

國立清華大學 電機工程學系

實作專題研究成果報告

Deep Learning for Predicting Mid-Price Movement
in High Frequency Trading

深度學習之應用於高頻交易中間價走勢預測

專題領域：系統領域

組 別：B253

指導教授：翁詠祿 教授

組員姓名：陳昭維、林禹陞

研究期間：111 年 2 月 1 日 至 111 年 11 月 25 日止，共10個月

Abstract

The progress of statistical science and transparency of real-world data, along with the improvement of hardware and computing efficiency. An increase of application of statistical inference on real-world problem and professional field research is observed.

Among all the field, financial industry is a field and sector that has a high usage of statistical inference. Since the scale and speed of every financial product increases throughout the days, traditional method of quote-driven and human-based buy/sell strategy has been eliminated gradually from the financial world. Meanwhile, statistical-based flow of data collection, data preprocessing, feature extraction, algorithmic strategy decision and trading has become the main focus of current financial market and related field.

In this topic, we reference the paper “DeepLOB: Deep convolutional neural networks for limit order books”^[1] for our methodology foundation and neuron network model basis. We implement the model on the open-source data set - FI-2010 dataset ^[2] limit order book with preprocessing, then generates mid-price for market trend prediction. With add/buy information throughout a sequence of timestamps which leads to a future time market trend. This provides investor signals for market decision with advantage of making profits among all the other buyers/bidders.

On the basis of algorithmic trading theory and domain knowledge of financial market, we were able to build the deep learning model structure for stock prediction. We went through a series of mathematical calculation, construction, training and testing of neuron network to ensure the feasibility of the strategy. Eventually, we succeed our goal with exceeding the original author’s performance on prediction and latency.

摘要

隨著資訊科學以及資料的公開和透明化，以及同時間硬體的進步以及運算能力的提升。推論統計學(Statistical Inference) 應用於解決現實生活問題以及專業領域的研究的實現也逐年提升。

其中，金融交易為大量使用推論統計學的產業和領域，隨著交易規模和速度的提升，傳統使用人為因素去判斷是否買賣金融商品的策略已逐漸被淘汰。而使用收集交易時的資訊並將其量化與特徵選取後制定演算法與選定買賣的標籤去實現自動交易已成為金融單位的主要策略。

本篇專題將以參考論文“DeepLOB: Deep convolutional neural networks for limit order books”^[1]作為實作的原理參考和模型建置基礎，對開源數據 FI-2010 dataset ^[2]的限價委託簿(Limit Order Book)數據進行前處理，利用委買賣資料獲得中間價以預測股市的走向而產生決策。其中的原理在於能夠利用一定數量的時刻點資訊預先獲得買賣時刻點所該進行的決策，領先其餘投資人進行交易，在金融市場中獲利。

透過參考論文之實驗理論之啟蒙以及對於金融時序列資料之專業知識，我們奠基於之上建立了深度學習模型來做為預測股市的架構。我們透過許多數學計算和深度學習網路的嘗試和訓練，驗證了方法的可行性，最終搭建出屬於我們的模型並且超越原作者^[1]架構的預測能力以及延遲表現。

章節目錄

章節目錄

Abstract.....	i
摘要	ii
章節目錄	iii
圖目錄	iv
表目錄	iv
一、 前言	1
二、 研究動機	1
三、 實作前置介紹	2
3.1 使用資料集之介紹	2
3.2 結果驗證標準	2
四、 實作架構建置	3
4.1 資料前處理	3
4.2 資料特徵提取與標籤.....	3
4.3 深度學習模型總覽	5
4.4 深度學習模型詳述	6
五、 實作數據驗證和討論	9
5.1 實作前置設定	9
5.2 實作測試與結果	10
六、 結論	12
七、 參考文獻	13
八、 計畫管理與團隊合作方式.....	14

圖目錄

圖 1	本專題實作模型總覽	5
圖 2	DeepLOB ^[1] 結構	6
圖 3	ResNet ^[3] 結構	6
圖 4	ResNeXt ^[4] 結構	7
圖 5	SENet ^[6] 架構	7
圖 6	第一種測試方式 ^[1]	10
圖 7	第二種測試方式 ^[1]	10

表目錄

表 1	使用限價簿分析之五間公司 ^[2]	2
表 2	硬體實作環境	9
表 3	第一種測試方式與其 baseline 比較	11
表 4	第二種測試方式與其 baseline 比較	11
表 5	與參考論文之推論延遲比較	11

一、前言

限價委託簿(Limit Order Book)在現代的許多金融單位中所使用的交易機制，別於傳統市場單價的買賣點。委買方和委賣方可以選擇在不同等級的委買以及委賣的價格及數量進行搓合。而也由限價委託簿乘載著不同的價格及數量，其對應到市場的預測也成為了一個多維度的問題。

隨著金融單位的策略轉換以及硬體不斷地進步之下，高頻交易慢慢成為許多投顧單位和公司所採用的策略。高頻交易為一種高頻率、取樣數量點非常多的精準交易模式，透過電腦程式和硬體的運算能力，能夠避免人為、非理性因素的交易決策。高頻交易的預測模型的前置執行流程可分為三個階段:資料蒐集、資料處理、預測模型建置。

因為近年來機器學習之分支-深度學習對於學習時間序列資料和捕捉大量數據趨勢的優勢，因此深度學習網路也成為預測股市模型的核心。隨著運算晶片日益成熟，演算法能夠透過硬體的實現後將數學應用於現實問題之中，成為許多學者團隊研究的前進方向。

二、研究動機

在大學的課程中，我們學習基礎的知識和題材，但是我們往往只能從最表層的角度去了解應用面。作為電機系的學生，將來無論是朝向學界抑或是業界發展，我們會遇到的問題層面不會單純屬於電路設計或是軟體撰寫層面，我們會結合自身的專業於解決現實世界的問題。我們希望結合電機系的數學訓練在參考論文^[1]的基礎之下，發揮想法以及實作精神，嘗試超越作者以及其他 state-of-the-art 的模型，並且學習利用科學思維解決人文問題。

三、實作前置介紹

3.1 使用資料集之介紹

研究相關理論，開源的證券限價簿成為至關重要的存在本專題以及參考論文^[1]中共同對於 FI-2010 dataset^[2]進行模型建置的輸入資料。此資料集為首個於高頻交易中公開的限價簿時序列資料，其中數據由納斯達克北歐交易所五間公司^[2]的限價簿組成(表 1)。資料集的收錄時間為 2010 年 6 月 1 日至 2010 年 6 月 14 日。每日(平常日)的收錄時間為 10:30 至 18:00。

表 1

使用限價簿分析之五間公司^[2]

Id	ISIN Code	Company	Sector	Industry
KESBV	FI0009000202	Kesko Oyj	Consumer Defensive	Grocery Stores
OUT1V	FI0009002422	Outokumpu Oyj	Basic Materials	Steel
SAMPO	FI0009003305	Sampo Oyj	Financial Services	Insurance
RTRKS	FI0009003552	Rautaruukki Oyj	Basic Materials	Steel
WRT1V	FI0009000727	Wärtsilä Oyj	Industrials	Diversified Industrials

數據通過 ITCH flow 將資訊轉變成實作數據。ITCH flow 經由 C++ 將相關的文字、事件和交易行為都轉換為以日為單位經過分類的 h5 檔案。再來再經由 MATLAB 程式重新重組句子和限量委託簿以及其他的資訊。最終再經過機器學習得出最終實作使用資料集。

3.2 結果驗證標準

對於實作的可行性和評價，我們需要建立標準去評斷不同模型間的指標。其中我們使用了 Accuracy、Precision、Recall、F1 score 作為評斷的指標，而其中定義為

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

TP/FP 為真/偽陽性、TN/FN 為真/偽陰性，而實作中我們側重 F1 score。由於 F1 score 的計算方式由 Precision 和 Recall 組成，因此評斷上不會因為高度不對稱資料而失去意義，而能夠捕捉真陽性以及真實達到漲跌和持平的三種分類。

四、實作架構建置

4.1 資料前處理

對於深度學習網路而言除了訓練最適的權重外，數據特徵標準化對於訓練過程也至關重要，若未經過標準化的數據，容易在訓練過程中需要過多的迭代或是造成梯度爆炸或是消失而導致訓練失敗。對於提供的數據，我們嘗試了三種特徵標準化的方式: z-score, min-max 以及 decimal precision 的方式，而經由 baseline 測試比較後，選擇使用了 z-score 標準化的方式進行。

z-score 標準化的方式如下公式所示:

$$x_i^{(Zscore)} = \frac{x_i - \frac{1}{N} \sum_{j=1}^N x_j}{\sqrt{\frac{1}{N} \sum_{j=1}^N (x_j - \bar{x})^2}}$$

先將所有資料的特徵做分類，將個別特徵去平均後除上標準差而得到標準化資料。

4.2 資料特徵提取與標籤

在獲得了特徵標準化的數據後，我們將數據中的委買/賣的價格和數量作為訓練的特徵，可以將 input 看為是二維的矩陣，而每個行則是由 $x_t = [p_a^{(i)}, v_a^{(i)}, p_b^{(i)}, v_b^{(i)}]_{i=1}^{n=10}$ 的向量為結構，其中 p 為價格(price)、v 為交易量(volume)、a 為賣出(ask)、b 為買入(bid)。我們使用 10 個時刻點的同結構向量作為列，因此輸入資料的維度如下所示:

$$X = [x_1, x_2, \dots, x_{10}]^T \in \mathbb{R}^{10 \times 40}$$

在資料標籤中，我們利用中價去標示價格漲跌的方向。中價定義為下:

$$p_t = \frac{p_a^{(1)} + p_b^{(1)}}{2}$$

若單純比較 p_t 以及 p_{t+k} 的差異來判斷漲跌會失去資訊其中真正的意義。因此我們定義 m_- 為前 k 個中價的平均值而定義 m_+ 為後 k 個中價的平均值，公式如下:

$$m_- = \frac{1}{k} \sum_{i=0}^k p_{t-i}$$
$$m_+ = \frac{1}{k} \sum_{i=0}^k p_{t+i}$$

我們又定義以下的公式:

$$l_t = \frac{m_+ - p_t}{p_t}$$

作為 t 時刻時的漲跌方向，我們又訂一個閾值 α 做為判別漲跌的參數:

$$price\ movement = \begin{cases} +1 & , l_t > \alpha \\ -1 & , l_t < \alpha \\ 0 & , else \end{cases}$$

在數據測驗中我們使用了 5 種 k 預測，其中為 k = 10、20、30、50、100。

4.3 深度學習模型總覽

我們所搭建的模型總參數量為 31,209 個，模型結構如圖 1 所示：

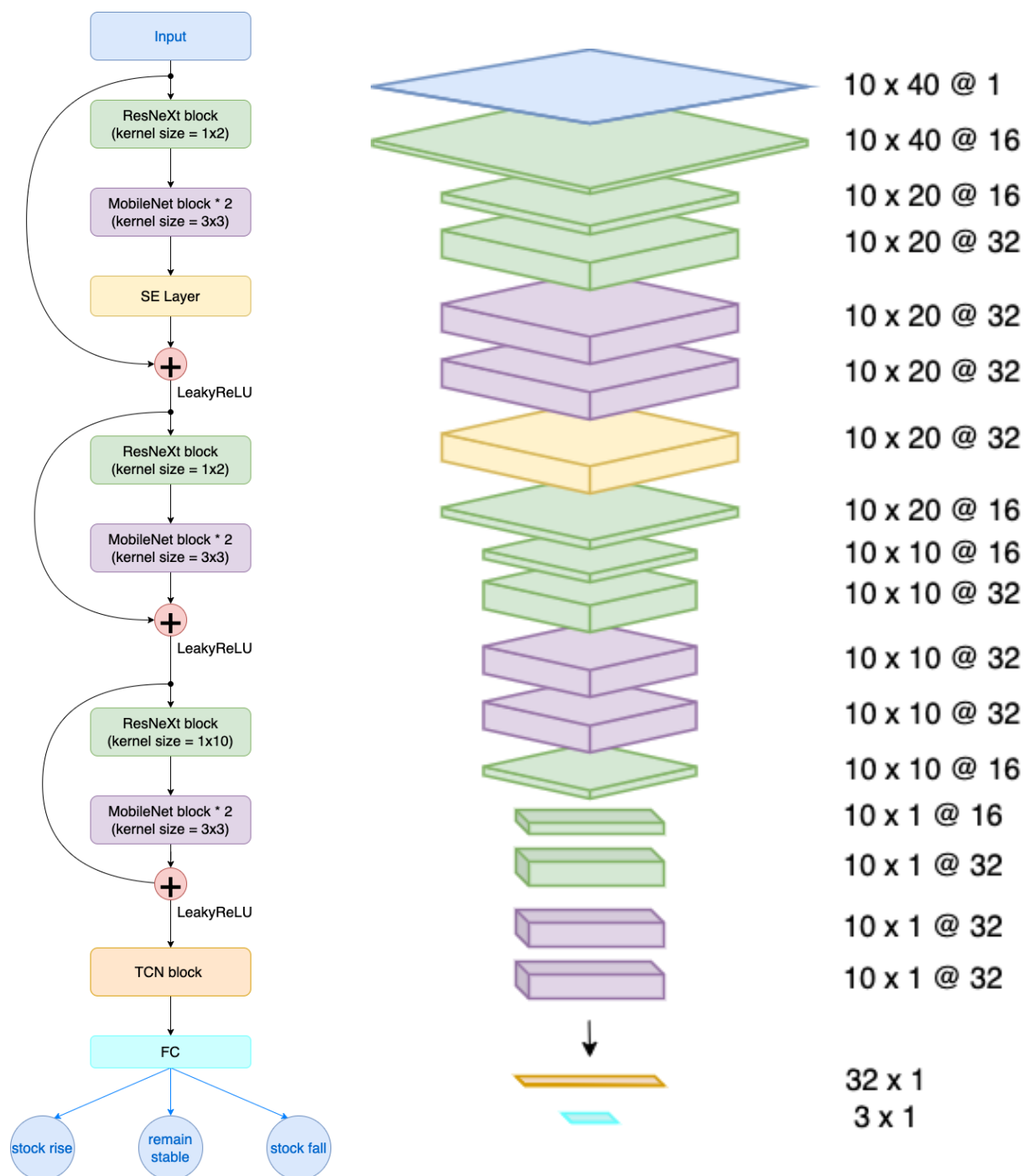


圖 1 本專題實作模型總覽

4.4 深度學習模型詳述

DeepLOB^[1] CNN 架構:

我們沿用了參考論文^[1]的概念架構，只保留原架構中第一層(1×2) CNN filter，來統整價格和交易量的資訊；第二層的(1×2) CNN filter，形成 micro price:

$$p_{micro\ price} = IP_a^{(1)} + (1 - I)P_b^{(1)} \quad , \quad I = \frac{v_b^{(1)}}{v_a^{(1)} + v_b^{(1)}}$$

這項指標的重要性在於它同時考慮到買入和賣出的交易量，而買入和賣出交易量的不平衡又是接下來股價變動趨勢的重要因素；最後再使用(1×10) CNN filter 來統整所有的資訊。我們同時沿用參考論文^[1] 的通道數(channel = 32)。

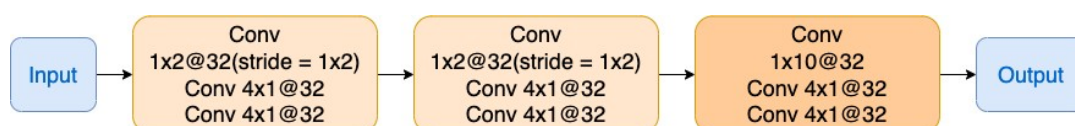


圖 2 DeepLOB^[1] CNN 結構

ResNet^[3]:

我們參考 ResNet^[3]當中的殘差(residual)結構，目的在於解決一系列卷積層堆疊造成的梯度消失或梯度爆炸問題，以及退化(degradation)問題。殘差結構就是於主分支(原卷積層)外另外加上捷徑分支，之後再進行兩分支輸出端的矩陣相加。

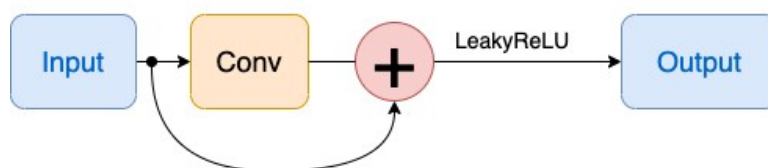


圖 3 ResNet^[3] 結構

ResNeXt^[4]:

在 ResNet^[3]之上再加上 Inception 結構(split-transform merge)以及對 channel 分組的概念，形成了 ResNeXt^[4]的架構，這樣的架構能夠在不增加參數量的情況下提高準確度，而這樣的模型結構更加的模組化，也減少了超參數的數量。

Inception 結構為在原先的卷積層之前先透過 1×1 卷積層降維，再進行原先的卷積，接著再做升維；而 channel 分組則是將原先的卷積層分組為多個相同的卷積層，個別進行卷積運算之後再融合起來，這個概念也稱作 Cardinality。

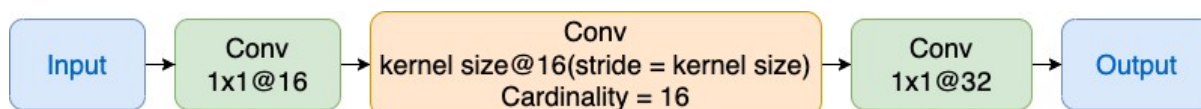


圖 4 ResNeXt^[4] 結構

SENet^[6]:

SENet^[6]的作法為先壓縮(Squeeze)channel，通常是做 average pooling，再做激發(Excitation)，通常為兩個 fully connected layer(降維再升維)，由於這樣的結構參數非常少，因此好處就在於使用非常少的參數卻可以增進模型表現。

SE Layer 我們選擇將 reduction 參數設為 32，代表的意義為先前提到的兩個 fully connected layer 當中，第一個 fully connected layer 將原先 32 個 channel 減少到 $32 / 32 = 1$ 個 channel，第二個 fully connected layer 再升維回到 $1 \times 32 = 32$ 個 channel，針對每個 channel 的重要程度學習。

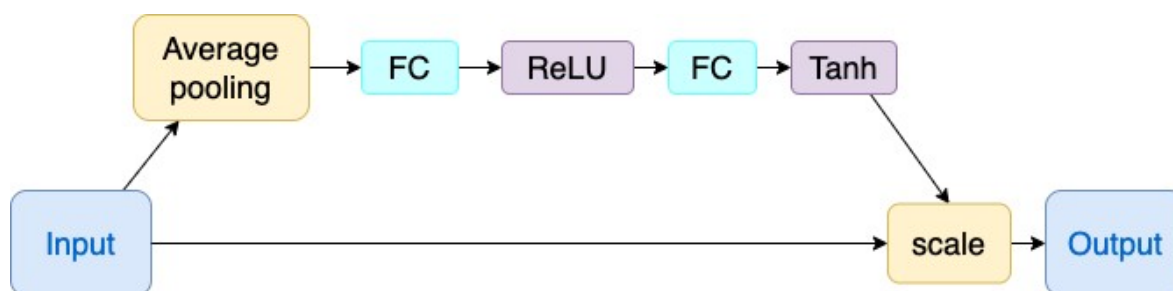


圖 5 SENet^[6] 架構

MobileNet^[7] :

在巨大的網路模型中，大部分的計算量都花費在卷積計算上，因此使用了 Depthwise Separable Convolution 這樣的技巧，在盡量不減低模型表現的同時，去減少原本在 convolution 當中的計算量。

Depthwise Separable Convolution 主要可以分成 depthwise convolution 和 pointwise convolution 兩個部分: depthwise convolution 針對每一個 input channel 都建立一個 kernel，每個 channel 都針對對應的 kernel 分別做卷積運算；pointwise convolution 則是代表在每個 channel 各自做完卷積操作後，再針對每個點的所有 channel 做 pointwise convolution，藉此來融合不同 channel 之間的資訊，達到和一般正常做卷積類似的效果。

我們在加上 MobileNet^[7]架構之後，模型準確度反而獲得了提升，推論是因為模型架構中已經使用了 SENet^[6]架構，依據每一層通道的重要性給予不同權重，彌補 MobileNet^[7]喪失通道之間訊息的缺點。與此同時，pointwise convolution 是針對每一個資料點進行學習，而我們的輸入資料也是每個資料點有其特別的意義，不像圖像資料與鄰近資料點有強大關聯，因此這樣的學習模式反而對我們的模型學習有所幫助。

TCN^[8]:

實驗證明一個簡單的卷積結構在不同的任務和數據集範圍內都優於典型的循環網路如 LSTM 等，同時也顯示出具有更長的有效記憶。TCN 的特點在於使用了因果卷積，沒有洩漏未來時刻點的資訊，使用了殘差結構和擴展卷積來增強網路記憶。擴展卷積(dilated convolution)透過跳過部分輸入來使 filter 的感受野(Receptive field) 可以更大，解決了因果卷積需要多層 filter 或者很大的 filter 才能增加卷積感受野的問題。

這樣的技巧可以在不增加參數和模型複雜度的條件下，指數倍的擴大感受野的大小；為了避免很深的神經網路梯度爆炸或梯度消失的問題，在 TCN 的結構中也加入了殘差(residual)的結構，改善模型的表現。透過 TCN 的架構，利用足夠大的感受野來獲取時序列上的歷史資訊，使其可以作為 RNN、LSTM 等的替代品。我們選擇 dilation=2、總層數 2、kernel size 1×4 來建構我們的模型，已達到最好的學習效果。

此外，我們試著將目光放到其他影響模型好壞的因素上。我們採取其中「訓練及正則化方法」當中的「增加 training epoch 次數」、「cosine LR decay」等概念，來對我們模型的訓練方式加以改善。

五、實作數據驗證和討論

5.1 實作前置設定

我們在實作當中唯一的操作變因是在標籤時所使用的 k 值。在訓練時，我們使用的優化器為 Adam，其可以對過去梯度方向做梯度速度調整。使用的學習率為 0.007。有別於原作者^[1]使用的固定的學習率，我們採用 cosine annealing learning rate。而 learning rate 的改變曲線可由下列公式說明：

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min}) \times (1 + \cos\left(\frac{T_{cur}}{T_{max}}\pi\right))$$

我們採用的 T_{max} 為 8 以及我們使用的總訓練迭代次數為 72。而使用的批次訓練數量為 32，這使得訓練時間不會過久，但又能足夠捕捉資訊的輪廓。

模型的訓練完成後，我們獲得了一個參數訓練完成後的模型，而我們利用 NVIDIA 提供的深度推論平台-TensorRT 達到推論加速。TensorRT 能在 GPU 上實現低延遲(latency)、高吞吐量(throughput)的效果，最主要的優化方式為層間融合(Layer Fusion)和數據精度校準(Precision Calibration)。

由於真正耗時的不是在於 CUDA 的計算時間，而是在於啟動不同的 CUDA 以及對每一層輸入、輸出 tensor 的讀寫操作上，因此 TensorRT 透過對模型當中結構相同的層之間的融合，以及 convolution、batch normalization、activation function 之間的融合，使得占用的 CUDA 數量大幅減少，模型結構變得更加高效。

數據精度校準使用到量化(quantization)的技巧，神經網路訓練時的 tensor 都是 32 位浮點數的精度，但是一旦模型訓練完成，不再需要反向傳播(back propagation)，將 tensor 量化至 16 位浮點數，甚至 8 位整數的精度來做模型推論，再透過 TensorRT 提供的完全自動化校準過程，來最小化量化造成的性能損失，透過這樣的技術，來達到更低佔用內存、模型延遲更少的效果。

在本專題中，我們使用翁詠祿教授-**錯誤更正碼理論與積體電路設計實驗室**中的伺服器做為實作硬體環境。參照下方表 2。

表 2
硬體實作環境

	Process information	Memory	Disk Space	Graphic card	OS version
workstation	Intel® Core™ 19-10980XE@ 3.00GHz	256 GB DDR4	3400.0 GB	GTX 2080 ti 11GB VRAM	CentOS 7.9

5.2 實作測試與結果

我們在實作當中與原作者使用相同的兩種測試方式作為 baseline 之間的比較。第一種測試方式我們使用前 x 天的資料訓練模型後使用第 $x+1$ 天的資料於驗證模型， x 為 1 至 9 天。而第二種測試方式則為使用前 7 天的資料訓練後，去驗證後 3 天的資料。

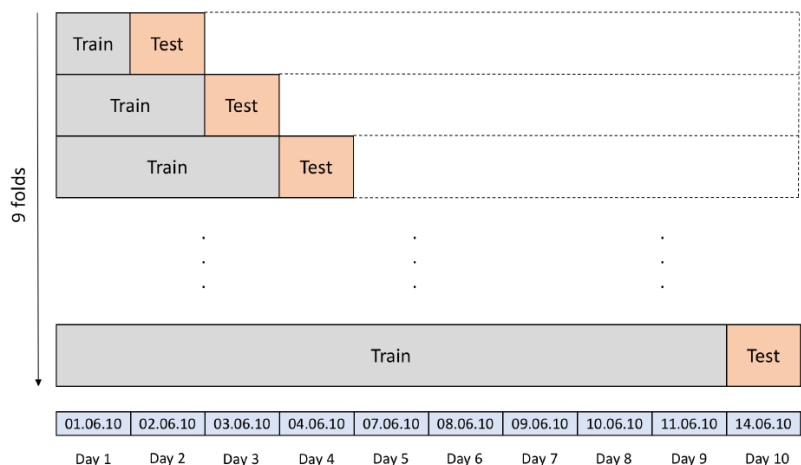


圖 6 第一種測試方式^[1]

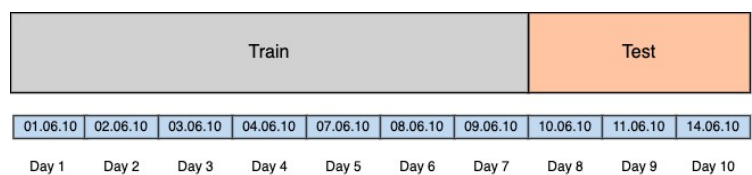


圖 7 第二種測試方式^[1]

我們採取與參考論文^[1]相同的驗證方式，採用比較 state-of-the-art 的方式來比較各模型對於 FI-2010 dataset 的四項指標:Accuracy、Precision、Recall 以及 F1-score。在原論文中與之比較的模型為 RR^[2]、SLFN^[2]、LDA^[9]、MDA^[9]、MTR^[9]、WMTR^[9]、MCSDA^[10]、BoF^[11]、N-BoF^[11]以及 B(TABL)^[12]和 C(TABL)^[12]、SVM^[13]、MLP^[13]、LSTM^[13]、CNN-I^[14]、CNN-II^[14]。

在第一種測試方式中，我們使用先前提到的 k 個中價來做為漲跌判斷的標籤，我們使用了 $k=10$ 、50、100。而於第二種測試方式中我們使用了 $k=10$ 、20、50。

表 3 和表 4 中為我們的實作結果，其中表中黑字為原論文^[1]提出與其模型比較之 state-of-the-art 模型表現，而橘字即為原論文^[1]中所提出之模型，藍字為我們實作的模型。

從表格中可以看出我們無論是方法一或方法二中，使用的各項 k 值皆可以超越參考論文中的四項指標，這驗證了我們在於模型建置上的選擇是優於原作者的選擇^[1]。我們能更加正確地捕捉了時序列之中的訊息，準確地預測出了股市漲跌之預測。

而除了精準度之外，我們在模型的參數量上也達到了 50%的減少，可以減少硬體的運算時間以及數據推論得出結果的時間延遲。

Latency 的計算方式考慮到實際情況會以 batch size 等於 1 的情況下送進 model 處理，因此雖然在訓練模型時我們是用 batch size 等於 32 來做訓練，但在測試 latency 時我們使用 batch size 等於 1 來量測，較符合真實情況。

表 5 中為我們與其於 state-of-the-art 的參數之間比較，在參考論文^[1]中，作者使用的硬體和我們測試時的硬體環境並非相同，因此我們還原作者的架構在相同的硬體下進行實作而得出表 5 之數據表現。

表 3

第一種測試方式與其 baseline 比較

Model	Accuracy %	Precision %	Recall %	F1 %
Prediction Horizon k = 10				
RR [2]	48.00	41.80	43.50	41.00
SLFN [2]	64.30	51.20	36.60	32.70
LDA [9]	63.83	37.93	45.80	36.28
MDA [9]	71.92	44.21	60.07	46.06
MCSDA [10]	83.66	46.11	48.00	46.72
MTR [9]	86.08	51.68	40.81	40.14
WMTR [9]	81.89	46.25	51.29	47.87
BoF [11]	57.59	39.26	51.44	36.28
N-BoF [11]	62.70	42.28	61.41	41.63
B(TABL) [12]	73.62	66.16	68.81	67.12
C(TABL) [12]	78.01	72.03	74.04	72.84
DeepLOB[1]	78.91	78.47	78.91	77.66
Our model	80.73	80.62	80.73	79.61
Prediction Horizon k = 50				
RR [2]	43.90	43.60	43.30	42.70
SLFN [2]	47.30	46.80	46.40	45.90
BoF [11]	50.21	42.56	49.57	39.56
N-BoF [11]	56.52	47.20	58.17	46.15
B(TABL) [12]	69.54	69.12	68.84	68.84
C(TABL) [12]	74.81	74.58	74.27	74.32
DeepLOB[1]	75.01	75.10	75.01	74.96
Our model	77.93	78.14	77.93	77.90
Prediction Horizon k = 100				
RR [2]	42.90	42.90	42.90	41.60
SLFN [2]	47.70	45.30	43.20	41.00
BoF [11]	50.97	42.48	47.84	40.84
N-BoF [11]	56.43	47.27	54.99	46.86
B(TABL) [12]	69.31	68.95	69.41	68.86
C(TABL) [12]	74.07	73.51	73.80	73.52
DeepLOB[1]	76.66	76.77	76.66	76.58
Our model	81.23	81.29	81.23	81.21

表 4

第二種測試方式與其 baseline 比較

Model	Accuracy %	Precision %	Recall %	F1 %
Prediction Horizon k = 10				
SVM [13]	-	39.62	44.92	35.88
MLP [13]	-	47.81	60.78	48.27
CNN-I [14]	-	50.98	65.54	55.21
LSTM [13]	-	60.77	75.92	66.33
CNN-II [15]	-	56.00	45.00	44.00
B(TABL) [12]	78.91	68.04	71.21	69.20
C(TABL) [12]	84.70	76.95	78.44	77.63
DeepLOB[1]	84.47	84.00	84.47	83.40
Our model	84.89	84.36	84.89	84.02
Prediction Horizon k = 20				
SVM [13]	-	45.08	47.77	43.20
MLP [13]	-	51.33	65.20	51.12
CNN-I [14]	-	54.79	67.38	59.17
LSTM [13]	-	59.60	70.52	62.37
CNN-II [15]	-	-	-	-
B(TABL) [12]	70.8	63.14	62.25	62.22
C(TABL) [12]	73.74	67.18	66.94	66.93
DeepLOB[1]	74.85	74.06	74.85	72.82
Our model	76.13	75.30	76.13	74.59
Prediction Horizon k = 50				
SVM [13]	-	46.05	60.30	49.42
MLP [13]	-	55.21	67.14	55.95
CNN-I [14]	-	55.58	67.12	59.44
LSTM [13]	-	60.03	68.58	61.43
CNN-II [15]	-	56.00	47.00	47.00
B(TABL) [12]	75.58	74.58	73.09	73.64
C(TABL) [12]	79.87	79.05	77.04	78.44
DeepLOB[1]	80.51	80.38	80.51	80.35
Our model	81.83	81.74	81.83	81.73

表 5

與參考論文之推論延遲比較

Models	Forward(ms)	Number of Parameters
DeepLOB[1]	2.334	60k
Our Model	0.434	31k

六、 結論

本專題對於開源的資料集進行中價走向的預測策略，最終採取建置深度學習網路，運用 DeepLOB 結構^[1] 對於股市數據結構的特徵解析與數學推演，結合了能夠降低延遲並且同時能夠強化在深度學習中捕捉重要特徵的神經網路，超越了原作者所建立模型之預測之四項指標，並且能夠在相同的硬體環境達到五分之一延遲。

在模型的優化過程中，我們參考了許多 ImageNet 架構，並融合其概念進入模型當中，由於我們的測資終究與圖像資料不盡相同，每個「像素」所代表的涵意更是與圖像資料有所差距，因此融入各種 ImageNet 時並不是只有將其架構直接套入我們的模型，而是更進一步根據原論文了解其核心概念，再將優化的關鍵以我們模型的架構模式加以導入，藉以提升我們模型的準確度，其中也充滿許多挑戰。

針對模型的表現，除了 Accuracy 的單純提升外，我們在 F1 score 能夠獲得提升，說明著並非只有對於各項數據的 label 能夠有正確的判斷，而是奠基於其上能夠預測出相比 True Negative 更為重要的 True Positive。對於亂度非常高，且在短周期的區間內並無規律性的數據而言，是建置演算法策略最大的目標。

本專題使用之資料集為標籤分布不均衡，因此在達成上述之目標又增添了困難度，而也因為如此我們對於訓練時所使用的 timestamp 數量也從 100 降為 10，與其捕捉較大時間區段的資訊，反而更加著重於瞬態的走向，與此同時也獲得了訓練模型速度較快之優勢以及推論時間延遲之降低。

本專題使用影像處理相關演算法先對時序列資料捕捉其中重要的資訊後，再使用時間卷積網路去解析資訊其中乘載的股市走向。此種模式取代了使用記憶性的時序深度網路，減少單一網路需要的參數量，但依然能對於時間序列的特徵做出辨別並且做出標籤判斷。可以作為相關情境訊號處理的新的想法出發點。

對於本專題，我們認為依然有很大的空間可以延伸。首先是軟體於硬體的實現，演算法於邊緣裝置(edge device)的實現會因為現實情況而受到妥協，精準度和延遲是否能夠於系統上維持表現成為最大的困境。另外，演算法的通用受到開源資料透明度的限制，但金融資料的透明度逐日提升，因此如何能以個人單位制定捕捉訊號特徵和統整的機制將成為此領域的突破口。最終，我們在時序資料的處理上同時使用了影像處理與時序分析之網路結構，但計算負擔小且所需參數少之時序模型仍是訊號處理相關尚未突破方向，若能成功制定正確的演算法，將會帶給訊號處理領域和生活應用的一大突破。

七、 参考文献

- [1] Zhang, Z., Zohren, S., & Roberts, S. (2019). Deeplob: Deep convolutional neural networks for limit order books. *IEEE Transactions on Signal Processing*, 67(11), 3001-3012.
- [2] Ntakaris, A., Magris, M., Kannianen, J., Gabbouj, M., & Iosifidis, A. (2018). Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods. *Journal of Forecasting*, 37(8), 852-866.
- [3] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- [4] Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- [5] Bello, I., Fedus, W., Du, X., Cubuk, E. D., Srinivas, A., Lin, T. Y., ... & Zoph, B. (2021). Revisiting ResNets: Improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 34, 22614-22627.
- [6] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7132-7141).
- [7] Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., ... & Adam, H. (2019). Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1314-1324).
- [8] Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.
- [9] D. T. Tran, M. Magris, J. Kannianen, M. Gabbouj, and A. Iosifidis, "Tensor representation in high-frequency financial data for price change prediction," in *Computational Intelligence (SSCI), 2017 IEEE Symposium Series on*. IEEE, 2017.
- [10] D. T. Tran, M. Gabbouj, and A. Iosifidis, "Multilinear class-specific discriminant analysis," *Pattern Recognition Letters*, vol. 100.

- [11] N. Passalis, A. Tefas, J. Kanninen, M. Gabbouj, and A. Iosifidis, “Temporal bag-of-features learning for predicting mid price movements using high frequency limit order book data,” IEEE Transactions on Emerging Topics in Computational Intelligence, 2018.
- [12] D. T. Tran, A. Iosifidis, J. Kanninen, and M. Gabbouj, “Temporal attention-augmented bilinear network for financial time-series data analysis,” IEEE transactions on neural networks and learning systems, 2018.
- [13] “Using deep learning to detect price change indications in financial markets,” in Signal Processing Conference (EUSIPCO), 2017 25th European. IEEE, 2017.
- [14] A. Tsantekidis, N. Passalis, A. Tefas, J. Kanninen, M. Gabbouj, and A. Iosifidis, “Forecasting stock prices from the limit order book using convolutional neural networks,” in Business Informatics (CBI), 2017 IEEE 19th Conference on, vol. 1. IEEE, 2017.
- [15] “Using Deep Learning for price prediction by exploiting stationary limit order book features,” arXiv preprint arXiv:1810.09965, 2018.

八、計畫管理與團隊合作方式

(一) 計畫管理

本次專題在前半學期由教授與學長提供相關領域的論文與文本進行研讀，熟悉金融資料之處理策略與演算法，並且從中獲得專題前進方向。並且制定進度匯報以確認對於方向以及理解是否有不熟稔或錯誤之處。

而在後半學期，便由學長和組員之間制定進度，並且由斷定好的演算法進行實作，規劃每周應完成事項，一樣維持著定期的匯報，以確認內容上的精確性以及是否能如期完成訂下的規劃。

(二) 團隊合作

本專題無論是前期的資料研讀或是到後期的模型建置以及模型訓練，組員間都會自主安排並分配應該完成的事項以有利進度的進展並且在固定的討論時間提出想法討論並且合併成可實現的計畫。

論文、文本閱讀：陳昭維、林禹陞

模型建置與數學推導：陳昭維、林禹陞

報告、摘要、海報製作：陳昭維、林禹陞