

具認知能力的理財機器人知識處理器規劃

夏肇毅(Chao-Yih Hsia)

[CubicPower](http://CubicPower.com) 晶智能中心

chaoyihhsia@gmail.com

摘要

本文規劃一個知識處理器架構，用來建構能持續接收新資訊的知識型理財機器人。它會處理收到的數據，儲存資訊，由句子產生概念，再歸納成規則。然後存入理財機器人的規則庫中，讓理財機器人建立認知能力。之後根據內存的結構化知識做出理財決定。

Abstract

This paper presents a design to build a robo-advisor based on knowledge processing units (KPU). Each KPU attaches a sticker which keeps a piece of knowledge to be processed. The hierarchical arrangement of knowledge stickers allows human experiences to be executed to manage the portfolios of clients properly.

關鍵詞：理財機器人, Robo Advisors, 知識處理器, 認知, Cognition, 金融科技, Fintech.

1. 前言

機器人理財[3]是指可以用比一般理財顧問更低的價錢，來自動處理客戶投資組合的一個平台。它使用資產配置軟體，根據客戶的喜好，來決定如何將客戶的金錢投資到不同的財務工具上面。客戶的資訊是經過一個叫KYC(Know Your Customers)的程序來蒐集的。這是一個財富管理單位對一個新客戶所必須要經過的程序。它藉由一個線上的調查系統，來收集客戶對於財富管理的需求及他們的財務態度，尤其是了解他們對於風險的忍耐底限。

理財機器人根據投資回報率的平均值及標準差，來決定如何將客戶的金錢投資到不同的ETF上面。股票，債券以及現金的比例，決定了投資的報酬率及風險。一個理財機器人必須分散資產來投資，並且定期的再平衡這個投資比例。我們通常將資產管理的運算法則直接寫成程式，藉以建構一個理財機器人，這程式會直接在平台上運行一些經過事先挑選的規則。

然而我們在這裡，想規劃一個可以持續接收新資訊的知識型理財機器人。它會處理收到的數據，儲存資訊，然後根據內存的結構化知識做出決定。一般的人工智慧系統，如類神經網路等機器學習運算法，會使用數值模型，靠著調整權重來訓練機器。然而我們的理財機器人，是靠一個儲存著現有人類知識的知識網路，根據這些所學習到的知識來處理收到的資訊。

知識處理器[1]

我們可以假設，當眼睛和耳朵接受到外界的訊號時，會經過神經網路，立刻將它們傳送到所有的神經元上。應該會有一些比較底層的神經元來處理母音，子音和字母等。然後這些神經元連結到更高層，負責認識字和句子的神經元上。

神經元能接受訊號，而且必要的時候還會再重新產生訊號。觀念上，我們把這些腦神經元對應到一般的小貼紙，我們把這個稱為人腦貼紙模型。低層的貼紙保存著聲音，片語和句子的樣本。一層一層往上，上層的貼紙會保存著公式，理論，程序與方法。所有的貼紙都能平行運作著。

當一個貼紙偵測到有輸入訊號跟它的內容相似的時候，它就會送出信號給所有與它連結的貼紙。因此當我們聽到聲音時，會把它轉成文字，片語，然後變成句子。看到字母時，我們可以把它們一個一個連在一起，變成單字和句子，而且知道它的意思，這就是一個認知的過程。

為了要模擬人腦貼紙模型，首先我們需要有很多的知識處理器KPU(Knowledge Processing Unit)，用來執行在腦中貼紙上所紀錄的工作。一個貼紙就像一張小抄一樣，上面紀錄著一段知識，然後我們需要網路連到所有的知識處理器。讓每一個處理器會接受訊息，處理信息，然後再將訊息當成一個反應送出。這個反應將再觸發其它的處理器，讓整個流程持續執行。

在建立知識型理財機器人系統時，我們可以用一個電腦的MQ(Message Queue)系統來模擬概念上的神經網路訊號傳輸方式，同時每一個神經元再用一個知識處理單元KPU來模擬。一個知識處理單元包括一個MQ訂閱者輸入，跟一個MQ的發行者輸出。而輸出訊息可以由此被廣播到所有的知識處理單元KPU中。

透過從MQ訂閱一些特定主題訊息的方式，一個KPU能夠接收到這些它希望處理的特定訊息。同時它也有一張記錄著所要執行的運算法貼紙，處理後的結果將會經過MQ的輸出，被發行送到其它的KPU上。

知識處理網路

一個整理過的知識網路連結，會像一本書開頭的目次，把所有的主題與子題攤開。每個主題有內容的描述，如同wiki一樣，當它提到另一個主題時就有連結連過去。這是從上到下，目的是展現知識架構。

同時還要有反向連結，像一本書後的索引一樣，針對某個名詞列出哪幾頁有提到。如同Google一樣，把有提到某一主題的內容，都連結起來。這是由下往上，目的是要快速索引。

學習的過程，就是把書一頁頁的記入腦中，再依序產生目次。剛開始時還生疏，所以只能靠目次從頭順序唸到尾。熟練後就建立索引，只要聽到內容中的某一名詞，就能擴展到所有提到它的相關內容。

理想的理財機器人應能夠了解經濟新聞，同時針對市場變化提供建議。因此它必須要了解財經市場的原理與規則，待判斷世界的變化後再針對財經市場的變化做出反應。為了要造出一個架構於知識上的理財機器人，我們必須設立許多關鍵字的KPU來處理專業知識。像USD, JPY, OIL, 利率等。當一個KPU被標示為USD時，它就必須偵測USD的關鍵字輸入，然後發送出一切有關USD的資訊。同樣的，一個被標示為OIL的KPU必須偵測OIL的關鍵字輸入，同時也發送出與OIL有關的最新資訊。

因此，我們的理財機器人必須包括下列各個型態的KPU：

語言KPU：動詞，名詞，形容詞，副詞，片語，短句與文法。

公式KPU：財務公式如選擇權定價Black Scholes模型，凱利公式等。

方法KPU：如線性方程式解法。

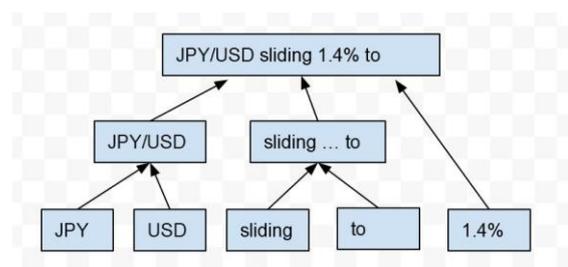
舉個例來說，我們來看一下應該要如何處理收到的外匯新聞：

”The safe-haven yen and Swiss franc were higher, with USD/JPY sliding 0.41% to 112.01, just off the previous session's two-month peak of 112.72, while USD/CHF fell 0.21% to trade at 0.9687. Elsewhere, EUR/USD gained 0.20% to trade at 1.1965, while GBP/USD edged down 0.10% to 1.3569, off Monday's 15-month highs of 1.3620.”

檢視一下這新聞的句子，我們發現這些關鍵字與短語：

- 數字，百分比
- USD/JPY, USD/CHF, EUR/USD, GBP/USD
- and, higher, sliding to, off peak of, fell to, gained to, edged down to, off highs of

因此，我們必須建造不同的KPU來處理數字，外匯商品符號以及它們變動的短語。第一層KPU處理字，第二層KPU處理短語，最上一層再處理句子的意義(圖一)。



圖一 財經知識貼紙階層結構[1]

當一個KPU偵測到句子中外匯商品的價錢時，它就應該將這價錢存入記憶體中。然後接著計算移動平均線，變異數，RSI, MACD, BBI等技術指標，同時它也計算與其它商品間的相關係數。這就是整個網路的運作型態。

建構理財知識

至於處理機器人理財策略的KPU，則要推測這商品的變動會影響哪些其它商品，以及影響力有多大。根據這些判斷，再做出買賣商品的決策。

同時，也應該要定義執行下列這些規則的KPU：

- 如果商品A上漲或下跌，則我們應該買或賣何種商品？數量多少？
- 各商品的收益平均值與變異數如何？
- 各商品的買賣點如何？
- 最佳槓桿比率是多少？

建構在KPU上的理財機器人，它的理財知識就是由許多理財元件所堆積起來[5]。每個元件就

是一個程式的函數，一個KPU。所謂聚沙成塔，最後堆起整個理財機器人。

最基本的理財知識就是金錢的時間價值計算，將現在的錢和未來的錢做轉換，也就是複利與折現。做這動作所用到的參數，有複利現值與複利終值兩個。所以所需要KPU包括有複利，折現，複利現值和複利終值。

如果開始有一連串的存提動作的話，那就變成了年金，會用到的參數有年金現值與年金終值。再加上長期投資報酬率計算，就還有IRR (Investment Return Rate)。需要KPU有：年金現值，年金終值，IRR。

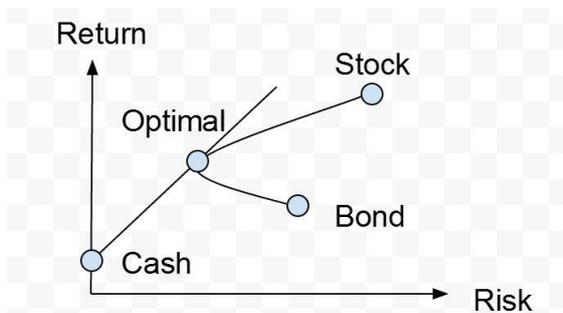
除了無風險的現金之外，我們還有股票與債券兩大類風險投資商品。所以要能對商品價值做出評價，因此需要的KPU包括：股利到股值計算，債券價格計算，殖利率計算。

有了現金，股票與債券三大類投資工具後，我們就要開始規劃如何做資產配置，來決定這三者間的分配比例，以期達成最大收益(圖二)。為了達成不同目標的配置[6]，我們需要提供下面各種資產配置的KPUs:

股票與債券資產配置 (Asset Allocation with Stocks and Bonds)

最小變異數資產配置 (Asset Allocation with Minimum Variance)

最佳風險資產配置 (Asset Allocation Optimal Risky Portfolio)



圖二：三種資產的報酬率與風險組合[1]

再加上對其他商品的避險運用，我們也必須提供各式期貨與選擇權估價KPU。

在資產配置比例確定後，若庫存現狀與設定不同，則須做再平衡動作，也就是執行買賣交易。但也不是說動就動，可能是在一定期限內等預設的買賣點到來才執行。尋找買賣點，就要靠Interday的技術分析和當日Intraday的算法交易Algo-Trading。前者決定何日買賣多少，後者決

定在當天每時段下多少量，以求得到最好的交易價。

一般技術分析是尋找商品的買賣點，可能是到達某個價位就買或是賣。一般有所謂的停損或停利。停損是跌到某個價位以下就立刻賣出，來停止損失。停利是漲破某個價位之後賣出，來停止獲利。所以這裡牽涉三個部分，第一是一條數值線，第二是向上穿越這個數值線時的動作，第三是向下穿越這個數值線時的動作。

一般股票最常見的數值線就是平均線，有年線，季線，月線和週線等。價錢在平均線之上時，會把這些平均線當成支撐。價錢在平均線之下時，會把這些平均線當成壓力。

不同的策略讓同樣一個穿越會有買跟賣兩種不同的選擇。所以一條數值線的向上穿越跟向下穿越兩種變化，乘上買跟賣兩種選項，總共就會有四種不同的信號產生：

- 向上穿越-買
- 向下穿越-買
- 向上穿越-賣
- 向下穿越-賣

一般的技術分析信號通常會有個箱型範圍，也就是會有上下兩個界限，所以就可能變成八種買賣信號。

針對上界限：

- 向上穿越上界限-買
- 向下穿越上界限-買
- 向上穿越上界限-賣
- 向下穿越上界限-賣

針對下界限：

- 向上穿越下界限-買
- 向下穿越下界限-買
- 向上穿越下界限-賣
- 向下穿越下界限-賣

所以有四種買進時機和四種賣出時機。交易就是一組買賣，可能是做多的先買後賣，或是做空的先賣後買。四種買進時機和四種賣出時機相乘，總共就有16種組合。一般簡單的交易規則設定就是從這16種中挑出一種來用，再配合著技術分析線如RSI，BBI等即可。

由此可知，理財機器人需要建立資產配置KPU，再平衡KPU，技術分析KPU和Algo-Trading算法交易KPU來執行整個機器人理財流程。

認知與內部概念建構

知覺(Perception)是外界刺激作用於感官時，人腦對外界的整體的看法和理解，它為我們對外界的感覺信息進行組織和解釋。

認知(Cognition)在心理學中是指通過形成概念、知覺、判斷或想像等心理活動來獲取知識的過程，即個體思維進行信息處理的心理功能。

莫查諾曾提出人腦的思考核心技巧[2]，包括：

- 聚焦技巧：定義問題，設定目標
- 資訊收集技巧：觀察，敘述問題
- 記憶技巧：編碼，回想
- 組織技巧：比較，分類，排序，呈現
- 分析技巧：指出屬性與成分，指出關係與模式，指出主題，指出錯誤
- 衍生技巧：推論，預測，闡述
- 整合技巧：總結，重組
- 評鑑技巧：建立標準，驗證

剛開始我們建立的理財機器人，都是執行hardcode的程式，也就是僅有記憶技巧。慢慢的我們若希望機器人能有學習能力的話，就必須建立資訊收集技巧，也就是了解句子的涵義。之後要有組織技巧和整合技巧，把一些句子歸納成規則，存入規則庫中。然後再運用分析技巧，將新收到的句子與規則庫中的規則比對，做出判斷。最後再運用評鑑技巧，來看看我們判斷的對不對。

要提供這些功能，首先我們要先定義出內部的概念描述方式。概念可能是一個或多個場景(scene)，一個場景是一張靜態照片 s_1 ，而動作就是連續的場景 $\{s_1, \dots, s_n\}$ 。比如說"台股收盤11000點"是一個場景，而"台股上漲100點，來到11000點"是一個動作，包括一前一後的兩個場景。之後再用短語到概念的對應函數(phrase to concept)：

$$p2c(p_i) = \{c_1, \dots, c_m\}$$

，來將短語轉為內部概念，這就是認知的過程。同時需要概念到短語的對應函數：

$$c2p(c_i) = \{p_1, \dots, p_n\}$$

，來將內部概念轉成短語以提供表達能力。

當收到一個句子之後，一般會做句義處理，之後產生概念。中文比英文麻煩一點，英文的字已經用空格斷開了，而中文第一步就是要斷詞，把句子斷成詞，然後才能了解詞意。至於句子意義的認知，則需要個別意義的判斷與紀錄，再存入概念函數中以建構場景。

當新的概念輸入後，就會和規則庫中的原有規則來比對，有不符合者，則新概念會被懷疑與反對。但若符合舊概念，就會覺得合理，了解，然後會加入規則庫中成為新規則。規則庫就

是資訊判斷法則的依據，要能預先輸入規則到理財機器人的規則庫中，或是輸入知識讓機器人學習後自動加入。提供一堆訊息讓它去學習理解，之後建立規則，然後再照學習到的步驟去做事，這樣就是讓理財機器人去建立認知能力。

好比當我們聽到"美元跌1%，黃金漲1%"時，我們就會把"美元跌，黃金漲"記到腦內規則庫裡。之後若再聽到"美元跌0.5%，黃金漲0.6%"時，我們比對規則庫，發現符合之前的規則，所以會覺得能理解。但當有一天聽到了"美元跌，黃金跌"時，立刻就覺得困惑，因為和腦內的規則不同，所以就會開始懷疑訊息不對。多聽幾次不同的訊息後，就會開始累積"美元跌，黃金漲"與"美元跌，黃金跌"的出現比率。之後的判斷就會由"一定"這樣變成"比較有可能"這樣，這就是認知理解後的學習成果。

有了對商品變化趨勢的推敲後，就會連接到操作規則上。看看所採行的策略是低買高賣的箱型區段操作，還是追高殺低的趨勢操作，之後就按照設定的規則去買賣。

知識型理財機器人的設計要點

歸納起來，整個知識型理財機器人的設計要點包括：

1. 建立結構化的知識貼紙

我們必須要組織上述的財經知識，來建立一個知識貼紙的組織。最低層的貼紙它保存著關鍵字，中間層貼紙保存片語，最上面一層的貼紙保存著句子與運算法。

2. 要能處理輸入訊息

進來的訊息將會被最底層的貼紙偵測到，然後最終觸發上層的運算法貼紙。

3. 設計週期執行工作

要週期性的執行如再平衡等工作。

4. 建立認知能力的規則庫

理解事實描述句，保存各式規則，當輸入事實描述句時與現有規則驗證。符合所規則時才記住，否則就存疑。

結論

我們在這裡提出了一個不同的理財機器人建構規劃。這個規劃的主要概念，是要在財機器人裏保存人類的知識結構。依循這個規劃，我們希望這個理財機械人在處理輸入訊息的時候，

同時也能夠學習更多的知識，存入判斷法則，之後也慢慢能將理財機器人變得更聰明一些。

參考文獻

- [1] Chao-Yih Hsia(夏肇毅), "Blueprint of Building Robo-Advisors with a Structured Knowledge Processing Network", NCS 2017全國計算機會議論文集
DOI:10.29428/9789860544169.201801.010,
<http://tanetsys.mcu.edu.tw/index.php/ncs2017/ncs2017/paper/viewFile/7379/4275>
- [2] Robert J. Marzano, "Dimensions of Thinking: A Framework for Curriculum and Instruction"
- [3] Investopedia, "Robo Advisor"
<http://www.investopedia.com/terms/r/robo-advisor-roboadviser.asp>
- [4] 夏肇毅, "夏肇毅觀點：複製小抄大腦，讓金融科技搭起人工智慧塔", 2017 風傳媒
<http://www.storm.mg/article/289062>
- [5] Chao-Yih Hsia, "CubicPower Money Management".
<http://www.cubicpower.idv.tw>
- [6] Zvi Bodie ,Alex Kane and Alan Marcus, "Essentials of Investments"