題目:LSTM深度學習預測股票之模型

大家好，我們這組做的是 LSTM 深度學習預測股票之模型，為什麼 LSTM 這個model 呢?

首先，股價擁有 time series(時序)的特性，也就是昨天的股票和走勢對於今天的股價或多或少還是有影響，在之前的作業中我們已經透過5日均線、20日均線等技術指標作為交易進出的依據。

那 LSTM 在處理 sequential 資料上會將時間的屬性考慮進去，我們利用這個深度學習演算法搭配交易策略來預測未來的股價，主要以八支股票，包和台股(台積電、長榮、友達光電)及美股(Microsoft、Apple、Amazon、AMD)前五十天的股價去預測明日的股價。

我們簡介一下LSTM 第一，是基於RNN的架構，會不斷由前面的資料來預測現在的資料，理論上時間越長預測會越準，但RNN的缺點就是記憶力差，越前面的資料隨著時間會漸漸被遺忘。

第二，LSTM透過 Memory Design 來增加所謂的"長期依賴"(long-term dependency)。 LSTM 由四個 unit 組成：Input Gate、Output Gate、Memory Cell以及Forget Gate。

除了 Output Gate 為預測的輸出，Memory Cell 為數值運算後的記憶位置外， 由 Forget Gate 及 Input Gate 組成的記憶分支會隨著時間更新，來決定是否更新記憶

接著看資料實際 input 進 LSTM cells 的流程，數學上表示為 g(z)，第一個遇到的 input gate 使用 Activation function f(Zi) 來表示 input gate 開啟的機率。

第二個 Memory cell 會運算當下 input 值加上前一次 Memory cell 裡的值並乘上 forget gate 的機率，若結果與上一筆差距很大，或是一個從未出現過的值，表示上一筆 data 參考價值不大，會被過濾掉，反之就會繼續被保留在記憶中，最後的 output gate 會確認是否把值輸出，也是以機率的方式。

下面實作採用的是 LSTM 中 many to one 的類型，因為我們是以多個時間點來預測下一個時間點。 :::warning 以上圖片取材於李弘毅教授 [ML Lecture 21-1: Recurrent Neural Network (Part I)] :::

實作部份:

- Dataset Dataset的是利用 AlphaVantage 跟 Finmind 這兩個套件來收集資料，AlphaVantage 以美股資料為主，而 Finmind 則是收集台股。我們主要會用到的資料是"開高收低量"再加入"DCO"跟"DHL"七種feature，將資料下載下來後存為csv，這樣之後可以隨時使用這些資料。
- Preprocessing 我們的目標是想要利用過去50天的股票歷史紀錄來預測第二天的開盤價。首先先將資料做正規化，為了提高網路的收斂速度，需要將資料做 scaling，利用 sklearn 的 preprocessing 這個 library 來將資料 scale 成0到1之間。我們想要取得正規化的資料X，還有想要預測的資料y以及正規化後的y，X的部分是將每50筆歷史資料去做正規化，並存成 numpy 陣列，而y則是取50天後的開盤價以及正規化後的資料，且為了讓資料可以放到之後的 model 裡面，透過 expand_dims 這個方法去展維成二維。同時我們保留了 y_normaliser 這個變數，因為 model 會輸出一個介於0到1之間的數字，我們可以利用這個變數把正規化後的資料轉換回真實的數值，之後可以用它來計算誤差。此外我們還新增了 SMA 這個技術指標來提升準確度。
- Split train and test 再來就是將資料切成 training data 跟 testing data，分布為 ohlcv_train: 0.9; y_train: 0.9; ohlcv_test: 0.1; y_test: 0.1; unscaled_y_test: 0.1
- Model

  圖為我們 LSTM 的模型架構，code 的部份也是依照這個架構來設計，在輸入層(也就是第一層)每一筆 input data 的 shape 為(history_points, OHLCV)，history_points是50天代表50個神經元，OHLCV 代表五個價錢變項，那適度增加 dropout layer 可以避免模型 overfitting；透過 dense layers 將 lstm 的 data 更好的聚合在一起，這個 network 很重要是最後的 activation 是 linear_output，能使模型準確地調整其倒數第二層的權重。

  由於最初模型 evaluate 出來的 mse(均方誤差)偏高，我們將技術指標SMA(簡單移動平均線)，作為 network 的額外輸入，那由於 SMA 不是時間序列的 data，我們把他輸入在倒數第二個 64-node的dense layer，模型重新訓練後，得到的 mse 跟原先比起來低了許多，繪製圖形後也可以發現預測是很接近test data 的。

  最後，我們用模型預測出來的價格來判斷買賣點，在價格預測上漲的交易日，買入$10股票，在預測價格下跌的交易日，則賣出全部所持有的股票(全部出清)，由此計算最後的交易獲利。

- **經過演算法交易策略的最終結果:**
- AAPL
- MSFT

- AMZN
- AMD
- 2330
- 2603
- 2409