

機器學習第二次報告

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

宋相錡

7108053112

應數系計算科學所數據組,中興大學

g108053112@mail.nchu.edu.tw

Abstract—為訓練機器人，學會判讀金融市場上的資訊，從而做出金融商品的交易決策。此次報告，將從如何做特徵學習，到如何決定交易策略，詳細介紹[1]這篇論文所使用的模型與方法，並對實作的方向稍加論述。

I. INTRODUCTION

在訓練模型中，有兩大挑戰必須要克服，第一個是金融市場上繁雜充滿噪音的資訊，要如何匯整且找到重要的特徵表示方式；第二個是如何優化模型的決策，以實現報酬的最大化。論文的模型使用Fuzzy representation與Deep neural network (DNN) 解決第一個問題，緊接著將通過DNN的輸出，再放到Recurrent Neural Network (RNN) 中學習如何做出最佳的交易決策，我們稱此模型為Fuzzy Deep Recurrent Neural Network (FDRNN)，Fig.1即是模型的全貌。

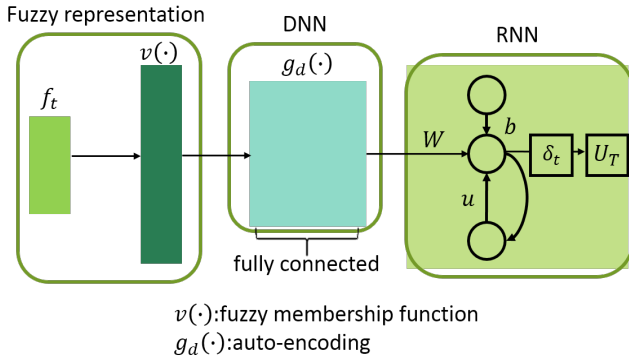


Fig. 1. FDRNN model

首先我們將在Sec.2中，介紹模型會使用到的一些變數與目標函數，隨後在Sec.3一一解說模型使用到的各種方法，緊接著論文應用了一些參數初始化方法作為預訓練，我們將在Sec.4做說明，最後Sec.5會針對實作內容做進一步講解。

II. DEFINING VARIABLES

將取得的金融商品價格定義為 $p_1, p_2, \dots, p_t, \dots$ 的數列，接著我們可以計算出與前一個時間的價差 $z_t = p_t - p_{t-1}$ ，再將價差的時間數列轉成 m 維向量，這個向量就是我們模型的輸入。

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

基於實際的交易策略(action)，我們可以將其區分為三種， $\delta_t \in \{\text{做多}, \text{平倉}, \text{做空}\} = \{1, 0, -1\}$ ，所做出的決策。有了以上的變數，可以進一步定義時間點的獲利 R_t ，即是藉由訓練模型所得到的獲利，

$$R_t = \delta_{t-1} z_t - c |\delta_t - \delta_{t-1}| \quad (2)$$

R_t 的第一項為買賣的價差所產生的盈虧，第二項為交易成本，當 $\delta_t \neq \delta_{t-1}$ 時，則需要繳交強制性費用，例如交手續費給證券公司、交易稅， c 代表的是這些交易成本。在目標函數中加入成本項，是希望避免過度頻繁的交易，而造成巨大的支出。

有了每個時間點的方程式(2)後，對其做加總 $U_T = \sum_{i=1}^T R_t$ ，而我們希望總獲利可以近可能最大化，因此得到了目標函數。

$$\max_{\Theta} U_T = \{R_1, \dots, R_T | \Theta\} \quad (3)$$

III. FDRNN 模型

藉由觀察Fig.3我們可以了解模型的全貌，此章節將詳細解釋模型使用到的各種方法。

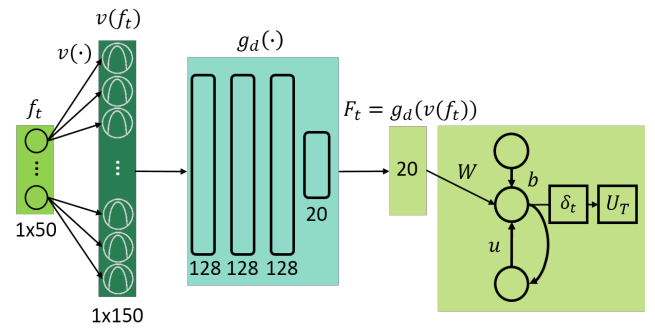


Fig. 2. FDRNN model

A. Fuzzy extensions

首先看到Fig.3的第一步，論文中設定的輸入資料為一個50維的向量 f_t ，為了降低輸入的不準確性、消除市場上存在的眾多雜訊，並讓模型了解到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，

$$v_{i,k}(z_i) = \exp\left\{-\left(\frac{z_i - \mu_k}{\sigma_k}\right)^2\right\}, \forall i = 1, \dots, 50; \forall k = 1, \dots, 3 \quad (4)$$

而這3個隸屬度分別表示為價差隸屬於下降趨勢、平盤、上升趨勢的程度。

方程式(4)使用的是Gaussian membership function，透過此函數將特徵標準化，對價差的差異程度做伸縮，以避免模型只學到極值，其中的平均數 μ_k 和標準差 σ_k 如何設定，會在初始化的章節中做說明。

B. DNN

得到了fuzzy representation 後，我們再將 v_{f_t} 通過非線性轉換函數 $g_d(\cdot)$ ，轉為DNN做運算。論文中設定的DNN為4層隱藏層的結構，分別具有128、128、128、20個神經元(在實做中，我會將不同層間的神經元個數改為陸續減少，逐步得到較為精簡的特徵，比較合理。)，通過DNN後我們得到 $F_t = v(g_d(f_t))$ 的輸出，其中DNN使用到的activation function為sigmoid function。

$$o_i = \frac{1}{1 + e^{a_i}} \quad (5)$$

C. RNN

特徵已經經過兩個步驟的處理，得到帶有精確訊息的 F_t ，並來到模型的最後一步，最終RNN將依據 F_t 來判別出我們的交易決策 δ_t ，通過模型的訓練與跟新，以達到總獲利的極大化。

我們的交易決策由此得出，

$$\delta_t = \tanh[< W, F_t > + b + u \times \delta_{t-1}] \quad (6)$$

其中 $\tanh(\cdot) : \mathbb{R} \rightarrow [-1, 1]$ (實做中將對 $[-1, 1]$ 的範圍進行不同比例的分割，來判定做空、平倉、做多的決策，比較哪種分割方式可以得到較大的總獲利)，觀察方程式(6)為一次遞回，因此RNN可以經由過往的經驗，學習到如何做出正確的決策。

D. Summary of Model

經由上面的三小節，我們知道了FDRNN模型是由三個部分所組成：fuzzy representation、DNN、RNN，它們在模型中分別扮演，降低資料的不準確性、特徵學習、做出交易決策。模型的訓練，須對以下目標函數的參數進行跟新。

$$\max_{\Theta=v(\cdot), g_d(\cdot), W, b, u} U_T \quad (7)$$

IV. SYSTEM INITIALIZATIONS

建構模型完畢後，在訓練模型之前，需給訂參數的初始值，論文中非隨機給定迭代的起始點，而是用以下的方式給予模型做參數初始化，來作為預訓練，以降低找到較差局部最佳解的可能。

在Fuzzy extensions中，使用的K-means無監督式學習的方法，我們給定 $k=3$ ，對資料集中的前1500筆 z_t 做分群，計算出三群的平均數 μ_k 與標準差 $\sigma_k, \forall k = 1, \dots, 3$ ，這

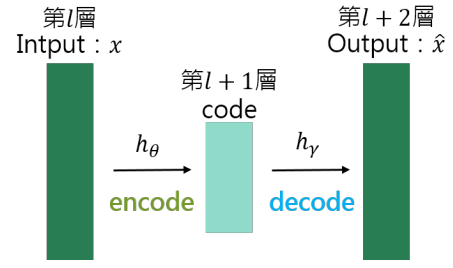


Fig. 3. auto-encoding

三群分別對應到下降趨勢、平盤震盪、上升趨勢，因此我們找到了fuzzy membership function $v_i(\cdot)$ 參數的初始值。

接著我們使用auto-encoding無監督式學習的方法，找出一組比較好的 $g_d(\cdot)$ 來作為DNN的預訓練。auto-encoding的想法是，希望輸入在經過encode與decode後的輸出，可以盡可能的相似，這也就表示中間層的code，雖然被降維但仍然保留了原本輸入的一些特徵。

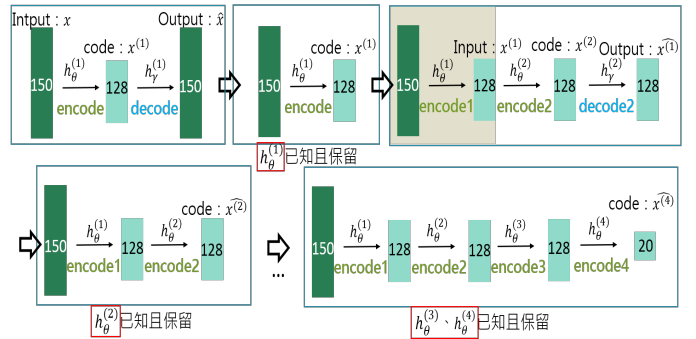


Fig. 4. 使用auto-encoding找出初始參數

Fig.4說明了如何找出DNN四層隱藏層的參數。先藉由一次的auto-encoding，找出第一個code當做DNN的第一層，並且保留第一個encode $h_\theta(1)$ ，接著再做第二次的auto-encoding，找出第二個code當做DNN的第二層，並且保留第一個encode $h_\theta(1)$ ，以此類推，就找到我們要的 $g_d(\cdot)$ 。

V. IMPLEMENTATION DIRECTION

論文中使用中國期貨市場中的，滬深300指數、糖、白銀，來作為資料集，如Fig.5所示，其中紅色折線的部分為域處理使用到的資料。

由於其資料難以找尋，因此我欲將使用股票市場中的S&P500指數、台灣0050ETF資料。

我會比較兩個目標函數所造成的差異，將原先的目標函數替換成Sharpe Ratio，與先前的 $U_T = \sum_{i=1}^T R_t$ 做出比較，Sharpe Ratio是一般用來衡量投資績效與穩定性的重要指標，將風險的概念考慮進來，模型所得到的總收益也應當更為穩定。

另外論中提到將價差轉為fuzzy representation，對震盪格局的收益有所幫助，因此我也會針對這個結論做出驗證。

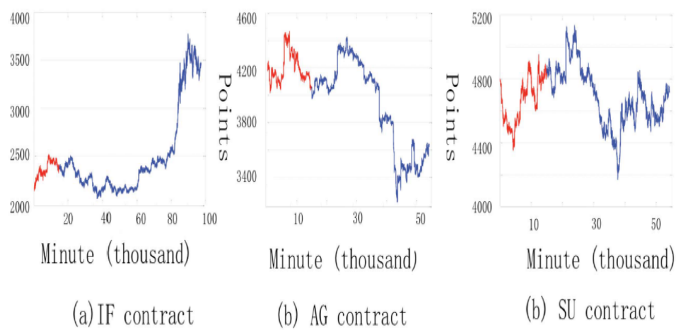


Fig. 5. 論文中使用的資料集

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.