Automated trading system for Taiwan Index Futures using LSTM networks

利用長短期記憶模型實作自動化交易系統

組別：B255 指導教授：馬席彬 組員姓名：劉振昇、鍾岷翰

摘要

本專題主要為設計深度學習模型與執行策略，在台指期的交易中可以獲取利潤。

首先，我們將台指期逐筆交易封包資料取出，並將此資料整理及運算成所需參數， 例如 Orderbook, Order flow, MACD,EMA,等等參數。將這些參數放入 LSTM 模型訓練，而得到的模型在 testing data 內配合設計的策略實際操作，並依據情形制定策略來獲利使得成效彰顯。

1. Introduction

自動化交易系統是一種利用演算法讓電腦根據事前設定好的策略自動在市場上進行買賣行為的交易系統，相較於人工交易，除了有更加快速以及自動化的優點外，還能夠應用深度學習模型讓系統自行分析當前市場情況決定買賣的時機點，進一步提高交易獲利。

本專題目標為利用深度學習模型的趨勢預測設計一套適用於台灣期貨市場的自動化交易系統，整體流程從解碼交易所傳輸的行情資訊封包至設計深度學習模型，最後到交易策略的制定，涉及範圍包括 OSI 網路模型的認識、深度學習網路的架構理解，以及期貨交易方面的策略分析。

專題結果為訓練出了能有效預測市場趨勢的 LSTM 模型，並能套用我們設計的交易策略在台指期商品上獲得有效的利潤。

2. Research Methodology

2-1. System Design

本專題所設計的自動化交易系統流程首先必須要解碼台灣期貨交易所(TAIFEX)傳輸的交易行情資訊，我們可以根據交易所發布的逐筆行情資訊傳輸手冊[1]裡制定的資訊格式來解密封包，得到我們所需的資料，接下來將這些交易資料做預處理得到模型所需的 features，並將其導入 LSTM 模型做市場價格的趨勢預測，然後將預測的趨勢導入設計的交易策略進行決策判斷，最後會得到買、賣或是不採取行動的行為決策，下圖為系統整體的架構圖，以整個系統來

看，交易所傳輸的交易資訊是整體系統的輸入，而經過一連串的處理後所產生相對應的交易決策，則是整體系統的輸出。

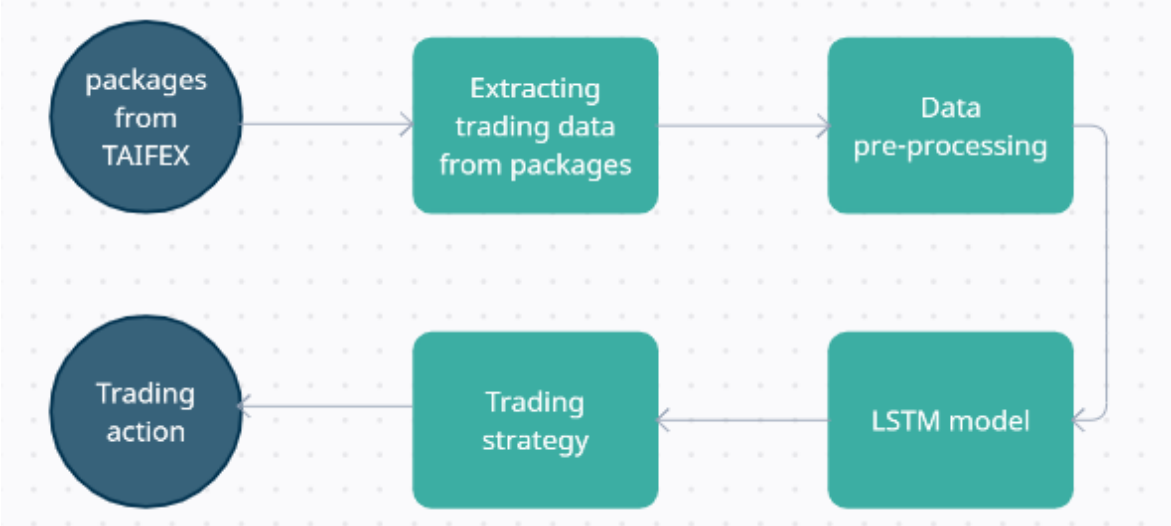

Fig. 1 System Structure

以上為整體架構的大致介紹，下面會詳細講述各個區塊的原理以及實作流程。

2-2. Extracting trading data from packages

這個區塊的任務是將我們所需要的特定商品資訊從封包裡解密出來，並做出該商品完整的 orderbook 供 Data pre-processing 做使用，根據 OSI 的網路框架我們可以知道網路包含了實體層、資料鏈結層、網路層、傳輸層、應用層等，通過讀取各層的 header 和 padding 便能夠逐層解開封包，而對於交易所傳輸的交易行情資訊，我們可以根據 PCAP 文件格式，以及 Ethernet II、IPv4、TCP 等通訊協定去擷取出應用層的資料，然後再根據逐筆行情資訊傳輸手冊的格式解碼得到商品的各項資訊。

要做出商品五檔價量的 orderbook，我們需要委託簿揭示訊息(I081)以及委託簿快照訊息(I083)兩種訊息，委託簿揭示訊息能夠提供其中一檔的價量變化，而委託簿快照訊息則是提供全部五檔的更新資訊，兩種訊息合併起來便可以為 orderbook 提供完整的更新訊息，而解出封包後完整的 orderbook 包含了五個檔位的買價、賣價、買量、賣量，我們可以針對這些資料進行一些運算處理來提供給 LSTM model 當作 features 來使用。

2-2. Data pre-processing

在這個區塊我們需要把拿到的 raw data 經過處理轉換成深度學習模型能夠接收的 features，在一開始我們先對 orderbook 做除頻，這是因為期貨交易的 orderbook 更新最快可以到微秒等級，在兩次更新中間間隔這麼短的情況下很容易會出現兩筆資料差異不大的狀況，原因是交易掛單的頻率並沒有高到一次更新會讓整體 orderbook 狀態產生大變化的程度，如果一直給模型重複性太高的

資料，會對模型的訓練成效產生不好的影響，也就是說，對模型而言會是 noise 很高的資料，因此在本專題中我們將 orderbook 每更新 10 次的結果視為一筆資料來進行處理。

我們之後透過 talib 套件，利用除頻過的 orderbook 的價量來製作技術性指標，包括 EMA、MACD、BBands、以及 order flow[2]，並且依據每一筆資料之間的第一檔價格漲幅變化來 label 每個 sample 的漲跌，若當前價格高於上一個時間點則標示為漲，低於上一個時間點則標示為跌，不變則為平，再使用 one hot encoding 的方式將漲跌平標示成三個 label，而下面將詳述各個指標的資訊。

2-2-1. EMA

EMA(指數移動平均)是 MA(移動平均)的延伸版本，EMA 透過對當前時間越接近的價格給予更高的權重來強化 MA 不足的地方，EMA 比較能反映出近期價格的變化，下列為 EMA 的計算公式。

$\text{EMA (N)} = \frac{2X + (N-1)\times \text{EMA}(N-1)}{N+1}$ x = current price, N = time period

我們選擇 time period 為 20 的 EMA 作為模型的 features。

2-2-2. MACD

MACD 指標分為三個部分，分別為 MACD(快線)、Signal line(慢線)、MACD histogram(柱狀圖)，其中快線是將兩條 EMA 線相減所得到的結果，慢線則是對快線再去做 EMA 後得出的結果，柱狀圖則是快線與慢線相減的值，在本次專題中我們選擇快線作為我們模型的 features [3]，下列為 MACD 的計算公式。

MACD = EMA(N)－EMA(M), N=fast period, M = slow period

我們選擇 fast period 為 12，slow period 為 26 的 MACD 作為 features。

2-2-3. BBands

BBands(布林通道)分成三個部分，分別為 upper band(上線)、middle band(中線)、lower band(下線)，中線為一段期間的 MA，upper band 是中線加上兩倍的價格標準差，lower band 則是中線減掉兩倍的價格標準差，用這三條線可以表示出市場價格的波動程度，所以在本次專題中我們選擇這三條線作為模型的 features，下列為 BBands 的計算公式。

middle band = MA(N), N= time period

upper band = middle band + $2 \times \sigma$, σ = variance for prices of N period

lower band = middle band - $2 \times \sigma$, σ = variance for prices of N period

2-2-4. Order flow

order flow 分為兩個部分，分別為 ask order flow 和 bid order flow，order

flow 的指標會根據價格的漲跌對 orderbook 的口數進行不同的運算，可以作為商品流動性的一個有效指標，下列為 order flow 的公式。

i = 檔位, t = 當前時間。

$a_{t,i}, b_{t,i}$ 為在時間 t 時第 i 檔的買入價和賣出價。

$v^a_{t,i}, v^b_{t,i}$ 為在時間 t 時第 i 檔的買入口數和賣出口數。

$$aOF_{t,i}\begin{cases} v^a_{t,i}, & if\ a_{t,i} > a_{t-1,i} \\ v^a_{t,i} - v^a_{t-1,i}, & if\ a_{t,i} = a_{t-1,i} \\ -v^a_{t,i}, & if\ a_{t,i} < a_{t-1,i} \end{cases}$$

$$bOF_{t,i}\begin{cases} v^b_{t,i}, & if\ b_{t,i} > b_{t-1,i} \\ v^b_{t,i} - v^b_{t-1,i}, & if\ b_{t,i} = b_{t-1,i} \\ -v^b_{t,i}, & if\ b_{t,i} < b_{t-1,i} \end{cases}$$

2-3. LSTM model

本專題中我們使用 pytorch[4]來實作 LSTM model，model 最後會輸出未來漲跌趨勢的預測，判別漲、跌、平三個 label，我們所建構的模型是多對多預測的 LSTM 分類模型，利用目前一段時間的資料去預測未來一段時間的 label，而並非做 regression 來猜測未來的期貨商品價格。

2-3-1. Model specification

本模型使用 Adam 作為 optimizer，而根據 pytorch 中 LSTM 的官方文檔，torch.nn.LSTM 的輸入為一個三維的 tensor，形狀為(N, L,H_{in})，輸出也是一個三維 tensor，形狀為(N, L, D ∗ H_{out})，其中 N 為 batch size，L 為 sequence length，H_{in}為 input size，H_{out}為 output size，D 為 bidirectional 的設定值，若模型為雙向 LSTM，則 D = 2，若為單向 LSTM，則 D = 1 則 LSTM 模型本身可設置的參數則有 input size、hidden size、number of layers、drop out、bidirectional，在本專題中使用的參數設定如下面表格所示。

Table 1 LSTM model parameters setting

Parameters	Value
batch size	512
input size	28
sequence length	39
hidden size	64
number of layers	2
drop out	0.5
bidirectional	1
initial learning rate	0.001
output size	3

由於 LSTM 比較容易 overfitting 的原因，我們將此模型的 drop out 設為 0.5，並採用 early stopping 的方式防止模型 overfit 到訓練資料上，最後在使用 cross-validation 去評估模型的表現好壞，在 learning rate 的部分，我們採用 scheduler 讓模型在訓練過程中能夠調節本身的學習率，使模型容易達到收斂的狀態。

在 torch.nn.LSTM 的格式下，模型輸入以及輸出的 sequence length 為一致的大小，在實作上要做出 20 對 20 的預測模型，我們需要對輸入資料做 padding 的動作，在本專題中 padding 是採用 zero padding 的方式來完成。

2-4. Trading Strategy

這個區塊的任務是用 LSTM model 所產生的趨勢預測配合買賣策略在市場上即時做出交易決策[5]，我們所使用的方法是根據模型的漲跌預測決定買賣的時機，並用市場上當前的價格情勢對每筆交易行為做風險管理，流程圖如下。

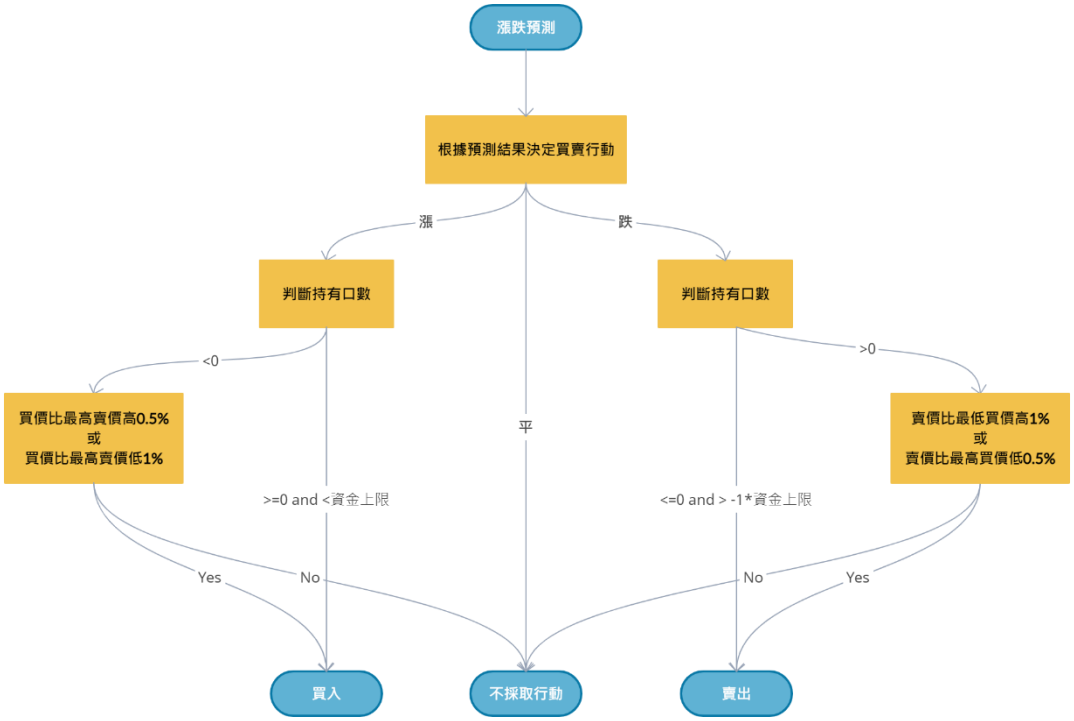

Fig. 2 trading strategy flow chart

每次的決策會先從機器學習的預測結果開始判斷買賣行動，例如預測上漲就買或做多，下跌就賣或做空。之後再拿當前市場上的價格與持有商品之價格來對比去做進一步的判斷，決定口數多寡，以及持倉後的買賣時機判斷。在設定賺賠比為 1:0.5 後也可以有效做好風險管控。此外可以由此讓交易頻率可以穩定，避免手續費過高而吃掉獲利。搭配模型的預測結果以及交易策略來將利益最大化。

3. Experiment Results

3-1. LSTM model performance

以5月30日的 TXFF2商品資訊做為訓練資料，5月31日的 TXFF2商品資訊做為測試資料，最終模型的測試結果如 Fig. 3所示，總體的預測準確率為67%，而由於經過 Data pre-processing 和 label 後，總體趨勢為平的樣本數無論在訓練資料或是測試資料都會很少，所以模型不太會做出趨勢為平的預測，從 Fig. 3可以看到模型並沒有預測樣本為趨勢平的 class，但是這個現象並不會對後續系統在交易上的效益產生太大的影響，因為根據我們的交易策略，虧損主要是來自於將漲勢預測為下跌與將跌勢預測為上漲的情形，所以我們並沒有針對 imbalance 的狀況去對模型進行調整。

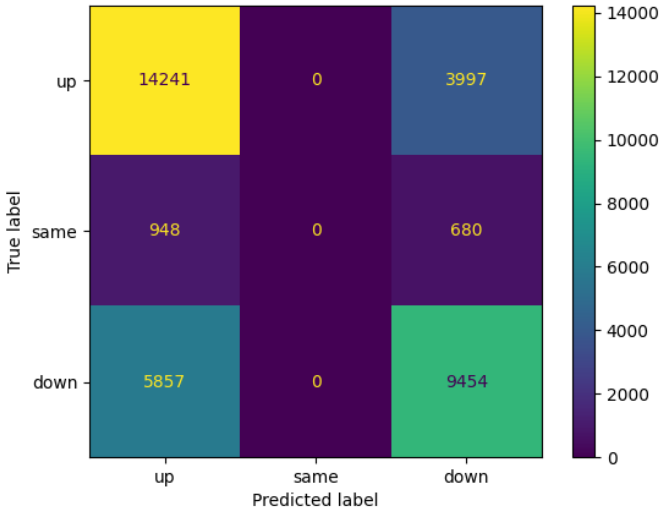

Fig. 3 confusion matrix

3-2. Trading profit testing

我們利用交易所過往傳輸的市場交易資訊對系統進行交易的模擬並測試其獲利率，此部分的測試以持倉資本700萬元作為上限來進行。

Fig. 4的 train_data1和 test_data 分別為5月30日和5月31日的市場價錢趨勢，而 Fig. 5則為系統在5月31日除去剛開盤與收盤的整日累積利潤走勢圖，profit 的單位為點，在此商品相當於新台幣200元。

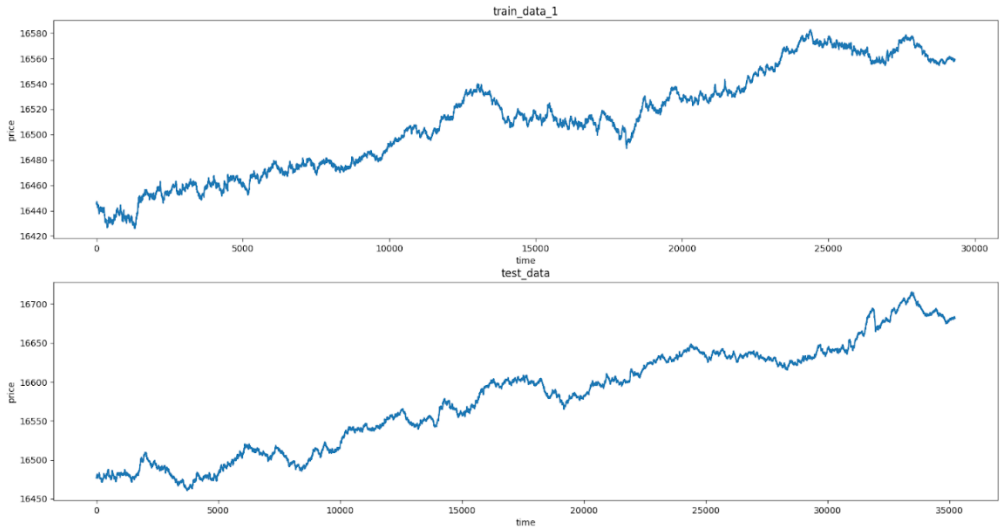

Fig. 4 market trend

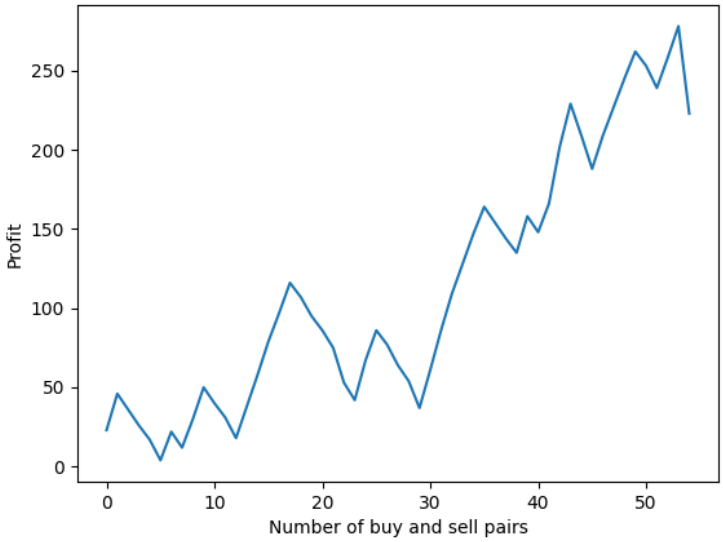

Fig. 5 profit run chart

在此次測試中，一天下來的利潤為223點，折合新台幣44600元，單日獲利率為0.637%。

從實驗結果來看，系統在訓練資料和測試資料的市場行情差異不過大的情況下能夠帶來可觀的收益，但是在兩者趨勢差別很大時，收益則會明顯減少，我們認為此種狀況可能是模型有些 overfitting 的現象，要改善這種狀況可以從調整 LSTM 模型的 hidden size 和調整 drop out 這兩個方面開始，或是經由改變 LSTM padding 的方式來提升模型的表現能力。

4. Conclusion

從本專題中我們瞭解透過 LSTM 模型來進行預測與交易的可能性。LSTM 模型在使用參數 Orderbook, Order flow, MACD,EMA 預測效果良好，使用多對

多的 label 並取用預測結果的後半資料，搭配策略可以獲得不錯的獲利效果。但有些模型的設定上仍須修正使得結果可以因應更多的情形，或者需要考量更多情況制定策略，來讓獲利更好。

5. 心得感想

我們在這次專題研究中接觸到了計算機網路概論、深度學習、期貨交易的實作內容，一開始因為不太熟悉這些領域的知識，所以起步的比較慢，但是隨著專題的進行，我們逐漸從閱讀資料以及實作問題上開始找到研究的方向。感謝教授以及同時修課的每位同學，當遇到困難時都能夠給予指導與討論，讓我們找到解決問題的方向。

6. Reference

[1] Tick-by-tick Market Information Transmission Operation Manual (ver.1.2.0), Taiwan Futures Exchange, 2020.

[2] PETTER N. KOLM, JEREMY TURIEL AND NICHOLAS WESTRAY, Nicholas, "Deep Order Flow Imbalance: Extracting Alpha at Multiple Horizons from the Limit Order Book", August 5, 2021

[3] Deep Reinforcement Learning for Trading, Z. Zhang, S. Zohren and S. Roberts The Journal of Financial Data Science 2020 Vol. 2 Issue 2 Pages 25-40

[4] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, *Pytorch*, https://pytorch.org, Meta AI, October, 2016

[5] High-frequency trading: Definition, implications, and controversies, K. Z. Zaharudin, M. R. Young and W. H. Hsu, Journal of Economic Surveys 2022 Vol. 36 Issue 1 Pages 75-107