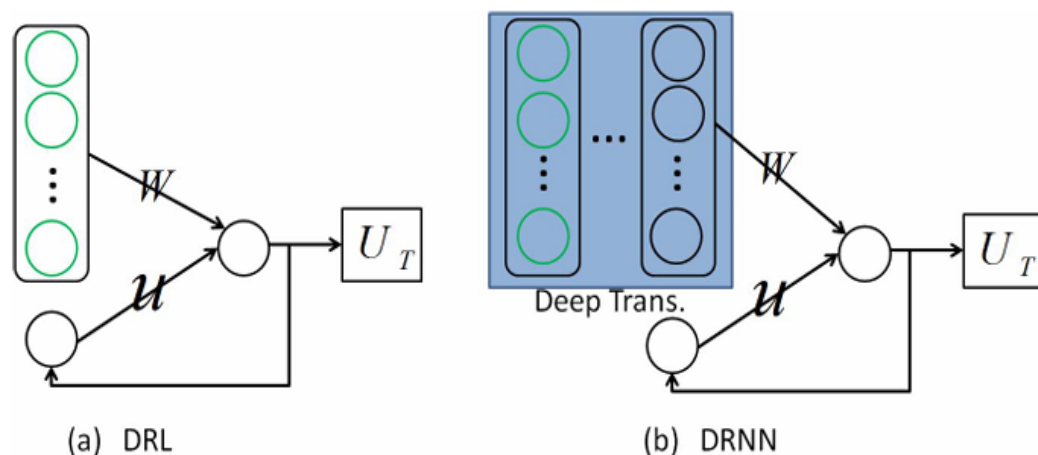
A. Tsantekidis, N. Passalis, A. Tefas, J. Kannianen, M. Gabbouj, and A. Iosifidis, "Forecasting stock prices from the limit order book using convolutional neural networks," *2017 IEEE 19th Conference on Business Informatics (CBI)*, 2017, pp. 7-12, doi: 10.1109/CBI.2017.23.

Deep Direct Reinforcement Learning for Financial Signal Representation and Trading

文章摘要

- 這篇文章以RNN作為基礎，進一步增加了Fuzzy layers(模糊層)以及task-aware BPTT([任務感知反向傳播)，最終產生FDDR模型
- 實驗目標：驗證FDDR模型的性能，測試模型在實際金融數據中的交易表現，並且對於不同情境下的表現，進行多模型比較。

FDDR架構演化



L (direct RL) 將過去的交易行為保存在記憶中

Fig. 1. Comparisons of DRL and the proposed DRNN for joint feature learning and DRT.

- DRL(左半)直接根據當前市場的特徵和上一時間步的交易策略，生成當前的交易策略
- 在原有基礎上，加入神經網路的概念，進行特徵擷取，達到最基本的RNN架構

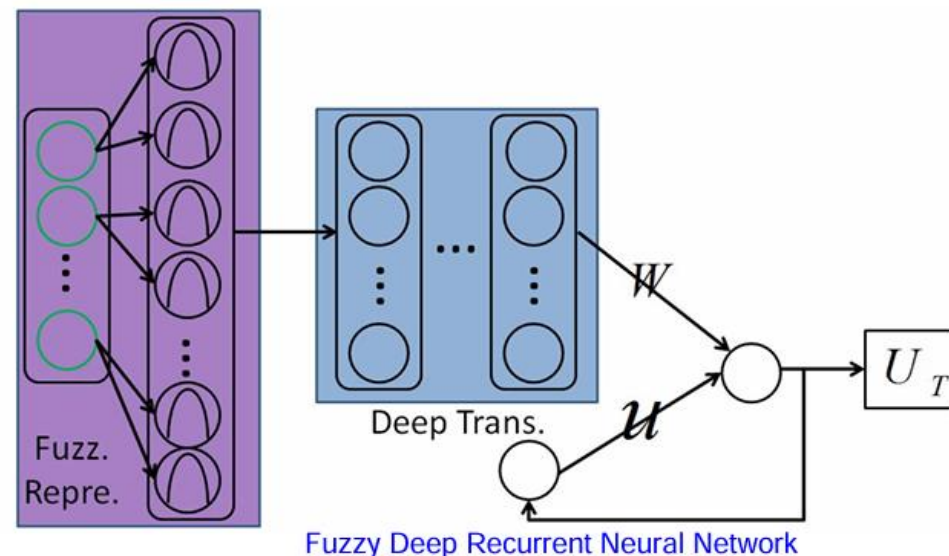


Fig. 2. Overview of fuzzy DRNNs for robust feature learning and self-taught trading.

- FDRNN則是再加入一個Fuzzy Representation Layer(模糊層)
- 通過模糊化的方式處理不確定性，將輸入數據(價格)轉換為模糊集合，最終輸出具有更高語義解釋的「上升」、「下降」、「穩定」三個模糊特徵

FDDR架構演化

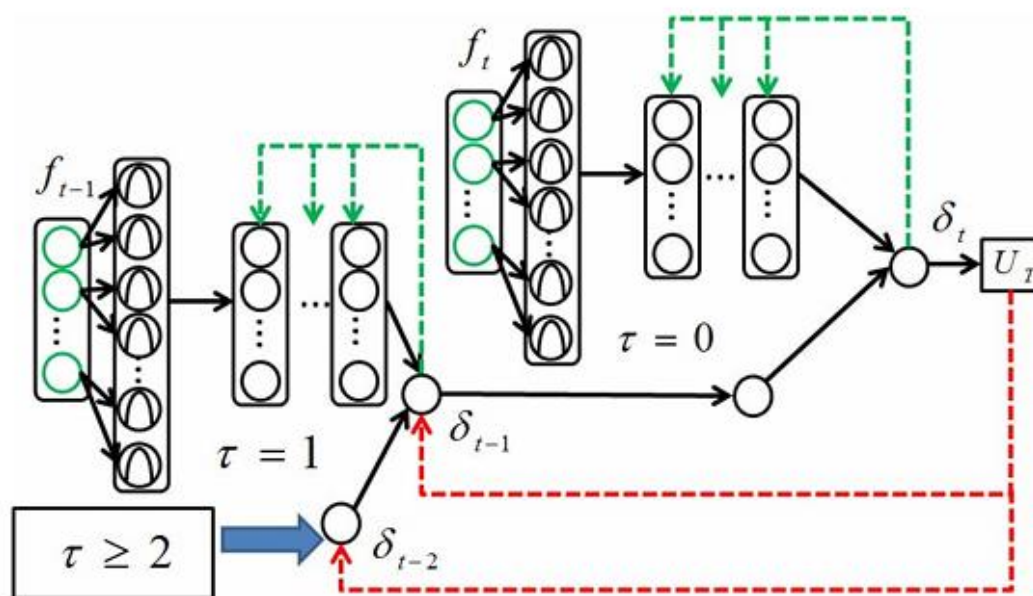


Fig. 3. Task-aware BPTT for RDNN fine tuning.

- 神經網路在反向傳播要優化參數時，常常會遇到梯度消失的問題，就使很靠近輸入層的參數更動得很慢，常見的方法是使用Relu激活函數或是直接改用其他神經網路(LSTM)解決。
- 本篇是使用Task-aware BPTT解決，建立兩個虛擬連結(virtual link)
 - **紅線**:每次在做決策都知道目標是什麼→達到最大化累積報酬
 - **綠線**:確保神經網路靠近輸入的部分也能學到有用的特徵→如何調整參數(權重、偏置、迴歸係數)

FDDR訓練過程(架構)

Algorithm 1 Training Process for the FRDNN

Input : Raw price ticks p_1, \dots, p_T received in an online manner; ρ, c_0 (learning rate).

Initialization: Initialize the parameters for the fuzzy layers (by fuzzy clustering), deep layers (auto-encoder) and reinforcement learning part sequentially.

```
1 repeat
2    $c = c + 1$ ;
3   Update learning rate  $\rho_c = \min(\rho, \rho \frac{c_0}{c})$  for this outer iteration;
4   for  $t = 1 \dots T$  do
5     Generate Raw feature  $\mathbf{f}_t$  vector from price ticks;
6     BPTT: Unfold the RNN at time  $t$  into  $\tau + 1$  stacks;
7     Task-aware Propagation: Add the virtual links from the output to each deep layer;
8     BP: Back-propagate the gradient through the unfolded network as in Fig. 3;
9     Calculated  $\nabla(U_t)_\Theta$  by averaging its gradient values on all the time stacks;
10    Parameter Updating:  $\Theta_t = \Theta_{t-1} - \rho_c \frac{\nabla(U_t)_\Theta}{\|\nabla(U_t)_\Theta\|}$ ;
11  end
12 until convergence;
```

Step 1: 輸入原始價格資料+參數初始化

Step 2: 原有價格序列特徵提取, 轉變成「價格上升」、「價格下降」、「價格穩定」三個特徵指標

Step 3: 輸入三個特徵指標作為神經網路的input

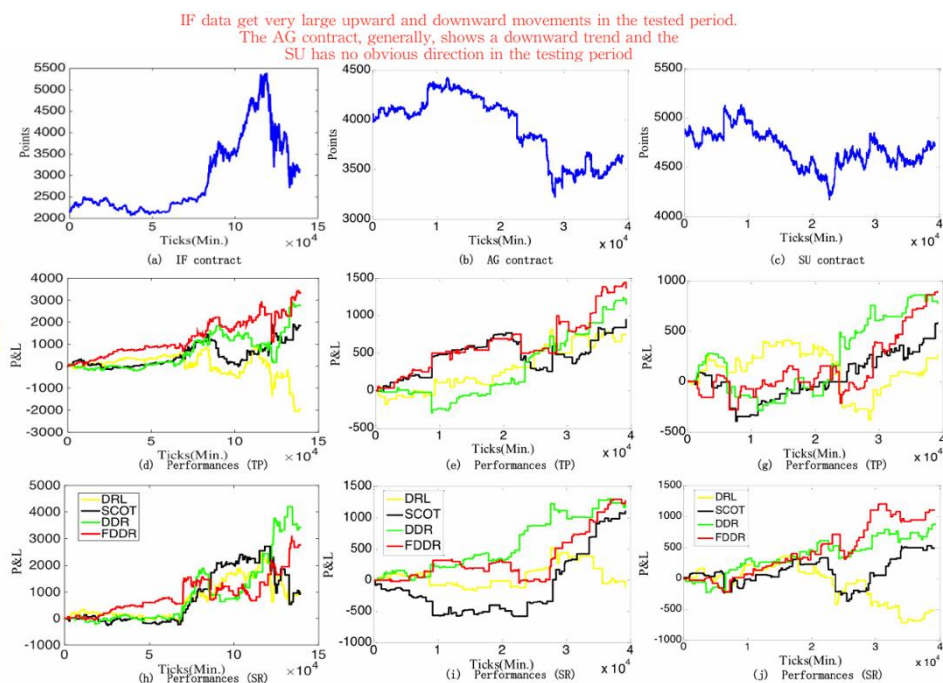
Step 4: 時間展開

Step 5: 建立虛擬連結

Step 6: 反向傳播更新參數

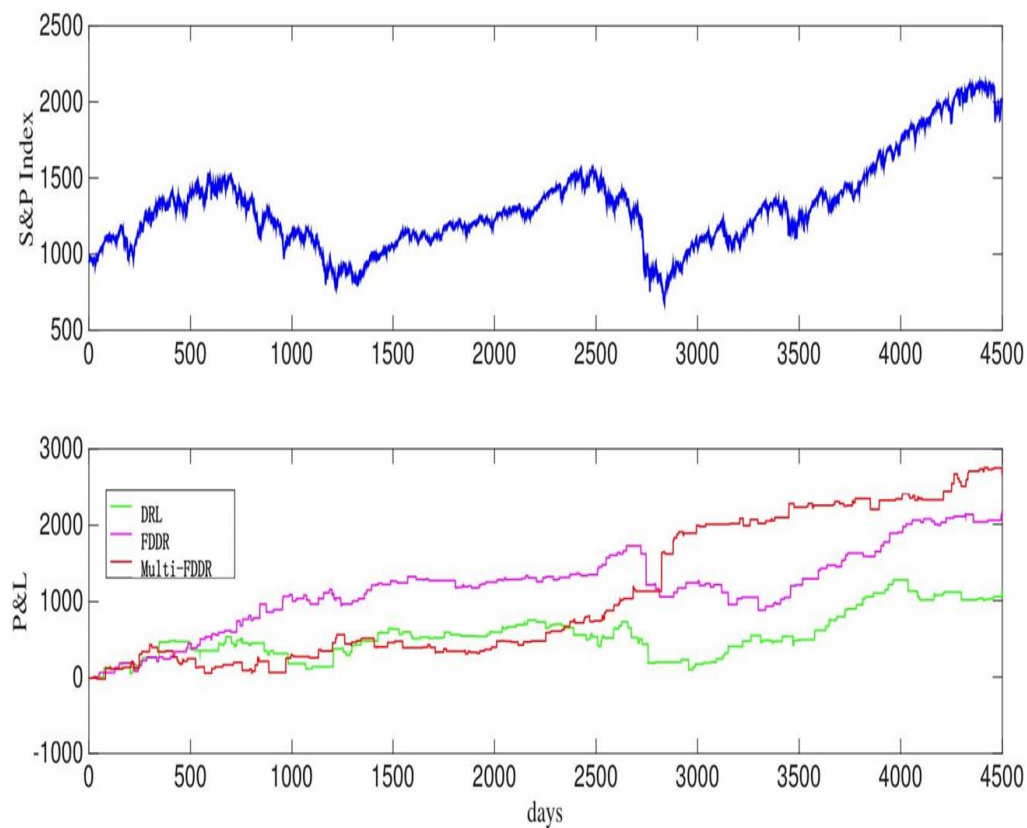
Step 7: 當模型參數不再顯著改變時 (收斂完畢), 則結束訓練過程, 產生交易策略「買入,賣出,維持」 $\{1,-1,0\}$

結果詮釋：商品市場比較



IF市場	<ul style="list-style-type: none"> IF忽然下跌時，所有模型損益也大幅下跌 →可能是因為此情況在訓練時，都沒出現過 為何DRL沒回到較好的水平？ →因為在訓練時的資料都是上升趨勢。但在IF市場中，出現了一個大幅下降的情況，但卻沒被訓練過
AU市場	<ul style="list-style-type: none"> 所有模型都有抓到下降趨勢，因此都產生正TP的結果 但DRL模型的SR數值都很接近0，可能是因為頻繁交易導致的
SU市場	<ul style="list-style-type: none"> 一開始有一個明顯下降趨勢，因此DRL的得分很高，但中間發生回調時，表現就不如神經網路模型 DRL的SR變很慘，主因是收到的噪音太多了，所以就會一直大賺大賠，風險變很高

結果詮釋：跨市場指數比較



- Yahoo Finance 獲取了 S&P 500 的日級歷史數據，時間範圍涵蓋了 1990 年 1 月至 2015 年 9 月，共超過 4500 個交易日
- 從圖中可以看出，FDDR(紫線)比起DRL(綠線)的表現好，在P&L的點數上多了約1000。