

# 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補丁學習機制的 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資料更新全域模型。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。

## 1. 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、影像處理、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機制的 AI-FML 代理，以提高機器學習的性能。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，在他第一次雕刻後，他會檢查整個雕像，並且注意需要對哪些地方進行改進。他會將重點擺到需要被改善的部分，接著將雕像的精緻部分與雕像的其餘部分結合在一起。持續不斷的修補，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，或是一個橢圓形...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 [11]在 1993 年提出了一種基於自適應網路的模糊推論系統 (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，並被認為是一種通用的評估器[11]。在我們之前的論文[3]中，我們使用 AlphaGo Master [14] 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路 (DFNN) 訓練預測勝率，以使 Darkforest AI 機器人 [12]預測的勝率更接近 ELF Open Go AI 機器人 [13]。在本文中，我們進一步將 FML、補丁學習和基於自適應網路的模糊推理系統 (ANFIS)與深度學習相結合，以訓練模型來預測 AlphaGo Master 60 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]。

本論文架構如下：第 2 節介紹了基於 ANFIS 的補丁學習機制的結構。第 3 節介紹了圍棋資料集基於 ANFIS 的補丁學習機制。第 4 節提出了一種基於補丁學習的 AI-FML 代理人，應用於圍棋，以及預測圍棋比賽的勝率。第 5 節為實驗結果。最後，第 6 節為本論文之結論。

## 2. 基於 ANFIS 補丁學習機制架構

### 2.1 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、11]。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)我們找出第 1 手( $M_1$ )到  $P_1$  手( $M_{P_1}$ )的輸入區域會引起較大的損失，因此我們使用屬於該區間的訓練資料來訓練一個補丁模型( $PM_1$ )以減少整體學習損失。接著我們採用第  $P_1+1$  ( $M_{P_1+1}$ )手到第  $P_2$  ( $M_{P_2}$ )手，...,第  $N$  ( $M_N$ )手到第  $P_N$  ( $M_{P_N}$ )手落於區間內的資料來訓練補丁模型 1 ( $PM_1$ )，補丁模型 2

( $PM_2$ ), ..., 補丁模型  $L$  ( $PM_L$ ) (3)最後再使用補丁 1, 2, ...,  $L$  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 2 盤棋局為例，假設  $L=3$ ,  $MP_1=20$ ,  $MP_2=40$  和  $MP_3=90$ 。

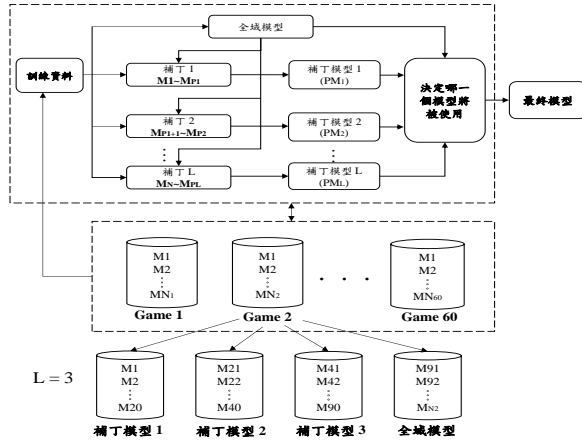


圖 1. 預測 Master 60 盤棋局勝率的補丁學習架構

## 2.2 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, 11]。ANFIS 是基於 Takagi-Sugeno 模糊推論系統的人工神經網絡模型之一，並於 1990 年初被提出[10, 11]。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，如圖 2[11]。其描述如下：(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。(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。

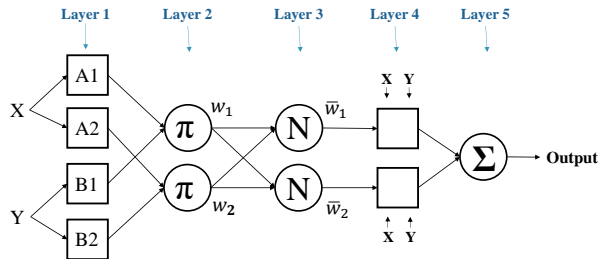


圖 2. 自適應網絡的模糊推理系統五層架構[11]

## 3. 圍棋資料基於 ANFIS 補丁學習機制

### 3.1 IEEE WCCI 2020 資料集簡介

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 IEEE CIS 旗艦會議上所舉行的競賽數據[3]。參賽者可以從 60 場比賽中選擇 40 場比賽作為訓練資料，將其餘 20 場比賽作為測試資料。並建構模糊推論系統的知識庫和規則庫。表 1 為第 1 盤棋局的第 1 步到第 20 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 1 手(即 B1)和白色的第 1 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 1 局到第 40 局共有 3758 筆數據。測試資料從第 41 局到第 60 局共有 1880 筆數據。

表 1. 第 1 局第 1 步到第 20 步圍棋數據

Move No	DBS N	DWS N	DBW R	DWW R	DBTM R	DWTM R	EBW R	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

說明

- (1) DBSN 和 DWSN 表示黑森林系統黑與白的模擬次數
- (2) DBWR 和 DWR 表示黑森林系統黑與白的預測勝率
- (3) DBTMR 和 DWTMR 表示黑森林系統黑與白的匹配率
- (4) EBWR 和 EWR 表示 ELF OpenGo 系統黑與白的預測勝率

### 3.2 圍棋資料分區前處理

Jang [11]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 3

本實驗採用 64 個高斯函數的 Type-1 分區，用於圍棋數據預處理。

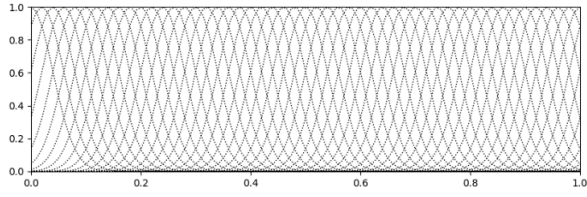


圖 3. 64 個高斯函數的 Type-1 分區函數

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 4 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 1 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。

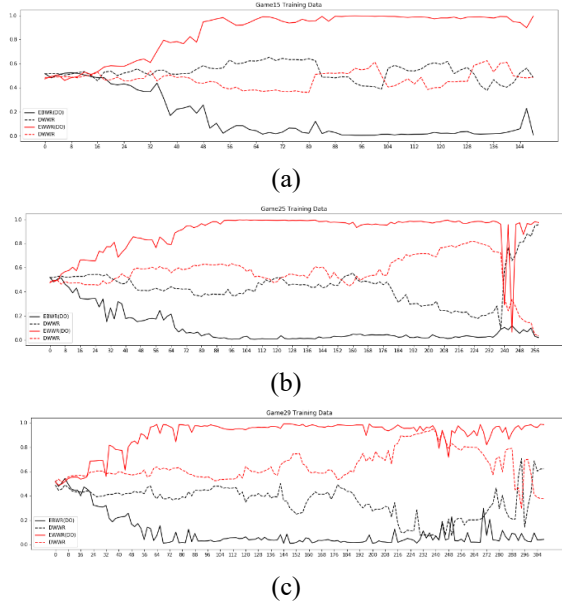


圖 4. Darkforest 與 ELF OpenGo 預測結果 (a)第 15 盤 (b)第 25 盤 (c)第 29 盤

## 4. 基於補丁學習 AI-FML 機器人圍棋應用

### 4.1 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。(5)去模糊化過程[11]。圖 5 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)在前鑑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)結合前鑑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鑑部。(4)匯總結果以產生最終輸出[11]。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。

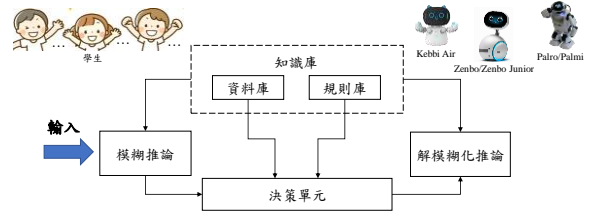


圖 5. ANFIS 於 AI-FML 代理人架構

表 2. 使用反向傳播學習的演算法的 ANFIS 模型

(a)	
<b>Input:</b>	
1) TDS: 帶有 N 個 TDS 數據的訓練集。	
2) $\mathbf{x} = [x_1, x_2, \dots, x_n]$ 其中 $x_1, x_2, \dots, x_n$ 表示 BSN、WSN、BWR、WWR、BTMR、WTMR。	
<b>Output:</b>	
$BWR = [BWR_1, BWR_2, \dots, BWR_{NTDS}]$ : 推論黑色的勝率	
<b>Method:</b>	
<b>Step 1:</b> 初始化 ANFIS 神經網路	
<b>Step 1.1:</b> 使用常態分佈隨機產生 $\mathbf{x}$ 中每個輸入的高斯隸屬函數的平均值 $\text{mean}(c_n^i)$ 和變異數 $\text{variance}(\sigma_n^i)$ 的常數。當 $i = 1, \dots, NFR$ 和 $n = 1, \dots, NINF$ 。NINF 即為輸入特徵數量，NFR 即為模糊規則數量。	
<b>Step 2:</b> 輸入訓練集的第一個數據	
<b>Step 2.1:</b> $\mathbf{x}$ 對於每個輸入特徵表示 $x_1, x_2, \dots, x_n$	
<b>Step 2.1.1:</b> 計算每個模糊規則的隸屬程度	
$\mu_{A_n^i}(x_n) = \exp\left(-\frac{(x_n - c_n^i)^2}{2(\sigma_n^i)^2}\right)$	
當 $i = 1, 2, \dots, NFR$ 。 $A_n^i$ 代表第 $n$ 個輸入特徵 $x_n$ 的第 $i$ 個高斯模糊集合平均值 $c_n^i$ 和變異數 $\sigma_n^i$ 。	
<b>Step 2.1.2:</b> 執行多個 T-norm 運算子計算每個模糊規則的觸發強度	
$\omega_{A^i}(\mathbf{x}) = \prod (\mu_{A_1^i}(x_1), \mu_{A_2^i}(x_2), \dots, \mu_{A_n^i}(x_n))$	
<b>Step 2.1.3:</b> 正規化每個模糊規則的觸發強度	
$\bar{\omega}_i = \frac{\omega_i}{\sum_{i=1}^{NFR} \omega_i}$	
<b>Step 2.1.4:</b> 計算每個模糊規則的輸出	
$BWR_i = \bar{\omega}_i \times f_i$	
<b>Step 2.1.5:</b> 計算所有 $BWR_i$ 的輸出	
$BWR_1 = \frac{\sum_{i=1}^{NFR} BWR_i}{\sum_{i=1}^{NFR} \omega_i}$	
<b>Step 3:</b> 重複 Step 2 執行剩下 TDS 中的所有數據 2, 3, ..., NTDS 並取得 $BWR_2, BWR_3, \dots, BWR_{NTDS}$ 。	
<b>Step 4:</b> 結束	

(b)	
<b>Input:</b>	1) $BWRDO = [BWR_1, BWR_2, \dots, BWR_{NTDS}]$ : 所有黑色勝率的標籤 $/*WWRDO = 1 - BWRDO*/$ 2) $BWR = [BWR_1, BWR_2, \dots, BWR_{NTDS}]$ : 推論黑色的勝率
<b>Output:</b>	學習的模型
<b>Method:</b>	
<b>Step 1:</b>	For $k \leftarrow 1$ to epochs
<b>Step 1.1:</b>	計算第 $k$ 個 epoch 的損失值 $Loss_k = \sum_{i=1}^{NTDS} (BWR_i - BWRDO_i)^2 / NTDS$
<b>Step 1.2:</b>	使用 Adam 優化器實現反向傳播算法，以更新我們的學習模型參數，包含表二(a)中所列出的參數 $c_n^l$ 、 $\sigma_n^l$ 和 $f_i$ 。
<b>Step 2:</b>	儲存學習後的模型
<b>Step 3:</b>	結束

在本研究中，我們使用 64 個 Type-1 高斯分區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  $y = y_1(x)$

規則 2：IF  $x$  為分區 2，Then  $y = y_2(x)$

⋮

規則 64：IF  $x$  為分區 64，Then  $y = y_{64}(x)$

其中  $y_1(x), y_2(x), \dots, y_{64}(x)$  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  $y = y_1(x)$ 。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 6 所示，在初始狀態下圍棋具有 64 個高斯函數的 Type-1 分區。

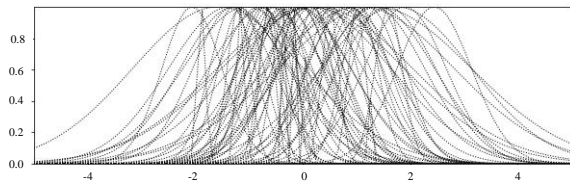


圖 6. 初始化高斯函數於圍棋補丁學習 AI-FML 代理人

## 4.2 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 7(a) 是 1-Patch 的學習模型，此模型僅有一個 Patch 專門來預測第 1 手到第 40 手。其餘的手數交由全域模型進行預測。圖 7(b) 是 2-Patch 的學習模型，此模型經由兩個補丁進行訓練。在此模型中我們將前 40 手切成兩等份，分別為第 1 手到第 20 手，以及第 21 手到第 40 手。其餘的手數交由全域模型進行預測。圖 7(c) 是 3-Patch 的學習模型，補丁 1 與 2 分

別為第 1 手到第 20 手，以及第 21 手到第 40 手。補丁 3 為第 41 手到第 90 手。其餘的手數交由全域模型進行預測。

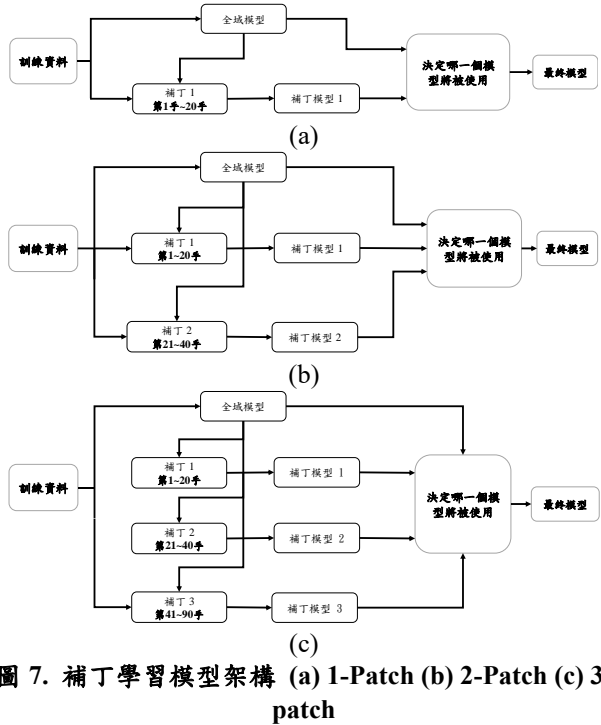


圖 7. 補丁學習模型架構 (a) 1-Patch (b) 2-Patch (c) 3-patch

## 5. 實驗結果

### 5.1 實驗 1: AI-FML 代理人 ANFIS 全域學習

在實驗 1 中，我們採用具有 Adam 優化器的 ANFIS 和學習速率為 0.001 的梯度下降法來更新神經網絡中的所有參數。表 2 說明了在訓練 2000 個迭代後使用平均絕對誤差(MAE)、均方誤差(MSE)和均方根誤差(RMSE)的評估指標數值。這種方法的性能比我們先前論文[3]中使用 Adam 優化器的 DNN 機器學習的方法還來的更好。此外實驗 1 減少了學習參數量和學習時間。圖 8 為學習 2000 個迭代後所有訓練集基於每一手的 MSE 的平均損失值。從圖中我們可以發現前 170 手的損失值比其他手有更大的方差。圖 9 顯示了經過 2000 個迭代學習後的 DBSN 高斯模糊集合。圖 10 顯示了實驗 1 中第 39 局和第 58 局的回歸分析。虛線標記的曲線為 EBWR(AL)/ EWWR(AL) 是實驗 1 的實驗結果。圖中的虛線和實線分別是 Darkforest 和 ELF OpenGo 的預測勝率，黑色和紅色曲線分別代表黑棋和白棋。如圖 10(a)所示，虛線曲線雖然有趨近於 ELF OpenGo 所預測的勝率，但前 5 手和第 56 附近的手數仍具有較大的方差。從圖 10(b)可以發現，除了前 40 手以外，虛線曲線的預測趨勢與實線類似。由於上述這些情況，我們在實驗 2 中實現了補丁學習的概念。



表 2. 基於 MAE、MSE 和 RMSE 的損失評估

	訓練集	驗證集	測試集
MAE	0.0683	0.0876	0.1197
MSE	0.0158	0.0286	0.0438
RMSE	0.1258	0.1692	0.2093



圖 8. 實驗 1 每一手的平均 MSE 損失

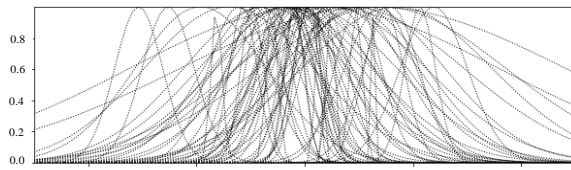
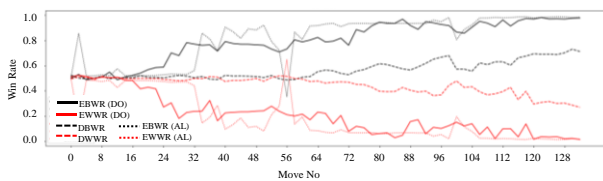
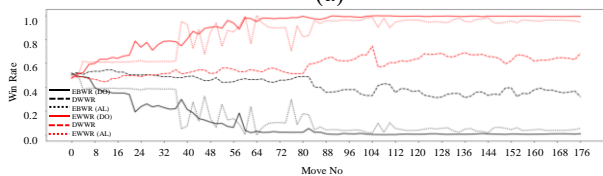


圖 9. 學習 2000 個迭代後的 DBSN 高斯模糊集合



(a)



(b)

圖 10. 實驗 1 迴歸分析 (a) 第 39 盤 (b) 第 58 盤

## 5.2 實驗 2: 3 個補丁的 AI-FML 代理人

在實驗 2 中，我們採用具有三個補丁機制的補丁學習 AI-FML 代理人。我們按照以下方式劃分三個補丁：補丁 1 為第 1 手到第 20 手，補丁 2 為第 21 手到第 40 手，以及補丁 3 為第 41 手到第 90 手。首先我們先學習 1000 代的全域模型(IGM)，然後再學習 100 代的補丁 1 (LPM 3.1)、補丁 2 (LPM 3.2) 和補丁 3 (LPM 3.3)。圖 11 (a)、(b)、(c) 和 (d) 分別顯示了 IGM、LPM3.1、LPM3.2 和 LPM3.3 在訓練期間的訓練集和驗證集的歷史學習曲線。我們也可以觀察到圖 11 (a)、(b)和(d)比圖 11(c)有更好的擬合曲線。也許這是因為第 20 手之後，棋局處於中間階段，處於不確定的情況下，這使得在學習模型時更難適應實際情況。在實驗 2 中，我們設計了兩個子

實驗，描述如下：(1)實驗 2.1：我們僅使用經過訓練 1000 代的 IGM 來預測訓練集。(2)實驗 2.2：我們使用訓練後的 IGM、LPM3.1、LPM3.2 或 LPM3.3 來預測訓練集，並根據每一手所相對應的預測模型。圖 12 顯示了每一手的平均損失曲線。橙色實線和紫色虛線是實驗 2.1 和實驗 2.2 的結果。損失值與實驗 2.1 相比，實驗 2.2 著重在補丁 1、補丁 2 和補丁 3 有下降的趨勢。

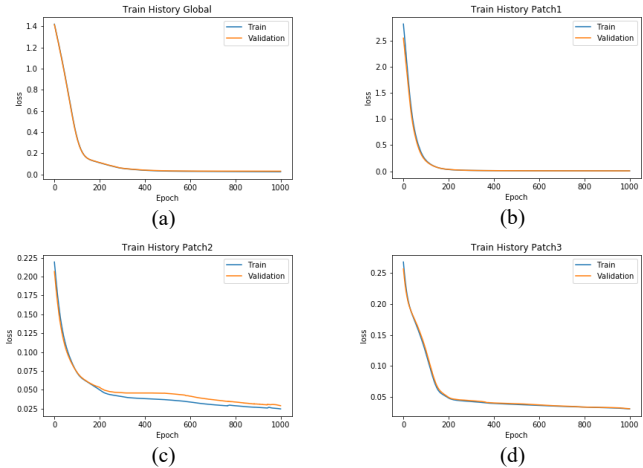


圖 11. (a) IGM、(b) LPM3.1、(c) LPM3.2 和 (d) LPM3.3 的學習曲線

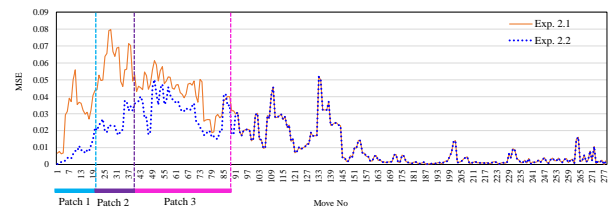


圖 12. 實驗 2 中每一手的 MSE 曲線

## 5.3 圍棋人機共學

圖 13 顯示了學生透過女媧積木與 Kebbi Air 人機共同學習的圍棋機器人應用場景。當虛擬機器人經過自動語音識別(ASR)積木接收人類的語言時，機器人會說出接收到的內容並將收到的語句賦值到指定的變數中。最後再將人類所說的語意經過邏輯規則來推論目前局勢為何。在此系統中我們有六個輸入分別有 DBSN(黑森林黑模擬次數)、DWSN(黑森林白模擬次數)、DBTMR(黑森林黑匹配率)、DWTMR(黑森林白匹配率)、DBWR(黑森林黑勝率)、DWWR(黑森林白勝率)。透過一系列問答，機器人 Kebbi Air 收到以下消息。例如，如果 DBSN 為「低」，DWSN 為「高」，DBTMR 為「高」，DWTMR 為「低」，DBWR 為「高」，而 DWWR 為「低」，接著 Kebbi Air 將會說：「根據 FB ELF OpenGo 的預測，黑棋將獲勝。」未來我們希望為凱比機器人 Kebbi Air 部署 AI-FML 代理人的方塊積木程式，使得該機器人表現得更像人類。

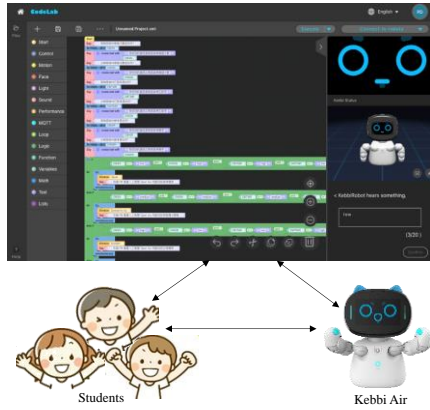


圖 13. 學生透過女媧積木與 Kebbi Air 共同學習機器人應用於圍棋的場景

## 6. 結論

在本論文中我們展示了具有補丁學習機制的 AI-FML 代理人應用於圍棋機器人。驗實驗中執行了各種補丁學習模型，包括全域模型、一個補丁、兩個補丁與三個補丁的模型。此外補丁學習可以在機器人應用 AI-FML 代理人中被實現。我們採用 Google DeepMind Master 60 盤棋局作為訓練集和測試集。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。

## 致謝

作者要感謝國立臺南大學知識應用暨網路服務研究中心 (KWS 中心) 的工作人員以及臺南市仁德小學的相關教職員工。最後，我們還要感謝女媧機器人的參與人員協助構建 ASR 系統，使以將開發的 AI-FML 代理人連接到虛擬機器人 Kebbi Air。

## 參考文獻

- [1] D. Wu and J. M. Mendel, "Patch learning," *IEEE Transactions on Fuzzy Systems*, Jul. 2019. (DOI: 10.1109/TFUZZ.2019.2930022)
- [2] J. M. Mendel, "Adaptive variable-structure basis function expansions: Candidates for machine learning," *Information Science*, vol. 496, pp. 124-149, 2019.
- [3] C. S. Lee, Y. L. Tsai, M. H. Wang, W. K. Kuan, Z. H. Ciou, and N. Kubota, "AI-FML agent for robotic game of Go and AIoT real-world co-learning applications," *2020 World Congress on Computational Intelligence (IEEE WCCI 2020)*, Glasgow, Scotland, UK, Jul. 19-24, 2020.
- [4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484-489, Jan. 2016.
- [5] C. S. Lee, M. H. Wang, L. W. Ko, Y. Hsiu Lee, H. Ohashi, N. Kubota, Y. Nojima, and S. F. Su, "Human intelligence meets smart machine: a

special event at the IEEE International Conference on Systems, Man, and Cybernetics 2018," *IEEE Systems, Man, and Cybernetics Magazine*, vol. 6, no. 1, pp. 23-31, Jan. 2020.

- [6] C. S. Lee, M. H. Wang, Y. L. Tsai, L. W. Ko, B. Y. Tsai, P. H. Hung, L. A. Lin, and N. Kubota, "Intelligent agent for real-world applications on robotic edutainment and humanized co-learning," *Journal of Ambient Intelligence and Humanized Computing*, 2019. (DOI: 10.1007/s12652-019-01454-4).
- [7] C. S. Lee, M. H. Wang, L. W. Ko, N. Kubota, L. A. Lin, S. Kitaoka, Y. T. Wang, and S. F. Su, "Human and smart machine co-learning: brain-computer interaction at the 2017 IEEE International Conference on Systems, Man, and Cybernetics," *IEEE Systems, Man, and Cybernetics Magazine*, vol. 4, no. 2, pp. 6-13, Apr. 2018.
- [8] C. S. Lee, M. H. Wang, S. C. Yang, P. H. Hung, S. W. Lin, N. Shuo, N. Kubota, C. H. Chou, P. C. Chou, and C. H. Kao, "FML-based dynamic assessment agent for human-machine cooperative system on game of Go," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 25, no. 5, pp. 677-705, 2017.
- [9] J. Castanon, "10 machine learning methods that every data scientist should know," May 2019.
- [10] 維基百科, 【自適應神經模糊推理系統】  
[https://en.wikipedia.org/wiki/Adaptive\\_neuro\\_fuzzy\\_inference\\_system](https://en.wikipedia.org/wiki/Adaptive_neuro_fuzzy_inference_system).
- [11] J. S. R. Jang, "ANFIS: Adaptive-Neural-based Fuzzy Inference Systems," *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 23, pp. 665-685, May 1993.
- [12] Y. Tian and Y. Zhu, "Better computer Go player with neural network and long-term prediction," Feb. 2016.
- [13] Y. Tian, J. Ma, Q. Gong, S. Sengupta, Z. Chen, J. Pinkerton, and C. L. Zitnick, "ELF Open Go: An analysis and open reimplement of AlphaZero," May 2019.
- [14] DeepMind, AlphaGo Master series: 60 online games, Jan. 2020, <https://deepmind.com/alphago-master-series>.