

機器學習第三次報告

Direct Reinforcement Learning for Financial Signal Representation and Trading (2016),IEEE

宋相錡

7108053112

應數系計算科學班數據組,中興大學

g108053112@mail.nchu.edu.tw

Abstract—此篇論文[1]中提出Fuzzy Direct Deep Neural Network(FDRNN)模型，希望模型經過訓練後，可以通過一連續時間價格的資料，來判讀金融市場上的買賣資訊，從而做出正確的交易決策。此次報告，首先將對論文使用的模型稍加說明，接著描述實作過程中發現的問題，最後再呈現實作成果，並與原論文中的結果做比較。

I. FDRNN 模型

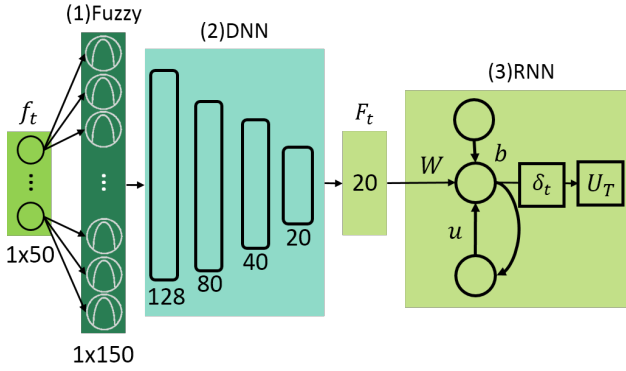


Fig. 1. FDRNN model

我們可以看到Fig.1即是FDRNN模型的全貌，模型由三個部分所組成。

1. 將取得的金融商品價格定義為 $p_1, p_2, \dots, p_t, \dots$ 的數列，接著可以計算出與前一個時間的價差 $z_t = p_t - p_{t-1}$ ，再取50個價差作為我們模型的輸入，即是 f_t 。

$$f_t = [z_{t-50-1}, \dots, z_t] \in \mathbb{R}^{50} \quad (1)$$

為凸顯輸入的準確性，且希望讓模型知道每個價差應當屬於何種趨勢，將每一個 f_t 的元素通過fuzzy membership function $v_i(\cdot) : \mathbb{R} \rightarrow [0, 1]$ 的轉換，轉換成3個隸屬度，得出一個150維度的向量。

2. 得到了fuzzy representation 後，我們再將 $v(f_t)$ 傳入非線性轉換函數 $g_d(\cdot)$ ，DNN做運算。DNN為4層隱藏層的結構，分別具有128、80、40、20個神經元，通過DNN後我們得到 $F_t = g_d(v(f_t))$ 的輸出，其中DNN使用到的activation function為sigmoid function。
3. 來到模型的最後一步，RNN依據 F_t 來判別交易決策 δ_t ，通過模型的訓練與跟新，盡可能達到總獲利 U_T 的極大化。

經由上述的三個步驟，得出我們的模型，以下的數學式為目標函數，與模型的設定。

$$\begin{aligned} \max_{\Theta=v(\cdot), g_d(\cdot), W, b, u} U_T \\ \text{s.t. } R_T = \delta_{t-1} z_t - c |\delta_t - \delta_{t-1}| \\ \delta_t = \tanh[< W, F_t > + b + u \times \delta_{t-1}] \\ F_t = g_d(v(f_t)) \end{aligned} \quad (2)$$

II. 實驗過程

A. RNN參數更新問題

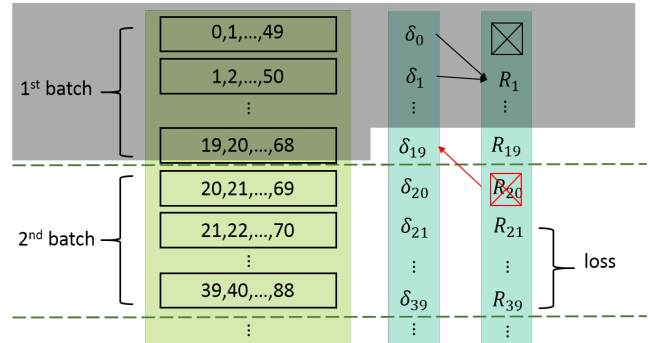


Fig. 2. RNN參數更新示意圖

在寫Tensorflow的過程中發現，設定讀入目標函數所使用變數的placeholder時，維度必須比我們給定的batch size少1，這原因來自於使用RNN演算法的問題本身。

舉個例子來看，也就是上方Fig.2中所呈現的狀況。首先給定batch size為20，在第一個batch的時後，模型一次批量讀入20筆資料，經過運算得出對應的20個 $\delta_t(\delta_0, \delta_1, \dots, \delta_{19})$ ，與每兩個 δ_t 計算出的 $R_t(R_1, \dots, R_{19})$ ，向前傳遞完畢，並將這19個 R_t 加總作為目標函數去更新模型的參數。

前述狀況會從第二個batch開始發生，第二個batch批量讀入20筆資料，並產生20個 $\delta_t(\delta_{20}, \delta_{21}, \dots, \delta_{39})$ ，同時可以計算出20個 $R_t(R_{20}, R_{21}, \dots, R_{39})$ 。但是在向後傳遞時，由於計算上RNN不能無止盡的遞回，因此給定的batch size，把RNN的遞回次數加上限制，也就是在更新第二個batch時，並不會考慮到前個batch做出的決策(δ_{19})，也就不會考慮到 R_{20} ，因此第二個batch的目標函數為 R_{21} 到 R_{39} ，這19個 R_t 的加總，以此優化做出決策的模

型參數。之後的每個batch同樣都只會使用到19個 R_t 的加總來更新參數。

B. 資料與輸入設定

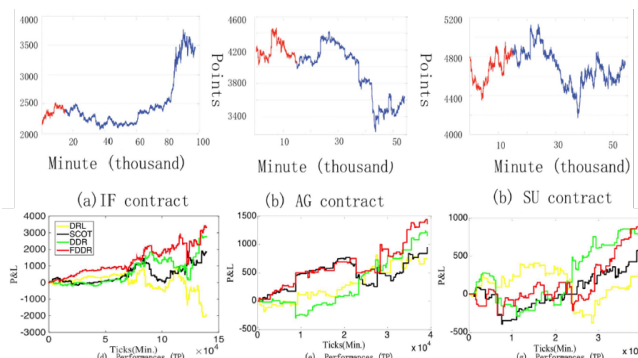


Fig. 3. 論文實驗資料與成果

論文使用中國股票市場中的滬深300ETF指數、期貨市場的糖與白銀的價格，作為實驗資料集，每個資料集分別有15萬、5萬、5萬筆資料，且每筆資料的時間間距為1分鐘。另外，論文中輸入 f_t 的設定並非直接取前50分鐘內的價差變化，而是，其中45個維度放前45分鐘內每天的價差，另外5個維度分別放入3小時、5小時、1天、3天、10天，與此刻的價差。論文認為這個輸入的設計，可以將更久遠以前的資訊考慮進來，以助模型判斷趨勢的變化。

由於資料取得的困難性，我在實作中採用的資料集有以下3個，台灣股市的台積電股價(4919筆)、美國股市的SPDR黃金ETF(3808筆)、外匯市場的歐元美元匯率(4166筆)，且每筆資料的時間間距為1天。而輸入則參考論文的設定，包含前45天內每天的價差，以及50天、60天、80天、120天、240天，與此刻的價差。

III. 實驗結果

下圖的黃線皆是作為訓練使用，約佔全部資料的一半。

A. 台股:台積電股價

模型的超參數設定: batch size為20、learning rate為0.001，訓練時總共跑了30個epoch。

台積電股票價差的變化，相對於其他兩個資料集漲跌幅較大，在訓練資料的部分呈現震盪走勢，而測試資料為上升趨勢，兩者有明顯的差異。模型經過30個epoch的訓練，loss function震盪向下，表示模型有在優化決策，訓練完模型後進行資料測試，結果發現，一開始走勢由震盪轉為上行，模型還無法即時做出正確決策，使得累積報酬難以增加，至2/3時間的累積報酬有明顯上升，一度增加至153.08的獲利，等同於平均年報酬有14.37%，但是當盤勢短期內快速發生變化，模型無法即時做出適當決策，從而導致累積報酬減少，測試資料的最終平均年報酬率為6.17%。

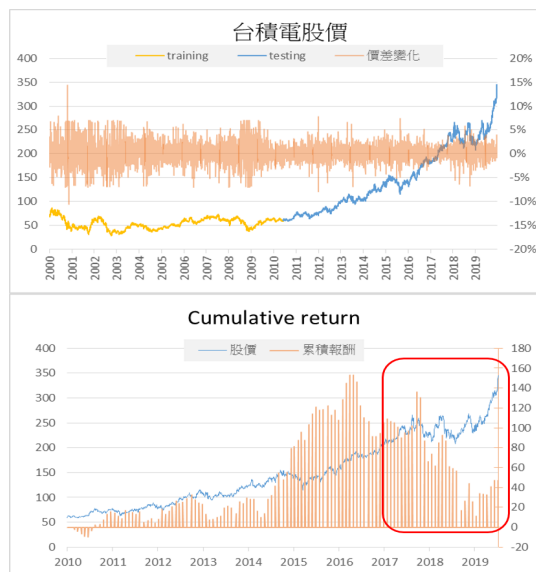


Fig. 4. 台積電股價與模型累積報酬變化

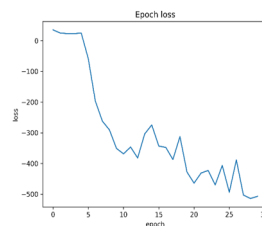


Fig. 5. 使用台積電資料的loss function

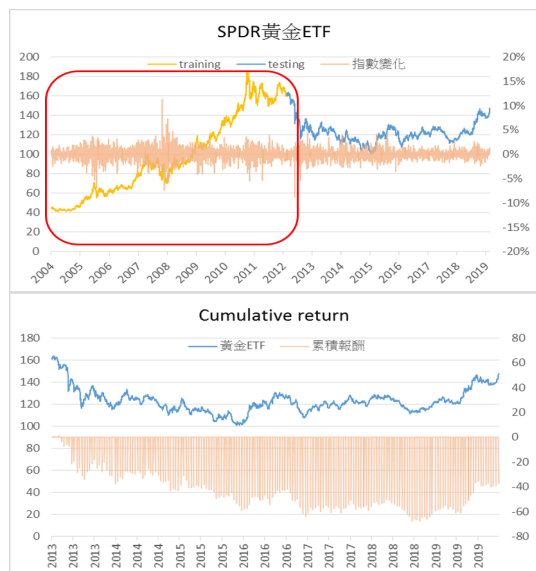


Fig. 6. 黃金ETF指數與模型累積報酬變化

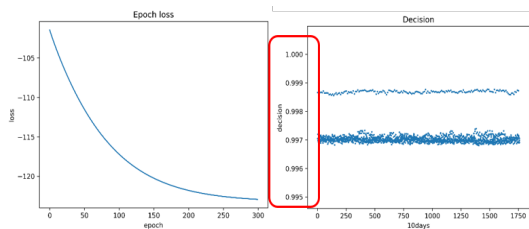


Fig. 7. 使用黃金ETF資料的loss function與決策

B. ETF:SPDR黃金ETF

模型的超參數設定: batch size為10、learning rate為0.000001，訓練時總共跑了300個epoch。

黃金ETF指數的變化，在訓練資料的部分呈現強勁的上升趨勢，而測試資料先下跌了一小段時間再進入整理盤勢，兩者有明顯的差異。模型經過300個epoch的訓練，loss function下降並漸趨收斂，表示模型有在更新參數，但測試結果發現，模型並沒有學到如何做出正確決策，而是一昧的只買不賣，由圖可知模型全部做出近似1的決策(δ_t)。我們發現當訓練資料的單一趨勢過強，訓練出的模型容易不斷地做相同決策。

C. 外匯:歐元美元

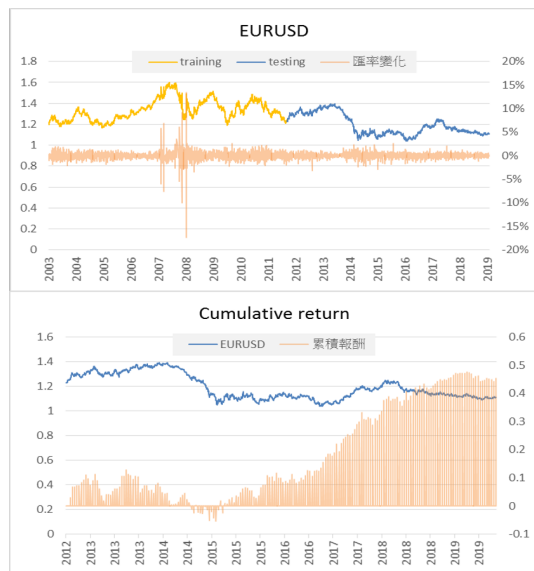


Fig. 8. 歐元美元與模型累積報酬變化

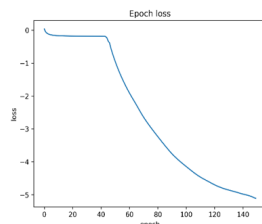


Fig. 9. 使用歐元美元資料的loss function

模型的超參數設定: batch size為10、learning rate為0.0001，訓練時總共跑了150個epoch。

歐元美元外匯的變化，在訓練資料、與測試資料的趨勢皆呈現震盪。模型經過150個epoch的訓練，loss function下降並漸趨收斂，表示模型有在優化決策，測試結果發現，累積報酬有穩定的增加。我們發現訓練資料為震盪格局，模型比較容易訓練，同時漲跌幅度相對較小，模型會有不錯的累積報酬。

REFERENCES

- [1] Youyong Kong Zhiquan Ren Yue Deng, Feng Bao and Senior Member Qionghai Dai. *Deep Direct Reinforcement Learning for Financial Signal Representation and Trading*. IEEE Transactions on Neural Networks and Learning Systems, 2016.