AI-FML Agent with Patch Learning Machanism for Robotic Game of Go Application

|  |  |
| --- | --- |
| Yi-Lin Tsai  Dept. of Computer Science and Information Engineering  National University of Tainan  Tainan, Tawain  m10759011@gm2.nutn.edu.tw | Chang-Shing Lee  Dept. of Computer Science and Information Engineering  National University of Tainan  Tainan, Tawain  [leecs@mail.nutn.edu.tw](mailto:leecs@mail.nutn.edu.tw) |

*Abstract—*In this paper, we propose an AI-FML agent with patch learning mechanism for robotic game of Go applications. The proposed AI-FML agent contains three kinds of intelligence, including a perception intelligence, a cognition intelligence, and a computational intelligence, for the robotic application. Additionally, we embed the PL mechanism into the AI-FML agent. The method for running PL involves three steps. It first trains an initial global model, then trains a patch model for each identified patch, and finally updates the global model using the training data that do not fall into any patch. This paper adopts the Google DeepMind Master 60 games to be the training data and testing data set. The experimental results show the AI-FML agent with the patch learning mechanism can improve the performance of regression for robotic game of Go applications.

Keywords—AI-FML, Agent, Patch Learning, Fuzzy Machine Learning, Game of Go

# Introduction

Nowadays, machine learning has been used in multiple real-world fields and industries, such as medical diagnosis, image processing, regression etc. However, training a high performance machine learning model is usually an iterative process which relies on experience and trial-and-error. Sometimes, we need to take some remedies to enhance its performance when it is not satisfactory [1]. Some strategies are implemented to enhance the performance on machine learning mechanisms, such as *using a single deeper model*, *using a single broader model*, *connecting multiple simple base learners in parallel*, or *connecting multiple simple weak learners in series* [1, 2]. In this paper, we propose an AI-FML agent with the patch learning (PL) mechanism to improve the performance of the machine learning [1, 2]. The idea of PL introduced by Mendel is as follows: *Consider a sculptor who is sculpting a human figure, after his first pass at this, the sculptor examines the entire figure and notices that improvements need to be made to certain parts of the figure. He zooms into the certain parts that need more work, after which he blends in the refined portions of the figure with the rest of the figure. He continues such iterative refinements until he is satisfied with the entire figure. Each patch in PL is analogous to a part in the figure that needs more work* [1]. According to [1], PL consists of three steps described as follows: (1) Train an initial global model using all training data first. (2) Identify patches from the initial global model and train a patch model for each patch. (3) Update the global model using training data [1]. Wu and Jerry [1] defined a patch as a connected polyhedron in the input domain. For example, a patch in a 1D input domain is an interval, and a patch in a 2D input domain can be a rectangle, an ellipse, and so on. However, generally identifying the patch locations is a very challenging task, and not every problem can be easily visualized. The PL connects multiple simple models both in parallel and in series to improve the learning performance. Mendel [2] introduced how PL can be performed using rule-based fuzzy systems, because it is easy to initialize patch candidates in a fuzzy system.

Regression, ensemble methods, and deep learning are important machine learning methods for data scientists [9]. An adaptive network-based fuzzy inference system (ANFIS) was proposed by Jang [12] in 1993. The ANFIS integrates both machine learning model and fuzzy logic principles and it has a potential to capture the benefits of both in real-world applications. Its inference system corresponds to a set of fuzzy rules that have learning capability to approximate nonlinear functions and is considered to be a universal estimator [12]. In our previous paper [3], we used AlphaGo Master sixty games as the experimental dataset to make the win rates predicted by the DDF AI bot closer to those predicted by the ELF Open Go AI bot based on FML-based genetic learning (GFML), XGBoost learning, and a seven-layered deep fuzzy neural network (DFNN) learning. In this paper, we further combine FML, PL, and adaptive network-based fuzzy inference system (ANFIS) with the deep learning to train a model to predict the win rates of the AlphaGo Master sixty games. The experimental results show that introducing PL mechanism has a better performance than the methods presented in our previous paper [3].

The remainder of this paper is as follows: Section II presents the structure of ANFIS-based patching learning mechanism. Section III describes the patch learning mechanism for game of Go data set. Section IV proposes the PL-based AI-FML agent for robotic application on predicting the win rate of Go game. Section V shows the experimental results. Finally, section VI draws the conclusion and discussion.

# Structure of ANFIS-Based Patch Learning Mechanism

## Patch Learning Mechanism

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(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間的訓練資料來訓練一個補丁模型(PM1)以減少整體學習損失。接著我們採用第P1+1 (MP1+1)手到第P2 (MP2)手, …,第N (MN)手到第PN (MPN)手落於區間內的資料來訓練補丁模型1 (PM1), 補丁模型 2 (PM2), …, 補丁模型 L (PML) (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，MP1 = 20，MP2 = 40和MP3 = 90。



1. Structure of patch learning for predicting the gane of Go Data set.

## Adaptive Network-based Fuzzy Inference System (ANFIS) Mechanism

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。



1. *Structure of ANFIS with five layers[??].*

# ANFIS-based Patch Learning for Go Game Data Set

## Introduction to Data Set from IEEE WCCI 2020

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。

1. Move 1 to Move 20 of Game 1.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

## Pre-processing Partition Function for Game Data Set

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。



1. Type-1 partition functions with 64 Gaussian functions for Go.

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顯示了兩盤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。

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

1. Prediction results of Darkforest: (a) Game15 (b) Game25 (c) Game29

# PL-based AI-FML agent for Robotic Game of Go Application

## ANFIS for AI-FML Agent on Game

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]。



1. Structure of ANFIS for AI-FML agent

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*)，…，*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。

## PL-based AI-FML Agent for Game of Go Dataset

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6(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6(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6(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。

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

1. Patch Learning模型架構 (a) 1-Patch (b) 2-Patch (c) 3-patch

# Experimental results

## Experiment 1: ANFIS Global Learning for AI-FML Agent

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7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7(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7(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。

1. Loss Evaluation based on MAE, MSE, and RMSE

|  |  |  |  |
| --- | --- | --- | --- |
|  | Training set | Validate set | Testing set |
| MAE | 0.0683 | 0.0876 | 0.1197 |
| MSE | 0.0158 | 0.0286 | 0.0438 |
| RMSE | 0.1258 | 0.1692 | 0.2093 |

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

1. Regression analysis for EXP. 1 in Games (a) 39 and (b) 58.

## Experiment 2: PL-based AI-FML Agent with 3-Patches

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8 (a)、(b)、(c) 和 (d) 分別顯示了IGP、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8 (a)、(b)和(d)比圖8(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IGP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9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。

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

1. Learning curves for (a) IGM, (b) LPM3.1, (c) LPM3.2, and (d) LPM3.3.



1. MSE curves for each move in Exp. 2.

# Conclusions and Discussions

##### 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。

##### References

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. (Accepted)
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, [Online] Available: <https://towardsdatascience.com/10-machine-learning-methods-that-every-data-scientist-should-know-3cc96e0eeee9>.
10. J. S. R. Jang, “ANFIS: Adaptive-Network-based Fuzzy Inference Systems,” IEEE Transaction on Systems, Man, and Cybernetics, vol. 23, pp. 665-685, May 1993.
11. Wikipedia, “Adaptive neuro fuzzy inference system,” 2020 [Online] Available: <https://en.wikipedia.org/wiki/Adaptive_neuro_fuzzy_inference_system>.
12. P. Jagtap and G. N. Pillai, “Comparison of extreme-ANFIS and ANFIS networks for regression problems,” 2014 IEEE International Advance Computing Conference (IACC), Gurgaon, 2014, pp. 1190-1194, doi: 10.1109/IAdCC.2014.6779496.
13. T. G. Ling, M. F. Rahmat and A. R. Husain, “ANFIS modeling and Direct ANFIS Inverse control of an Electro-Hydraulic Actuator system,” 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA), Melbourne, VIC, 2013, pp. 370-375, doi: 10.1109/ICIEA.2013.6566397.