**AI-FML代理人補丁學習機制應用於圍棋機器人**

|  |  |
| --- | --- |
| 蔡易霖  國立臺南大學  資訊工程學系  m10759011@gm2.nutn.edu.tw | 李健興 國立臺南大學  資訊工程學系  leecs@mail.nutn.edu.tw |

# 摘要

在本研究中，我們提出了一個具有有補丁學習(PL) 機制的AI-FML代理人應用於圍棋機器人。AI-FML代理人具有三種智慧應用於機器人，包括感知智能、認知智能和計算智能。此外，我們將補丁學習機制嵌入到AI-FML代理人中。執行補丁學習的方法包括三個步驟。首先訓練初始全域模型，接著為每個已識別的模型進行補丁訓練，最後使用剩下的訓練數據更新全域模型。本篇論文採用Google DeepMind Master 60盤圍棋做為訓練集和測試集。從實驗結果可以發現具有補丁學習機制的AI-FML代理人可以提高圍棋機器人在回歸模型上的性能。

**關鍵詞**：AI-FML、代理人、補丁學習、模糊機器學習、圍棋。

## 前言

如今，機器學習已在現實生活中的許多領域和行業中被使用，例如醫學診斷、圖像處理、回歸……等。但是訓練高效能的機器學習模型通常是一個迭代的過程，它依賴於經驗和嘗試錯誤。有時當模型表現不令人滿意時，我們需要採取一些補救措施來增強其性能[1]。我們可以使用一些策略來增強機器學習的性能，例如使用單個多神經元的模型、單個多層網路的模型、並行連接多個簡單基底模型或串聯連多個簡單的基底模型[1、2]。 在本文中，我們提出了一種具有補丁學習（PL）機制的AI-FML代理，以提高機器學習的性能[1、2]。Mendel所提出的補丁學習想法如下：「設想一位雕塑家正在雕刻一個人物，在他第一次琢磨後，他會檢查整個雕像，並且注意需要對哪些地方進行改進。他將重點擺到需要被改善的部分，然後將雕像的精緻部分與雕像的其餘部分結合在一起。他會繼續不斷的修補，直到對整個雕像感到滿意為止。」補丁學習中的每個補丁都類似於雕塑品中需要更多補修的部分[1]。根據[1]，補丁學習包括三個步驟，描述如下：（1）首先使用所有訓練數據訓練初始全域模型。（2）從初始的全域模型中找出補丁，並為每個補丁訓練一個補丁模型。（3）使用訓練集更新全域模型。Wu和Jerry [1]將補丁定義為輸入域中的連接多面體。例如，一維輸入域中是一個直線區段，而二維輸入域可以是一個矩形，一個橢圓形等。但是一般來說，尋找補丁的位置是一項非常艱鉅的任務，並非每個問題都可以輕鬆實現。補丁學習並行和串聯連接多個簡單模型，以提高學習性能。Mendel在論文中[2]介紹如何使用基於規則的模糊系統執行補丁學習，因為在模糊系統中易於初始化補丁的位置。

對於數據科學家來說，回歸、集成式學習和深度學習是重要的機器學習方法[9]。Jang [12]在1993年提出了一種基於自適應網絡的模糊推論系統（ANFIS）。ANFIS整合了機器學習模型和模糊邏輯原理，並且在實際應用中取得兩者的好處。它的推論系統對應於一組模糊規則，這些規則具有學習能力以近似非線性函數，並被認為是通用評估器[12]。在我們之前的論文[3]中，我們使用AlphaGo Master 60盤棋局作為實驗數據集，並基於FML基因學習（GFML）、XGBoost集成式學習和七層深度模糊神經網絡（DFNN）訓練預測勝率，以使Darkforest AI 機器人預測的獲勝率更接近ELF Open Go AI機器人。在本文中，我們進一步將FML、補丁學習和基於自適應網絡的模糊推理系統（ANFIS）與深度學習相結合，以訓練模型來預測AlphaGo Master六十場比賽的勝率。實驗結果顯示，引入補丁學習機制比我們在以前的論文中所提出的方法具有更好的性能[3]。

本論文其餘章節如下：第二章節介紹了基於ANFIS的補丁學習機制的結構。第三章節介紹了圍棋資料集基於ANFIS的補丁學習機制。第四章節提出了一種基於補丁學習的AI-FML代理人，用於機器人應用，以及預測圍棋比賽的勝率。 第五章節為實驗結果。 最後第六章節結論以及討論。

### 基於ANFIS補丁學習機制架構

* 1. **補丁學習機制**

在傳統ANFIS的設計階段，使用訓練資料來優化輸入的隸屬函數與輸入參數，並使用所有訓練資料來優化評估指標[1、12]。基於文獻[1]中補丁學習的概念，本論文提出了一種具有補丁學習機制的AI-FML代理人，從全域的模糊規則系統設計開始，然後找到評估指標最大的區域作為補丁。圖1顯示了用於預測Master 60 盤棋局勝率的補丁學習架構，其描述如下：（1）我們使用所有訓練資料來訓練全域模型。 （2） 我們確定第一手（M1）到P1手（MP1）的輸入區域會引起較大的損失，因此我們使用屬於該區域的訓練資料來訓練一個補丁模型（PM1）以減少整體學習錯誤。（3）最後再使用補丁1、2，… 到L未使用的剩餘訓練資料更新全域的模糊規則系統。​​圖1以第二盤棋局為例，假設L = 3，MP1 = 20，MP2 = 40和MP3 = 90。



**圖1. 預測Master 60 盤棋局勝率的補丁學習架構**

* 1. **基於自適應網絡的模糊推理系統機制**

基於ANFIS的補丁學習機制已嵌入到AI-FML代理人中，我們將其應用於預測圍棋資料的勝率。另外採用ANFIS算法訓練補丁模型以提高性能指標[1，12]。 ANFIS是基於Takagi–Sugeno模糊推理系統的人工神經網絡模型之一，並於1990年初被提出[12，13]。在本論文中，我們使用PyTorch來實作基於ANFIS的補丁學習模型框架。ANFIS的架構包括前鑑部和後鑑部兩部分，其架構由五層網路所組成[12]，描述如下：（1）第一層/模糊化層：我們使用隸屬函數來計算輸入值所對應的歸屬值。他們。 （2）第二層/規則層：負責通過乘積為輸入信號來生成規則的觸發強度。（3）第三層/正規化層：正規化觸發強度，將每個值除以所有的觸發強度總和。（4）第四層/推論層：將第三層的正規化值與後鑑部參數相乘，並將結果發送到第五層。 （5）第五層/解模糊化層：將所有輸出信號的總和計算，並生成最終輸出。

1. **圍棋資料基於ANFIS補丁學習機制**
   1. **基於自適應網絡的模糊推理系統機制**

在本論文中，我們採用2016年12月和1月的AlphaGo Master 60盤棋局作為實驗數據，這些資料被指定為在IEEE CIS旗艦會議上舉行的比賽數據[3]。參賽者可以從60場比賽中選擇40場比賽作為訓練數據，將其餘20場比賽作為測試數據。並建構模糊推論系統的知識庫和規則庫。表1為第一盤棋局的第一步到第二十步的資料。每個棋局都包含Darkforest AI Bot和EFL OpenGo AI Bot預測的數據。MoveNo是手數編號，但MoveNo僅列出「奇數」編號（即1、3、5 ...），因為每一行對應一對黑棋與白棋。因此MoveNo 1的行對應於黑色的第一手（即，B1）和白色的第一手（即，W2）。MoveNo 145對應於黑色第145手（即B145）和白色第146手（即W146）。如果最後的MoveNo為「奇數」，則最後一行的白色資訊將為空。訓練資料從第一局到第四十局共有3758筆數據。測試資料從第四十一局到第六十局共有1880筆數據。

**表1. 第一局第一步到第二十步圍棋數據**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Move  No | DBSN | DWSN | DBWR | DWWR | DBTMR | DWTMR | EBWR | EWWR |
| 1 | 3863 | 2274 | 0.52 | 0.48 | 0 | 1 | 0.49 | 0.50 |
| 3 | 9283 | 7866 | 0.51 | 0.48 | 0.5 | 1 | 0.45 | 0.54 |
| 5 | 11395 | 6798 | 0.51 | 0.47 | 0.66 | 1 | 0.45 | 0.57 |
| 7 | 4499 | 10703 | 0.51 | 0.46 | 0.75 | 1 | 0.49 | 0.52 |
| 9 | 7388 | 20017 | 0.52 | 0.46 | 0.8 | 1 | 0.47 | 0.53 |
| 11 | 20098 | 9693 | 0.53 | 0.46 | 0.83 | 1 | 0.47 | 0.52 |
| 13 | 20017 | 14595 | 0.53 | 0.47 | 0.85 | 1 | 0.48 | 0.49 |
| 15 | 6786 | 4892 | 0.52 | 0.45 | 0.87 | 0.875 | 0.48 | 0.53 |
| 17 | 20017 | 6432 | 0.53 | 0.46 | 0.88 | 0.7778 | 0.48 | 0.53 |
| 19 | 5267 | 13267 | 0.51 | 0.48 | 0.9 | 0.8 | 0.48 | 0.50 |

* 1. **圍棋資料前處理**

Jang [12]在1993年論文中提到：「模糊條件語句是IF A THEN B形式的表達式，其中A和B是具有適當的隸屬函數的模糊集的標籤」和「fuzzy if-then規則通常用於記錄了不精確的推論模式，這些模式對人類在不確定和不精確的環境中做出決策扮演著重要的角色」。

有多種用於劃分輸入域的分區函數，例如，明確分區函數或Type-1分區函數[2]。實值變數的明確分區由多個不重疊的相鄰區域所組成，這些區域是實數的間隔，其中每個區域中的隸屬程度為1，在該區域之外為0[2]。此外實值變數的一階不確定性分區由多個重疊區間組成，其中我們可以確定重疊的開始和結束位置，因此每個重疊中的歸屬程度是介於0到1之間。一階不確定性分區被視為與其最近鄰域重疊的非矩形型Type-1分區函數。如圖2本研究採用64個高斯函數的Type-1分區，用於圍棋數據預處理。



**圖2. 64個高斯函數的Type-1分區函數**

此外實驗中有兩個用於資料預處理的標準。 第一個標準是「如果Darkforest的最終預測結果是錯誤的，則該局的資料集將不會用於訓練集或測試集。」 例如我們在訓練集中將不採用第15盤、第25盤和第29盤的資料集。 圖6顯示了這三盤Darkforest與ElF OpenGo的預測結果。

第二個標準是「如果ELF OpenGo和Darkforest之間的白色或黑色預測勝負差異超過40％，則該局的資料將不會在訓練集或測試集中採用。」 假設一盤對局中有100手，但ELF OpenGo和Darkforest預測有40個手數不同，則該局將從資料集中刪除。基於此標準，我們將第8盤、第32盤和第36盤從訓練數據集中刪除。在實驗測試集中我們刪除了第44盤、第47盤、第56盤和第57盤。 此外，我們還檢查資料集中的缺失值。如果有一局的第一手與最後一手都是相同的黑棋或白棋，我們會將該局的最後一筆資料移除。經過檢查後，我們從訓練集中刪除了19筆資料，從測試集中刪除了6筆資料。最後我們在此實驗中保留了3143筆訓練資料，測試集中保留1239筆資料。

|  |
| --- |
|  |
| (a) |
|  |
| (b) |
|  |
| (c) |

**圖3. Darkforest與ELF OpenGo預測結果：（a）第15盤（b）第25盤（c）第29盤**

### 基於補丁學習AI-FML機器人圍棋應用

* 1. **ANFIS於AI-FML代理人圍棋應用**

基於補丁學習的AI-FML代理人中的模糊規則核心系統是ANFIS，它由以下功能所組成：（1）含有數據的知識庫、（2）規則庫、（3）規則推論的運算子、（4）模糊化過程，以及（5）去模糊化過程[12]。如圖4顯示了機器人應用於AI-FML代理人的ANFIS架構，該架構透過以下步驟建立：（1）在前鍵部使用隸屬函數檢索模糊變數的輸入值，以計算每個語意的隸屬程度。（2）結合前鍵部的隸屬程度以獲取每個規則的權重。（3）根據權重產生每個規則的後鍵部。（4）匯總結果以產生最終輸出[ 12]。人類輸入資料，AI-FML代理人將解模糊後的值輸出給機器人，例如Kebbi Air，Palro或Zenbo，以實現人類和機器人在教室中學習的目標[3]。表2是ANFIS模型使用反向傳播學習的演算法。



**圖4. ANFIS於AI-FML代理人架構**

在本研究中，我們使用64個Type-1高斯分區配分函數對六個輸入模糊變數進行分區，包括DBSN、DWSN、DBWR、DWWR、DBTMR和DWTMR。舉例來說，如果x是AI-FML代理人補丁學習的輸入值DBSN，有64條規則，如下所示：

規則1：IF *x*為分區1，Then *y* = *y1*(*x*)

規則2：IF *x*為分區2，Then *y* = *y2*(*x*)

⋮

規則64：IF *x*為分區64，Then *y* = *y*64(*x*)

其中*y*1(*x*)，*y*2(*x*)，…，*y*64(*x*)是*x*的不同函數。在分區 P(1|*x*) 中，假設只有規則1被觸發，因此模糊系統輸出為y = y1（x）。另外在分區 P(2|*x*) 中，如果規則1和規則2均被觸發，因此模糊系統的輸出是 *y*1(*x*) 和 *y*2(*x*) 在分區1和2的隸屬程度加權平均值。如圖5所示，在初始狀態下圍棋具有64個高斯函數的Type-1分區。

1. ANFIS model using backpropagation algorithm.

|  |
| --- |
| (a) |
| **Input:**   1. *TDS*: Training data set with *NTDS* examples. 2. **x** = [*x*1, *x*2, …, *x*n] where *x*1, *x*2, …, and *x*n denote *BSN*, *WSN*, *BWR*, *WWR*, *BTMR*, and *WTMR*, respectively.   **Output:**  = [*BWR*1, *BWR*2, …, *BWRNTDS*]: Inferred win rate of Black  **Method:**  **Step 1:** Initialize ANFIS neural network  **Step 1.1:** Randomize a constant formean () and variance () of Gaussian membership functions of each input feature in **x** usinganormal distribution.  **Step 1.2:** Randomize a constant for the output *fi* of the *i*th fuzzy rule using normal distribution.  where *i* = 1, …, and *NFR*, *n* = 1, …, and *NINF*, *NINF* is number of input features, and *NFR* is number of fuzzy rules.  **Step 2:** Input the first example of the training data set  **Step 2.1:** For each input feature *x*1, *x*2, …, *x*n in **x**  **Step 2.1.1:** Calculate the membership degree of each fuzzy rule by  where *i*=1, 2, …, *NFR* and represents the *i*th Gaussian fuzzy set of the *n*th input feature *xn* with mean and variance .  **Step 2.1.2:** Perform multivariable T-norm operator to calculate the firing strength of each fuzzy rule by  =  **Step 2.1.3:** Normalize the firing strengths for each fuzzy rule  **Step 2.1.4:** Calculate the outputfor each fuzzy rule by  *fi*  **Step 2.1.5:** Compute the overall output *BWR*1 by  **Step 3:** Repeat **Step** 2 for the remaining examples 2, 3, .., and *NTDS* in *TDS* to acquire *BWR*2, *BWR*3, …, and *BWRNTDS*, respectively.  **Step 4:** End |
| (b) |
| **Input:**   1. **BWRDO =** [*BWR*1, *BWR*2, …, *BWRNTDS*]: Label value of the win rate of Black /\**WWRDO* = 1-*BWRDO*\*/ 2. = [*BWR*1, *BWR*2, …, *BWRNTDS*]: Inferred win rate of Black   **Output:**  *Learned Model*  **Method:**  **Step 1:** For *k1* to epochs  **Step 1.1:** Calculate loss value of the *k*th epoch by  *Lossk* =  **Step 1.2:** Implement the backpropagation algorithm with an Adam optimization to update the parameters of our learned model, including , , and *fi* listed in Table III (a).  **Step 2:** Store the learned model  **Step 3:** End |



**圖5. 圍棋基於補丁學習AI-FML代理人的初始高斯函數**

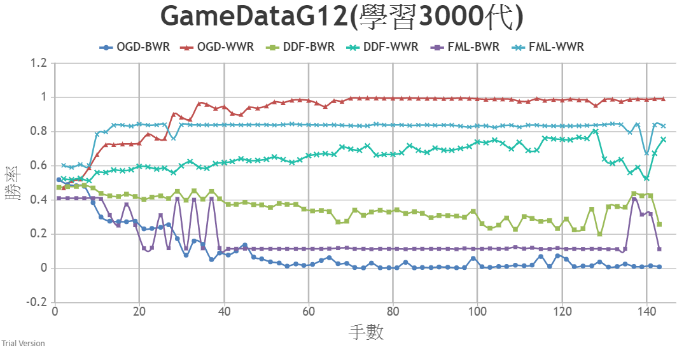
* 1. **圍棋資料集基於補丁學習AI-FML代理人**

基於補丁學習的AI-FML代理人可以將1-Patch、2-Patch和3-Patch應用於電腦圍棋資料集。圖13(a)是1-Patch的學習模型，此模型僅有一個Patch專門來預測第一手到第四十手。其餘的手數交由Global Model進行預測。圖五(b)是2-Patch的學習模型，此模型經由兩個Patch進行訓練。在此模型中我們將前四十手切成兩等份，分別為第一手到第二十手，以及第二十一手到第四十手。其餘的手數交由Global Model進行預測。圖五(c)是3-Patch的學習模型，Patch1與2分別為第一手到第二十手，以及第二十一手到第四十手。Patch3為第四十一手到第九十手。其餘的手數交由Global Model進行預測。

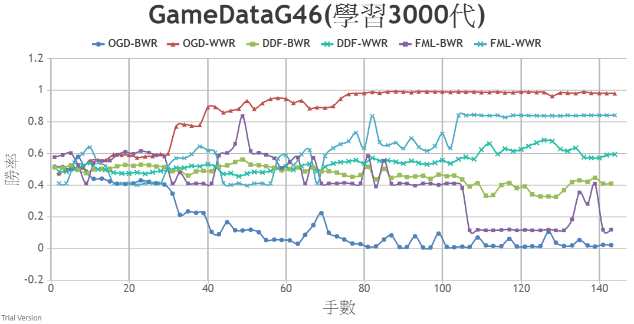
|  |
| --- |
| (a) |
| (b) |
| (c) |

### 實驗結果

實驗分為兩部分測試。第一部分為內部測試，我們使用45盤棋局下去做訓練並從中隨機挑選一盤做為內部測試，圖9是使用第12盤的棋局資料下去做FML推論的結果，OGD-BWR為ELF OpenGo所預測的黑棋勝率;OGD-WWR為ELF OpenGo所預測的白棋勝率;DDF-BWR為Darkforest所預測的黑棋勝率; DDF-WWR為Darkforest所預測的白棋勝率; FML-BWR為經過機器學習後利用FML所推論出來的預測的黑棋勝率; FML-WWR為經過機器學習後利用FML所推論出來的預測的白棋勝率。從內部測試的12盤棋局可以發現勝率確實有稍微逼近ELF OpenGo的預測，但勝率介於0.3~0.7之間的學習成效不太理想。我們猜測可能是這一區間的規則庫應該要再分更細才能做出更準確的決策。圖10為外部測試第46盤棋局，從此圖我們可以發現學習後的曲線約在第70手開始有明顯勝率的變化，而原本Darkforest所預測的勝率是在108手才有稍微小幅度的勝負區分。

經過機器學習後我們發現3000代的基因學習的效果最佳，超過5000代的學習發現會有過度擬合的問題。  
  


**圖9. Game12 學習3000代內部測試結果**



**圖10. Game46學習3000代外部測試結果**

### 結論及未來展望

目前 ELF OpenGo的棋力比業餘九段還高，也就是說人類已經很難贏了，因此人類不得不放下身段學習電腦 AI 的思維，不過我們可以從對局中發現當局勢緊張時電腦會以很高段的手法來指導人類下棋，相對的當人類無法理解這麼高段的下法時就會產生電腦與人類的思維衝突，從教育的觀點來看，真正好的學習是要循序漸進依據棋士的棋力來推薦每一手，而不是高壓式的學習， 因此目前的研究我們是必要詳細取得職業手每手的資訊，我們要知道當遇到衝突時，每位棋士要經過多少手後才會了解原來當時這樣下是不對的，這也正是教育中的學習認知的觀念。棋力越高的如ELF OpenGo他在前五十手內只有有一步下錯勝率就會懸崖式的下降，相對的Darkforest是要到後面一點可能二十手後才被計算出情勢不利，我想黑森林系統面臨的狀況跟人類是差不多的等級。

從這次的實驗結果中我們可以發現機器學習的演化次數並非越多越好，因此我們要不斷的測試與調整基因演算法的參數和模糊規則的設計及模糊區間的調整。在下次的競賽中我們會嘗試著調整模糊區間的數量並增加模糊規則來判斷看看是否會改善目前現在的實驗結果。

**致謝**

感謝科技部數位經濟技術創新研發與應用推動計畫: 智慧IRT機器人與人類共同學習於教育學習應用及國立臺南大學知識應用暨網路服務研究中心團隊提供模糊推論工具進行實作分析，以及感謝感謝西班牙研究團隊(Jose Manuel Soto Hidalgo, Giovanni Acampora, Jesus Alcala Fernandez, Jose Alonso Moral)提供JFML讓我可以了解Mamdani方法和整個計算流程，才能將全部的推論過程詳細的應用在電腦對局應用上。最後還要感謝國立臺南大學李健興團隊舉辦工作坊讓我們有資源管道學習模糊推論系統及智慧決策工具。

**參考文獻**

1. Deepmind, “AlphaGo Master series: 60 online games,” Jan. 2019. [Online] Available: https://deepmind.com/research/alphago/match-archive/master/.D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. v. d. Driessche, T. Graepel, and D. Hassabis, “Mastering the game of Go without human knowledge,” Nature, vol. 550, pp. 35–359, 2017.
2. A Mamdani Type Fuzzy Logic Controller. Ion Iancu University of Craiova, Romania. Available: http://cdn.intechopen.com/pdfs/34221/InTech-A\_mamdani\_type\_fuzzy\_logic\_controller.pdf
3. Lee, C. C., “Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Part I and Part II”, IEEE Transactions on Systems, Man and Cybernetics, March-April 1990.   
   Available:https://ieeexplore.ieee.org/document/52551, https://ieeexplore.ieee.org/document/52552
4. C. S. Lee, M. H. Wang, L. C. Chen, Y. Nojima, T. X. Huang, J. Woo, N. Kubota, E. Sato-Shimokawara, T. Yamaguchi, “A GFML-based robot agent for human and machine cooperative learning on game of Go,” 2019 IEEE Congress on Evolutionary Computation (IEEE CEC 2019), Wellington, New Zealand, Jun. 10-13, 2019. (Submitted)Available:https://arxiv.org/abs/1901.07191