專案主題:
LSTM 預測股價報酬

學生: 陳冠維 國立清華大學 計財所 碩一

大綱

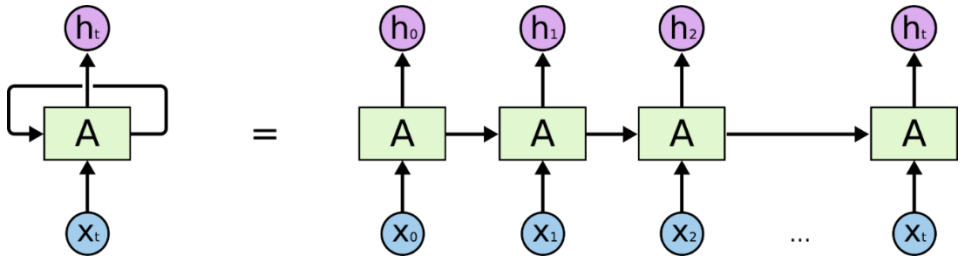
1

LSTM 模型介紹

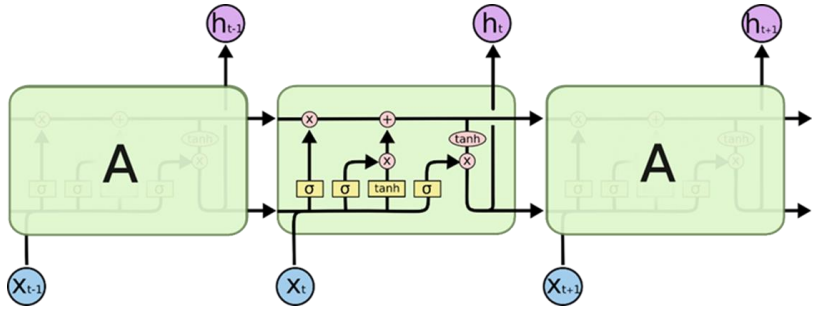
1. RNN
2. LSTM
3. RNN、LSTM之比較

RNN

- RNN 處理資料的特性: RNN 不會一次將整個序列X 讀入，而是像迴圈一般一次讀入一個小x，並在每個時間點更新細胞A，並輸出結果小h。
- 優點: 處理序列資料，例如: 文本資料

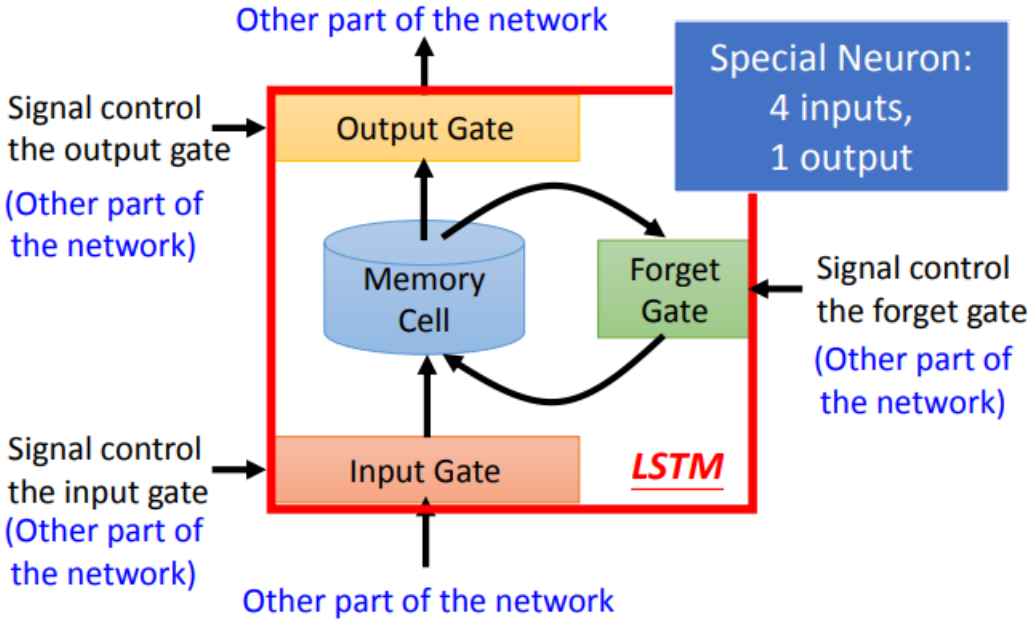


- RNN 的記憶狀態: RNN 在每次讀入任何新的數據之前，細胞 A 中的記憶狀態都會被初始化為0，這導致 RNN 沒辦法很好地「記住」前面處理過的序列元素，造成 RNN 在處理後來的元素時，就已經把前面重要的資訊給忘記了。
- 缺點: RNN無法記住前面處理過的元素，LSTM可以！



LSTM

- Forget Gate: 決定細胞是否要遺忘目前的記憶狀態
- Input Gate: 決定目前輸入有沒有重要到值得處理
- Output Gate: 決定更新後的記憶狀態有多少要輸出
- 優點: 處理時間序列資料



RNN、LSTM之比較

傳統RNN

- 語言閱讀的思路，接近於人。
- 在實作上，傳統的 RNN 很難捕捉到長期的記憶，數學上所產生的梯度消失的問題造成長時間的記憶會被短時間的記憶所隱藏。

LSTM

- LSTM 透過設計較佳的激勵函數，在神經單元(細胞A)中加入遺忘、輸入(更新)、輸出閥門，去決定哪些細胞狀態應該被遺忘、哪些新的狀態應該被加入、根據當前的狀態和現在的輸入，輸出應該要是什麼。
- 增加長時間的記憶表現，不只考慮最近的輸入，可以將任意時間點的狀態拿來使用。
- 運用於個股價量的時間序列資料。

2

Lasso 重要因子

1. 基本價量因子介紹
2. 因子特徵擷取
3. Lasso 尋找重要因子

基本價量因子介紹

Time-insensitive factor

- Bid-Ask spread
- Mid Price
- Price Difference
- Mean Price & Mean Volume
- Accumulative Bid-Ask spread
- Accumulative Bid-Ask Quantity spread

Time-sensitive factor

- 1-tick 價格取差分
- 5-ticks 價格取差分
- 10-ticks 價格取差分
- 1-tick 量取差分
- 5-ticks 量取差分
- 10-ticks 量取差分

因子特徵擷取

- 用意: 雖然一般使用深度學習時，不會特別做 feature engineering 的工作，因為模型的 hidden layer 其實就幫我們做了 feature engineering。然而台股自 2020 年 3 月才開始逐筆搓合，自今能夠收集的資料量不足以將深度學習模型建的夠深。因此我在這次研究會先將前述的基本價量因子去做特徵工程，透過 Lasso 找尋對股價報酬較有解釋力的因子，再將這些因子放入 LSTM 去做學習。

因子特徵擷取

- 5檔買賣邊的力道 (Value)

  Ask Price * Ask Quantity

  Bid Price * Bid Quantity

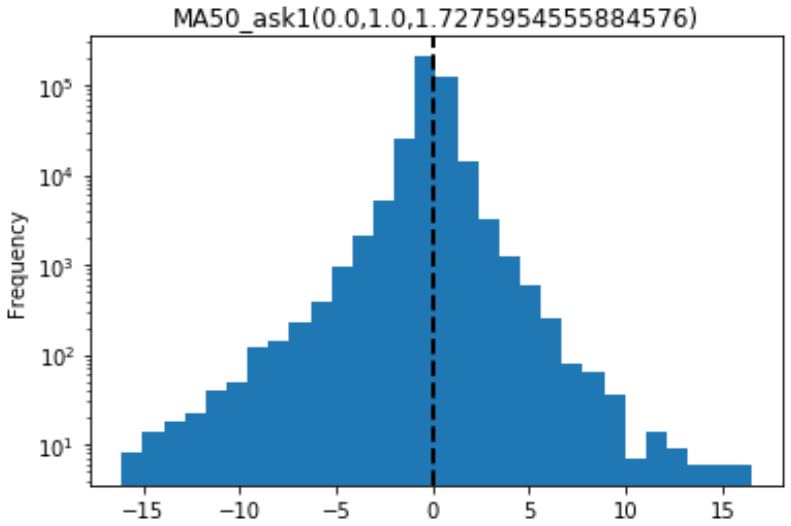
- Moving Average – Value

  Ask Value 50 ticks 的 MA – Ask Value

  Bid Value 50 ticks 的 MA – Bid Value

  Ask Value 100 ticks 的 MA – Ask Value

  Bid Value 100 ticks 的 MA – Bid Value



- 因子與報酬的主觀解釋

  MA_ask: 當 Ask Value 過去 50 個 ticks 的平均大於現在的Ask Value 時，代表市場處於多頭，與報酬呈現正相關。

  MA_bid: 當 Bid Value 過去 50 個 ticks 的平均大於現在的 Bid Value 時，代表現在市場處於空頭，與報酬呈現負相關。

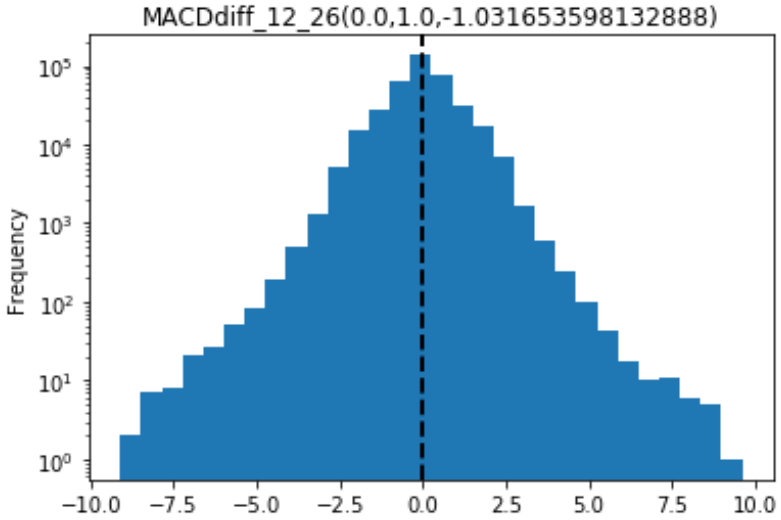
因子特徵擷取

- DIF

  短 EMA (12 個 ticks) – 長 EMA (26 個 ticks)

- MACD

  DIF – DIF 取指數移動平均 (9 個 ticks)



- 因子與報酬的主觀解釋

  DIF: DIF 代表短 EMA – 長EMA。當短EMA – 長EMA 大於0時，通常表示多頭，因此報酬為正，與報酬為正相關。

  MACD: MACD大部分只有在黃金交叉以及死亡交叉時有用。因為當黃金交叉或死亡交叉後一段時間，會有均值回歸的現象。因此MACD才會是與報酬呈現負相關。舉例來說，當快線由下往上穿越時，也就是MACD剛從負變成正的時候，通常價格會上漲，但是經過一段時間後會產生均值回歸，價格下跌。

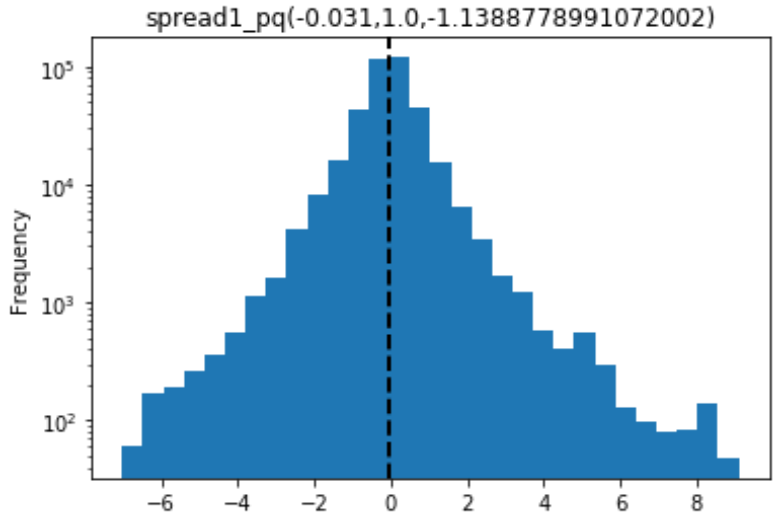
因子特徵擷取

- Spread Value

  Ask1 price – Bid1 price

  Ask2 price – Bid2 price

  ⋮



- 因子與報酬的主觀解釋

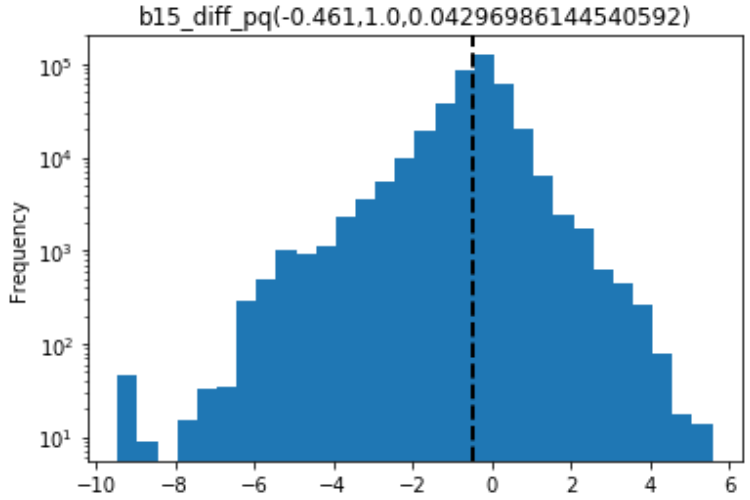
  Spread value: 我預期這群因子應該要與報酬呈現負相關，因為 Value 分別代表著買賣邊的力道。假如 Ask Value 大於 Bid Value，代表市場處於空頭，帶來的報酬為負。

因子特徵擷取

- Value Differences

  Ask1 price ~ Ask5 price 之間的差

  Bid1 price ~ Bid5 price 之間的差



- 因子與報酬的主觀解釋

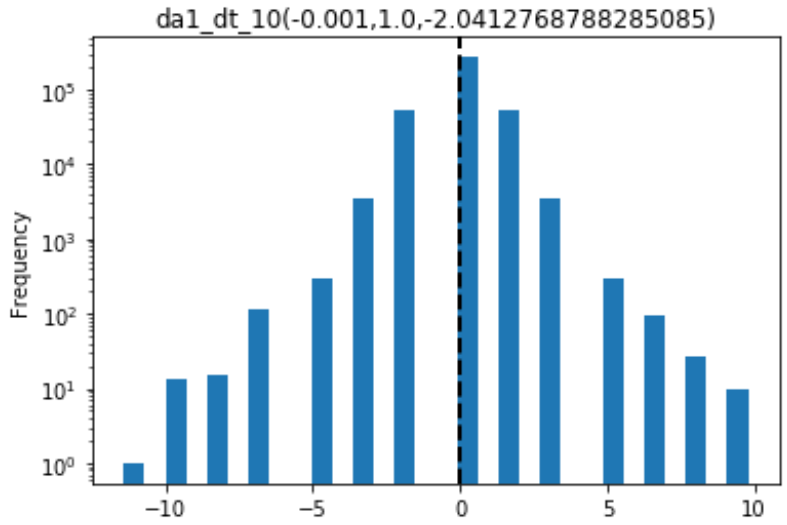
  Value Difference: 這群因子分別是Ask Value 之間的差以及Bid Value 之間的差。這群因子代表著5檔價量之間力道的差異。

  舉例來說，假如Ask1 的力道遠大於Ask5 的力道，那代表現在市場上可能看跌，大家搶著要賣出，因此我認為這個因子與報酬是負相關。

  反過來看，假如Bid1 的力道遠大於Bid5 的力道，那代表市場上現在想要逢低買進，未來趨勢看漲，因此我認為這個因子與報酬是正相關。

因子特徵擷取

- 5、10 個 ticks 價格取差分
  - 5、10 ticks Ask price 取差分
  - 5、10 ticks Bid price 取差分

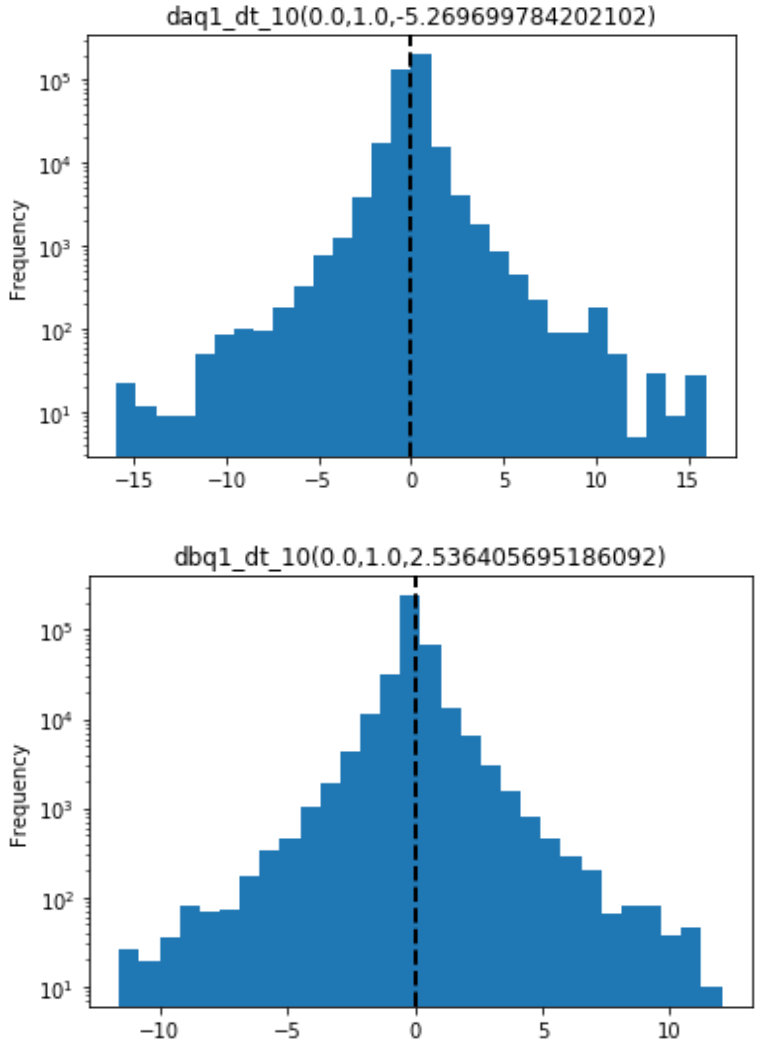


- 因子與報酬的主觀解釋
  - 5 & 10 ticks Ask 價取差分: 正相關
  - 5 & 10 ticks Bid 價取差分: 正相關

我預期這群價格去取差分的因子應該要與報酬呈現正相關，因為我認為價格取差分，無論是Bid 還是 Ask 帶來的應該會是動能。也就是說如果過去10 個ticks 價格被trade上來，那下10個ticks 之後的報酬應該也會往上。

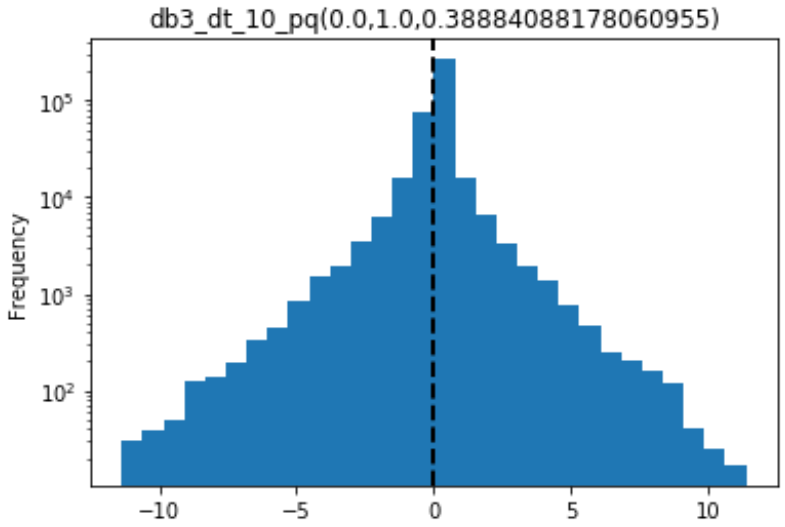
因子特徵擷取

- 5、10 個 ticks 量取差分
  - 5、10 ticks Ask price 量取差分
  - 5、10 ticks Bid price 量取差分
- 因子與報酬的主觀解釋
  - 5 & 10 ticks Ask 量取差分: 我預期Ask的量取差分應該要與報酬呈現負相關，因為 Ask 的量取差分的值愈高，市場應該是空頭，帶來的是負的報酬。
  - 5 & 10 ticks Bid 量取差分: Bid的量取差分應該要與報酬呈現正相關，因為Bid 的量取差分的值愈高，市場應該是多頭，帶來的是正的報酬。



因子特徵擷取

- 5、10 個 ticks Value 取差分
  - 5、10 ticks Ask Value 取差分
  - 5、10 ticks Bid Value 取差分



- 因子與報酬的主觀解釋
  - 5 & 10 ticks Ask Value 取差分: 我預期Ask Value取差分應該要與報酬呈現負相關，因為Ask Value代表賣邊的力道，Ask Value與前10個 ticks 取差分的值愈高，應該代表市場空頭，股價報酬會往下。
  - 5 & 10 ticks Bid Value 取差分: 我預期Bid Value 取差分應該要與報酬呈現正相關，因為Bid Value代表買邊的力道，Bid Value與前10個 ticks 取差分的值愈高，應該代表市場多頭，股價報酬會往上。

Lasso 尋找重要因子

factor	*Coefficient*
Ask1 Value 取 10 個 ticks 差分	4.19
Ask2 Price 取 10 個 ticks 差分	3.12
Bid1 取 5 個 ticks 量的差分	2.35
Bid2 取 10 個 ticks 量的差分	2.09
Bid1 取 10 個 ticks 量的差分	1.82

factor	*Coefficient*
Ask1 取 10 個 ticks 量的差分	- 4.66
Bid1 Value 取 5 個ticks 差分	- 2.64
Value1 的 spread	- 2.59
Ask1 Price 取 10 個 ticks 差分	- 2.40
Bid1 Price 取 10 個 ticks 差分	- 1.94

重要因子發現

- 模型選出的重要因子幾乎都是由 Ask1 或Bid1 所組成。這代表愈接近中間的價可以帶來愈多資訊，是符合預期的。

- 再者可以發現10個ticks 去取差分的效果比5個ticks 去取差分來的效果好。這也算是符合預期的，因為5個ticks 有時候可能擷取太短，發生價量都不太會動的情形，能夠提供的訊息量也較少。

3

LSTM
股價報酬預測

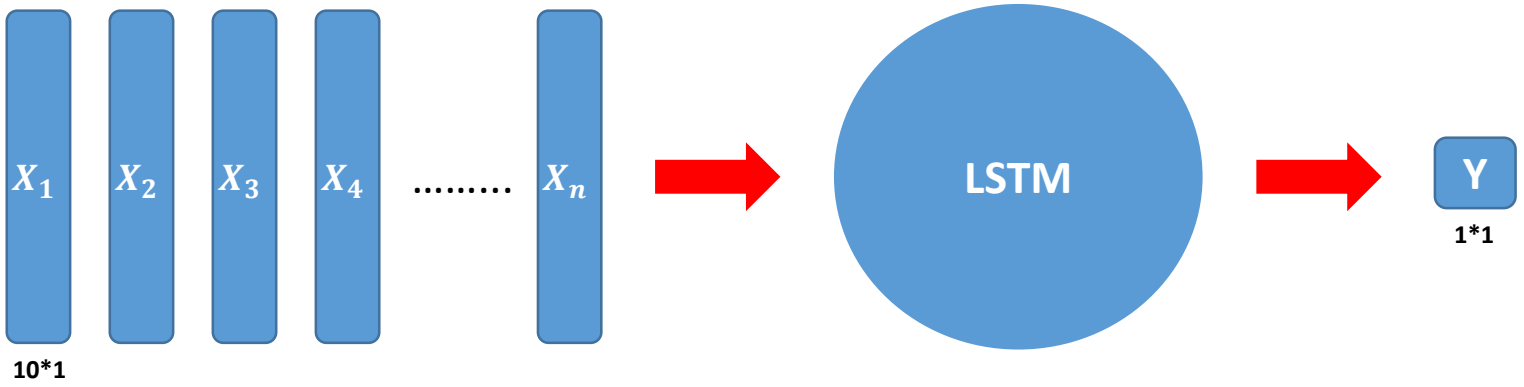
1. 資料前處理
2. 樣本內預測結果
3. 樣本內、外回測績效成果

資料前處理: Resampling

- 用意: 由於股價報酬的數值普遍貼近於 0，因此如果在未處理的情形之下，直接丟給模型做學習的話，模型會傾向將報酬都預測為0附近的值，以使得 loss function 最小。預測出來的報酬都貼近於 0 的情形，會造成多數股價報酬的低估。而我們知道，金融市場裡特殊事件往往含有重要的訊息，因此我們不希望預測出來的股價報酬出現此現象。
- 方法: 我透過 Resampling 的方法，對股價報酬再抽樣。使得訓練樣本股價報酬的分配不再全部聚集在 0 的上下，透過此方法，可以使模型學習到更多的價量 pattern。
  - EX: 假設原本是固定預測下 10 個 ticks 的股價報酬，可以將其改為預測下 N 個報酬有變動的時間，就不會常常發生 10 個 ticks 價格都沒有變的情形。

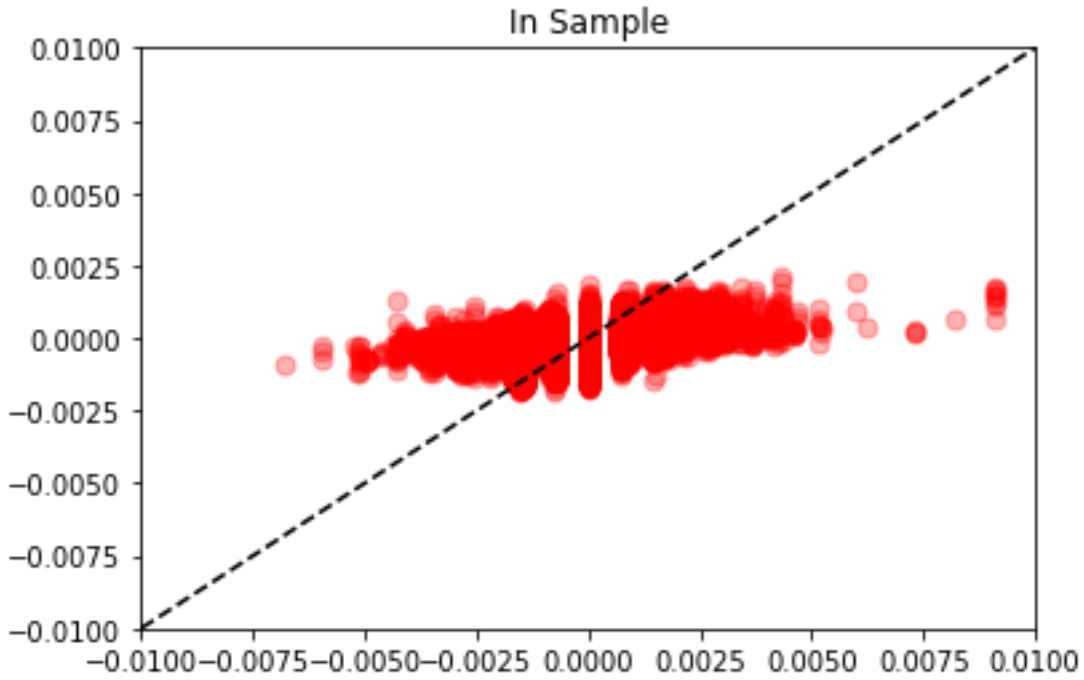
資料前處理: 資料型態

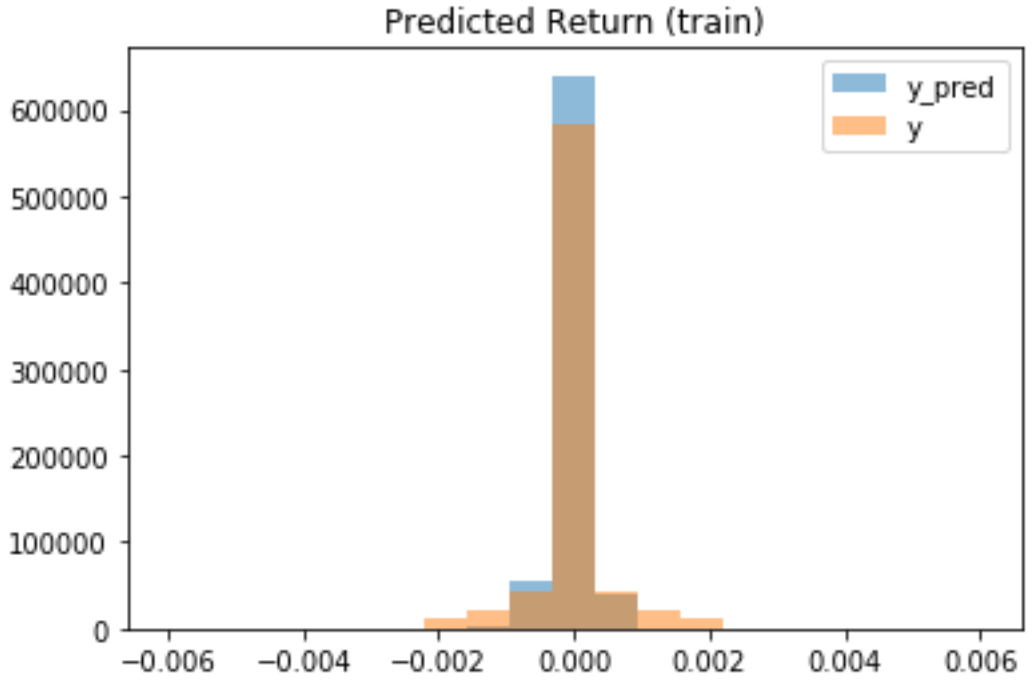
- 標的: 這次選擇的是價格波動較為劇烈的國巨 (2327) 作為標的，主要原因是高頻交易做當沖較傾向於關注價格波動頻繁、交易量大的股票
- 訓練集: 2020/03/25 ~ 2020/04/23　　測試集: 2020/04/24 ~ 2020/05/05
- Response (Y): 經過 Resampling 後的股價報酬
- Factor (X): 前述經過 Lasso 挑選出來的重要因子
- Time Periods: 運用**前 10 個 ticks 的因子 (vector sequence)**，去預測未來的股價報酬 **(a scalar)**



樣本內預測結果

Train R-squared: 0.15



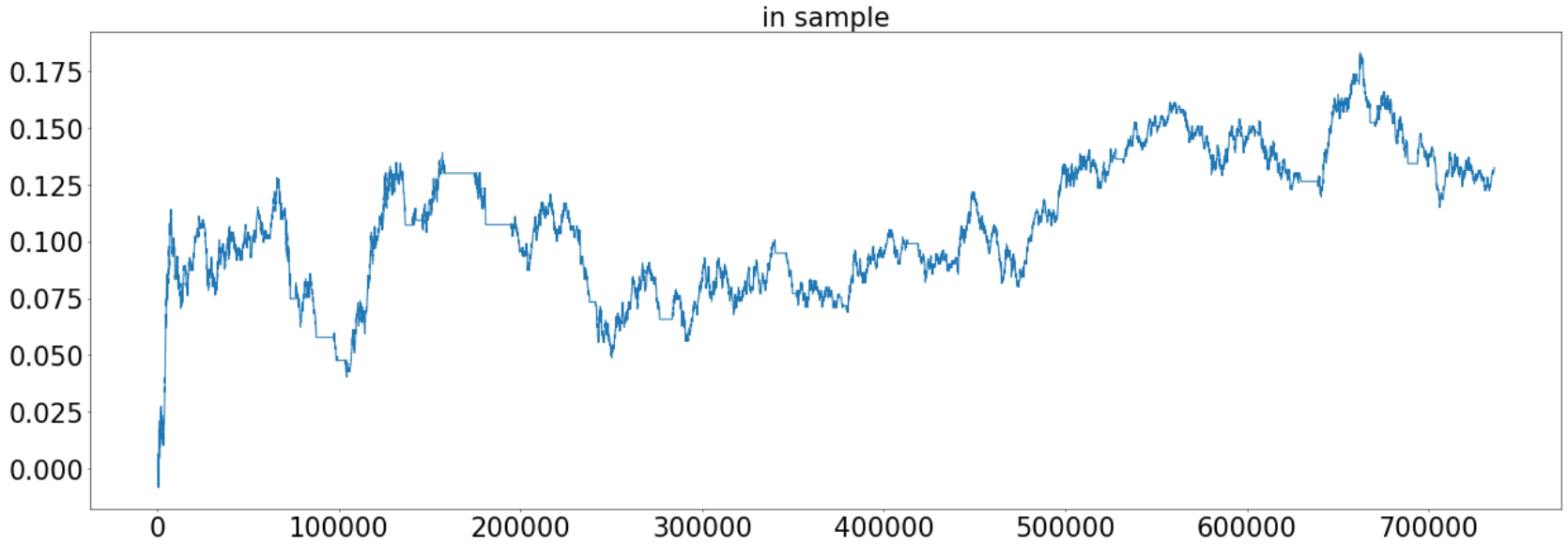


樣本內回測績效

交易策略: 以預測的未來股價報酬來判斷做多 (± 0.1% ~ ±0.3%) ，手上的部位保持在 [1, 0, -1]

樣本內回測績效

- Return: 13.2%
- Vol: 0.031
- Sharp Ratio: 4.14
- Transactions: 106
- Odds Ratio: 0.53
- Max Drawdown: 0.074

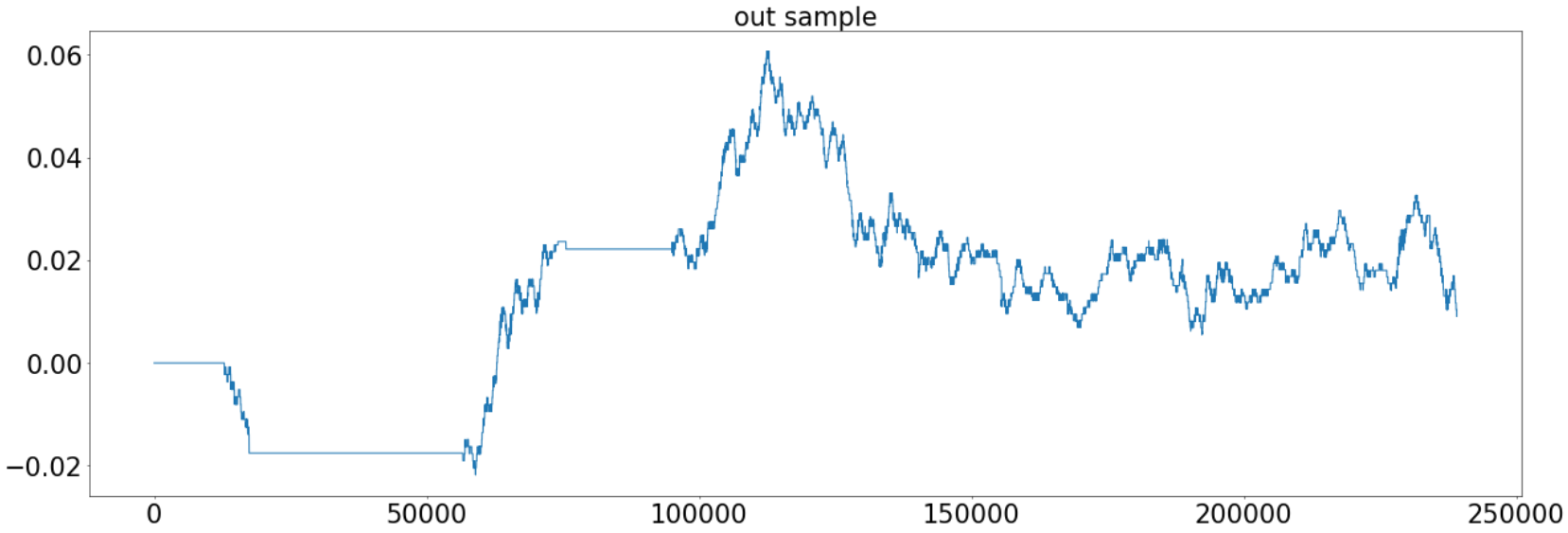


樣本外回測績效

交易策略: 以預測的未來股價報酬來判斷做多 (± 0.1% ~ ±0.3%) ，手上的部位保持在 [1, 0, -1]

樣本外回測績效

- Return: 0.77%
- Vol: 0.013
- Sharp Ratio: 0.183
- Transactions: 11
- Odds Ratio: 0.36
- Max Drawdown: 0.022



4

結論

1. 盤整盤與順勢盤的績效
2. 資料量的增加
3. 未來展望

結論

1. 可以從前面的績效看出，LSTM 運用於交易策略在順勢盤可以穩定獲利。然而遇到盤整盤時，容易因為頻繁的進出場而虧掉前面的獲利。不過這也是因為我的策略設計的較為簡單的原因，假如能設計策略於盤整盤時不要頻繁的進出場，相信 LSTM 也可以在盤整時守住虧損。
2. 台股自 2020 年 3 月才開始逐筆搓合，累積自目前訓練的資料量仍有所不足，相信日後累積更多資料後，LSTM 能夠建的更深，使預測股價報酬的表現更好。
3. LSTM 有個較大的問題是無法平行運算，因此即使用 GPU 加速了也慢於其他模型，在高頻交易中較為劣勢。最近有一些 CNN based 的深度學習模型，也開始被設計出來應用於時間序列資料。CNN 最為優勢的就是可以平行運算，GPU 加速後產生預測結果的速度遠高於 LSTM。因此在日後的研究裡，我也會嘗試運用如 Temporal Convolutional Network、Dilated CNN 等模型來預測股價報酬。