AI-FML Agent for Robotic Game of Go based on Patch Learning Application

|  |  |  |
| --- | --- | --- |
| Chang-Shing Lee, Yi-Lin Tsai, Mei-Hui Wang, Wen-Kai Kuan, Zong-Han Ciou  Dept. of Computer Science and Information Engineering  National University of Tainan  Tainan, Tawain  [leecs@mail.nutn.edu.tw](mailto:leecs@mail.nutn.edu.tw) | Naoyuki Kubota  Dept. of Mechanical Sytems Engineering  Tokyo Metropolitan University  Tokyo, Japan  kubota@tmu.ac.jp | Chang-Shing Lee, Yi-Lin Tsai, Mei-Hui Wang, Wen-Kai Kuan, Zong-Han Ciou  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*—電腦對局是人工智慧中相對重要的一部分，其中圍棋也被認為是目前世界上最複雜的棋盤遊戲之一，也是人工智慧中相當大的挑戰。電腦圍棋發展至今，由各個比賽的推動使得電腦圍棋在棋力上有明顯的進步甚至更超越於人類的思維。本篇論文延續IEEE WCCI 2020 FML-based Machine Learning Competition，先前的研究使用多種機器學習方法包括AI-FML基因學習、XGBoost集成學習、DNN神經網路來預測圍棋勝率。然而我們提出了一個基於ANFIS適應性網路模糊推論系統，並結合Patch Learning來優化我們的圍棋勝率預測模型。從實驗結果可以看出透過Patch Learning部分的補修與加強學習不佳的區域可以有效的降低錯誤率，使的均方誤差明顯的下降。

Keywords—Patch Learning, Fuzzy Machine Learning, Game of Go

# Introduction

電腦圍棋的發展已受到世界各國的關注，學界已思考如何將電腦圍棋決策模式應用至日常生活當中。圍棋是一個極富有挑戰性的遊戲，每一步棋都必須經過深入的思考、策略的權衡及空間的推論，因此近年來世界各國有諸多學者致力於電腦圍棋程式的研究。隨著電腦圍棋演算法的突破，人類到目前為止已經很難超越電腦了。跳脫傳統單純由人類跟 AI 對抗競爭關係，人類更謙虛像是學生在跟機器人老師的方式學習，這對人類學習是一個很好的模式，未來對弈會更有趣。

# System structure for game of Go application

## Patch Learning Mechanism

Patch Learning是由美國南加州大學Jerry M. Mendel與Dongrui Wu所提出的一種機器學習方法。所謂的Patch Learning是能夠有效的掌握表現不好的區間，經過訓練一個全域的模型後並結合ANFIS適應性類神經模糊推論系統的方式找出這些誤差大的Patch，透過多個斷點間的學習我會得到Global Model、Patch1 Model、Patch2 Model…Patch(n) Model。然而在機器學習中我們有很多種方法可以改善我們的模型，例如加深和加寬神經網路或是添加一些非線性的激發函數來最佳化我們的模型。或是使用集成學習的方法集合許多策略，最終形成一個強學習器並改善某些區域的弱點。圖一為Patch Learning的預測架構圖。在使用Patch Learning模型前我們要找出該段輸入所對應的Patch，若該區間剛好落於所劃定的範圍內，這些輸入就將會對應到相對應的Patch Learning model，否則就會使用Global Model進行預測。

****

1. Patch Learning架構流程

## Adaptive Network-based Fuzzy Inference System Mechanism

適應性網路架構的模糊推論系統（Adaptive Network-based Fuzzy Inference System, 簡稱ANFIS）。是模糊化類神經網路中最典型的一種方法，適用於Sugeno模糊規則與Tsukamoto模糊規則。最早由張智星博士在於1993年所創先提出[1]，目前此系統已被廣泛應用在諸多領域中，例如決策補助系統[2]，行為預測分析[3]…等。傳統模糊控制規則的建立必須依靠專家的不斷嘗試錯誤與修正才能建立很好的模糊規則，而適應性網路架構的模糊推論系統透過機器學習能達到減小誤差，補足表現不佳的缺點。本篇論文中我們實作ANFIS架構並採用PyTorch框架進行深度學習，透過倒傳遞演算法對每一參數進行修正調整。圖二為一個簡單的範例，假設模糊系統只有兩個輸入變數X與Y，與一個輸出Output。第一層為輸入層並定義隸屬函數，在此架構中使用高斯函式計算隸屬值。第二層的神經元π處理T-norm並執行「AND」運算，以便獲得模糊規則的前鑑部(antecedent)的啟動強度。第三層的神經元N代表正規化負責處理「OR」的運算，以便將所有模糊規則的前鑑部起動強度聯集起來。第四層為後鑑部(consequent)計算透過自定義函式加總起來。第五層的類神經元執行去模糊化的運算，以提供明確的輸出值。



1. 兩個輸入和一個輸出的ANFIS網路架構

# FML-BASED MACHINE LEARNING COMPETITION AT IEEE WCCI 2020

## Introduction to DataSet

The authors would like to thank the financially support sponsored by the Ministry of Science and Technology of Taiwan under the grants MOST 108-2218-E-024-001.

機器學習總共分為五個步驟，分別是定義問題、特徵蒐集與資料清理、選擇與建立模型、分析結果及修正模型、結果呈現。首先在此實驗中我們拿取AlphaGo Master60盤棋局作為我們的資料來源，每一盤的圍棋資訊都包含六個特徵，分別為DBSN(黑森林黑棋落子位置的模擬次數)、DWSN(黑森林白棋落子位置的模擬次數)、DBWR(黑森林黑棋落子位置的勝率)、WBWR(黑森白棋落子位置的勝率)、DBTMR(黑森林黑棋落子位置的匹配率)、WBTMR(黑森林白棋落子位置的匹配率)。每一手都有相對應的標準答案也就是 ELF OpenGO 所預測的黑白勝率，EBWR (ELF OpenGo黑棋勝率)、EWWR (ELF OpenGo白棋勝率)。我們的目標是要透過這六個特徵，利用機器學習的方式去學習出一個模型，來預測該手狀態下的勝率。表一是實驗Game1~Game40部分訓練資料共有3758筆數據，包含六個輸入特徵和兩個輸出特徵。

1. 部分實驗訓練資料

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | *DBSN* | *DWSN* | *DBWR* | *DWWR* | *DBTMR* | *DWTMR* | *EBWR* | *EWWR* |
| 1 | 3863 | 2274 | 0.52733 | 0.48533 | 0 | 1 | 0.49636 | 0.50884 |
| 2 | 9283 | 7866 | 0.51529 | 0.48532 | 0.5 | 1 | 0.45779 | 0.54844 |
| 3 | 11395 | 6798 | 0.51265 | 0.47717 | 0.6667 | 1 | 0.45671 | 0.57235 |
| 4 | 4499 | 10703 | 0.51885 | 0.46988 | 0.75 | 1 | 0.49086 | 0.5201 |
| 5 | 7388 | 20017 | 0.5288 | 0.46679 | 0.8 | 1 | 0.47218 | 0.53649 |
| 6 | 20098 | 9693 | 0.53309 | 0.46602 | 0.8333 | 1 | 0.47249 | 0.52224 |
|  | | | | | | | | |
| 3754 | 7211 | 12038 | 0.92754 | 0.28941 | 0.797 | 0.812 | 0.9566 | 0.08276 |
| 3755 | 20017 | 20134 | 0.81634 | 0.18352 | 0.7985 | 0.8134 | 0.87651 | 0.04687 |
| 3756 | 7691 | 12699 | 0.80645 | 0.15954 | 0.7926 | 0.8074 | 0.92438 | 0.18538 |
| 3757 | 17076 | 4603 | 0.85503 | 0.09789 | 0.7941 | 0.8088 | 0.9107 | 0.18915 |
| 3758 | 9015 | 20017 | 0.90843 | 0.10819 | 0.7956 | 0.8102 | 0.98549 | 0.00822 |

## Data Clean

資料清理是資料前處理的第一步，意指刪除、更正資料集中錯誤、不完整、格式有誤或不合理以及多餘的資料。在本次實驗中總共有60盤棋局其中前40盤為訓練資料共有3758筆數據，剩下20盤為測試資料共有1880筆數據。每一盤的圍棋資訊都包含六個特徵，分別為DBSN(黑森林黑棋落子位置的模擬次數)、WBSN(黑森林白棋落子位置的模擬次數)、DBWR(黑森林黑棋落子位置的勝率)、WBWR(黑森白棋落子位置的勝率)、DBTMR(黑森林黑棋落子位置的匹配率)、WBTMR(黑森林白棋落子位置的匹配率)。首先我們要找出每一盤棋局不合理的盤面，判斷的標準分為兩個層面。第一個為黑森林最終的預測結果是否與實際勝負吻合，若兩者勝率預測有誤即不採與訓練。最終我們從訓練集挑出Game15、25、29以及測試集Game48、60並移除，如圖所示。第二個判斷不合理局面的標準為黑森林黑白手與ELF OpenGo黑白手的預測勝負，當兩者預測出來的勝負相反的筆數超過該盤手數的40%即為不合理資料。例如100筆當中有40筆資料雙方的預測勝負相反即不納入訓練集。最終我們從訓練集挑出Game8、36以及測試集Game44、47、56、57並移除，如圖3 (a)(b)(c)、圖4(a)(b)所示。最後我們在檢查數據集中是否有缺失值，發生缺失值的狀況通常會發生在最後一手，當白棋先攻與最後一手為白棋，或是黑棋先攻與最後一手為黑棋，就會發生最後一手缺失值狀況發生。經檢查後我們在訓練集清理掉19筆，測試及清理6筆數據。最後資料清理結果在數據集中我們保留3143筆有效資料，測試集中我們保留1239筆有效資料。

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

1. 訓練集黑森林預測錯誤棋局(a) Game15 (b) Game25 (c) Game29

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

1. 測試集黑森林預測錯誤棋局(a) Game48 (b) Game60

# MACHINE LEARNING MECHANISMS

## ANFIS on Game

ANFIS結合了兩種機器學習的優勢，分別為模糊邏輯與神經網絡。並透過神經網路學習方法來調整模糊推論系統的參數，可調控的參數包含每個高斯隸屬函數mean與variance。表二為ANFIS結合倒傳遞演算法虛擬碼。輸入為訓練集40盤棋局資料以及必要參數，n\_inputs(特徵數量)與n\_rules(規則數量)。輸出為預測目前黑棋勝率。演算法第一步神經網路初始化。第二步類神經元執行輸入與相對應的模糊集合的歸屬程度計算。第三步規則層類神經元Π執行T-norm運算，執行的是模糊規則前鑑部的啟動強度計算。第四步正規化層類神經元N，執行的是將啟動強度正規化之運算。第五步推論層類神經元執行的是函數式加乘，每個模糊規則後項之運算。第六步輸出層只有單一個類神經元，計算前一層中類神經元輸出值的總合，執行解模糊化以作為最後網路的輸出值。第七步預測並計算每一筆資料的Error並得出損失值，並藉由損失函數找出梯度慢慢的修正每個模糊規則的mean與variance。第八步判斷是否停止訓練，當到達訓練次數即終止訓練。

1. ANFIS 演算法架構

|  |
| --- |
| **Input:**  訓練資料: *X\_train* // Game1~Game40局圍棋資料  輸入數量: *n\_inputs* = 6 // DBSN、WBSN、DBWR、WBWR、DBTMR、DWTMR  規則數量*: n\_rules* = 64  **Output:**  預測目前黑棋勝率   1. 神經網路初始化    1. 建立mean與variance 陣列       1. *meanArray* = Array(*n\_inputs\* n\_rules*);       2. *varianceArray* = Array(*n\_inputs\* n\_rules*);    2. FOR(*i* ← 0 to *n\_inputs\* n\_rules*) // 隨機0~1初始化mean與variance 陣列       1. *meanArray*[*i*]= random(0,1);       2. *varianceArray* [*i*]= random(0,1); 2. 輸入層計算歸屬程度    1. // 使用高斯歸屬函數 3. 規則層T-norm運算    * 1. *.* 4. 正規化層 5. 推論層 6. 輸出層 7. 計算loss並計算梯度更新權重 8. 返回step2直到訓練結束 |

## Patch Learning on Game

Patch Learning是一個新的機器學習的一種想法，搭配模糊推論讓整個訓練出來的模型更具人性化。表三為Patch Learning其演算法架構，首先第一步我們需要使用所有的訓練資料訓練一個全域模型。訓練模型採用ANFIS架構並設定六十四條規則訓練一千代。接著透過梯度下降演算法來修正模型參數，並使用均方誤差(mean squared error, MSE)來評估模型好壞。透過計算MSE我們可以知道每次訓練過後模型所推論出來的勝率是否趨近ELF OpenGO所預測出來的勝率。因此MSE計算出來的值越小代表越接近ELF OpenGO的勝率。第二步從全域模型中找出找出某個區段表現最差的區域，也就是MSE最大的區段。並為每個此類的Patch訓練一個區域的模型，訓練模型我們一樣採用ANFIS架構並設定六十四條規則訓練一千代。第三步計算每一手的誤差並觀察是否還有學習不佳的區域，若有則返回第二步繼續訓練，反之停止訓練。整個模型訓練完畢後我們就可以使用測試資料來測試我們模型。在使用Patch Learning模型預測前我們要找出該段輸入相對應的Patch。假設我們有k個Patch若該筆資料落於第一個Patch我們就採用第一個Patch進行預測。若該筆資料皆無落在預訓練的Patch中則使用全域模型進行預測。

1. Patch Learning演算法架構

|  |
| --- |
| **輸入**  訓練資料: *trainData* // Game1~Game40局圍棋資料  測試資料: *testData* // Game41~Game60局圍棋資料  **輸出**  Patch Learning模型對圍棋資料的勝率預測  **方法**  // 訓練Patch Learning模型   1. 訓練全域模型    1. 使用ANFIS進行N代訓練trainData    2. 計算出MSE並評估模型好壞 2. Patch區間訓練    1. 從*trainData*中挑選出學習不佳的資料patchData    2. 使用ANFIS進行N代訓練patchData    3. 計算出MSE並評估模型好壞 3. 挑選學習不佳區間並返回Step2進行訓練直到所有區間訓練完畢   // 使用Patch Learning模型進行預測  for *t* = 1, ..., do  = True;  if 落入第k個Patch then  使用第*k*個Patch模糊模型進行預測;  = False;  Break;  end  if *useGlobal* == True then  使用更新的AI-FML全域模糊模型進行預測;  end  end |
|

透過Patch Learning的概念我們將此學習模式運用在圍棋資料預測模型上。如圖五(a)(b)(c)是基於Patch Learning應用在圍棋資料，首先我們會先將所有訓練資料訓練出一個Global Model，接著我們透過找出學習不佳的區域來加強區段間的訓練。圖五(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) |

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

# Experimental results

在本論文之實驗中我們分別進行了

# Conclusions and Discussions

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集成式學習雖然學習速度快，但是在驗證集的表現與訓練集有落差，我們可以透過不斷的調整參數來解決過度擬合的問題，或是使用Early Stopping的機制提早讓模型訓練結束。而在深度神經網路的方法，因為有加上了Dropout機制夠有效的阻止過度擬合的發生。

##### Acknowledgment

The

##### References

1. J.-S. R. Jang, " ANFIS: Adaptive-Network-based Fuzzy Inference Systems," IEEE Trans. on Systems, Man, and Cybernetics, vol. 23, pp. 665-685, May 1993.
2. 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.
3. 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.