深度學習在預測圍棋著手之應用

呂金寶 [1], 曾子丞 [2], 顏士淨 [3]
[1, 2, 3] 國立東華大學資訊工程學系
E-mail: {[1]410921252, [2]410921231, [3]sjyen@gms.ndhu.edu.tw}

摘要

本研究利用深度學習技術進行圍棋著手預測。圍棋需要高度的觀察和判斷能力來決定每一步的最佳落子位置。隨著深度學習技術的快速發展，有越來越多的方法相繼被提出。本研究參加了「2023 AI CUP 秋季賽 圍棋棋力模仿與棋風辨識競賽」，利用提供的棋局資料進行模型訓練。本研究融合不同的神經網路架構和一種分階段的訓練方式。實驗結果顯示，該方法在測試集中取得了優異的表現，最終在比賽中獲得金牌及創意獎。

關鍵字：深度學習、神經網路、圍棋

Abstract

This study investigates the utilization of deep learning methodologies to forecast moves in the game of Go, which demands keen perception and astute discernment to determine the most strategically advantageous placement for each move. Inspired by the rapid development of deep learning technology and its innovative applications, we developed a novel approach for the "AI CUP 2023 Competition on Models of Virtual Players in the Metaverse of Go." Our method combines various neural network architectures and applies a two-step training strategy using the provided game data. The experimental results show that this approach achieved outstanding performance on the evaluation dataset, obtaining the first place award and receiving an innovation prize in the competition.

Keywords：deep learning, neural network, Go

1. 研究方法

本論文將探討如何使用深度學習技術來進行預測人類棋手面對一個圍棋盤面的下一步著手。本章節依序介紹資料處理、模型架構以及訓練步驟。

1.1 資料處理

本論文訓練集來自 AI CUP 官方提供的 ***kyu*** (10級玩家) 和 ***dan*** (初段玩家) 的棋局紀錄，分別包含 118,500 和 100,160 筆資料。原始資料格式需轉換為SGF格式，以便使用 Sgfmill 函式庫進行處理。[1] 每盤棋的紀錄視同一個字串，使用逗點分隔每一步著手，並進行模擬盤面及棋盤編碼。

每筆資料提供了一個棋局的所有著手，以及預測著手的指定顏色 (B 或 W)。我們可以對棋局中的前 N-1 手所形成的盤面編碼，並將其與第 N 手的著手位置結合，形成一筆訓練資料，我們跳過了 N=1 的情況，即不使用空盤面作為訓練資料，這種方法使我們能夠從原始訓練集中生成約 2,000 多萬筆的訓練資料。這些生成的訓練資料被分為兩類：一類是與預測著手顏色「相同」的 10M 訓練集；另一類是與預測著手顏色「不相同」的 20M 訓練集，這樣的分類有助於更有效地運用這些資料去進行模型的訓練。各自的 10M 和 20M 訓練集都是從所屬的原始資料中獨立生成。

1.2 模型架構

我們所使用架構以 KataGo[2]的策略網路為發想，設計出適合本次任務的模型，在這當中也融合了許多經典模型所使用的方法，以達到最好的結果。我們將模型主要分為 Input Conv、Blocks、Policy Head 需要調整的參數分別為：深度主要由 Blocks 中模組的數量決定、寬度設定為 Main Channels。所有模型內的卷積層和線性層都沒有進行額外的參數初始化，圖 1 為整體模型架構。其中 Input Conv 擴大輸入資料的 Channels 成為模型的 Main Channels。SE Block (Squeeze-and-Excitation)增強神經網路對全局資訊的理解能力，根據學習任務自適應地調整各 Channels 的激勵程度。Policy Head 將資料映射成所屬的機率分布，輸出每個棋盤位置的

著手機率分布。

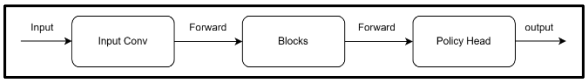

圖 1. 整體模型架構

此外在 Blocks 中還加入以下三個架構：

- NormActConv：結合 Batch Normalization、Activation Function 和卷積層。
- InnerResidualBlock：跳接網路設計，通過兩個 NormActConv。
- NestedResidualBlock：跳接網路設計，通過兩個 InnerResidualBlock。

1.3 訓練步驟

1. 載入已經完成資料前處理的 10M 訓練集。
2. 將資料以 8:2 的比例切分為訓練集和驗證集
3. 進行指定 Epoch 的訓練，包括 Train Step + Valid Step。對驗證損失進行 Early Stopping (patience = 2)，並儲存在 Valid Step 達到最高準確度的模型權重。
4. 在完成 10M 訓練集的訓練後，載入儲存的模型權重，以相同的方式訓練 20M 訓練集。

2. 研究結果與討論

表 1 和表 2 為實驗結果，相關的設定與說明如下：

1. 將模型進行英文字母編號，以方便說明。
2. 是否使用 Pre-Activation Policy Head，N 代表沒使用；Y 代表有使用。
3. Loss Function 的使用。
4. 模型深度，主要分為 5 層和 7 層。
5. 模型 Main Channels 數，皆為 192。
6. 使用的訓練集：10M / 20M。
7. 預測 Public 測試集的 Top1 分數、Top5 分數、加權後的分數。(比賽所使用的加權公式為 Top1 分數 * 0.25 + Top5 分數 * 0.1)
8. 預測 Private 測試集的結果排名。透過不同的組合所得到的分數進行交叉比對，得到相對的分數排名，數字越小越好；X 代表沒提交過 Private。
9. B、C 模型的訓練是先載入 A 模型的權重再完成 20M 訓練集的訓練。E 模型的訓練是先載入 D 模型的權重再完成 20M 訓練集的訓練。G 模型的訓練是先載入 F 模型的權重再完成 20M 訓練集的訓練。

表 1. 著手預測 (kyu) 結果

ID	Pre-Activation Policy Head	Loss Function	Depth	Main Channels	訓練集	Public Top1	Public Top5	Public 加權	Private
A	N	Cross-Entropy	5	192	10M	0.565873	0.86164	0.22763225	X
B	N	Focal Loss	5	192	20M	0.568519	0.863404	0.22847015	X
C	N	Cross-Entropy	5	192	20M	0.574074	**0.865168**	0.2300353	**1**
D	N	Cross-Entropy	7	192	10M	0.568342	0.863051	0.2283906	X
E	N	Cross-Entropy	7	192	20M	0.571958	0.863757	0.2293652	2
F	Y	Cross-Entropy	5	192	10M	0.571693	0.861552	0.22907845	4
G	Y	Cross-Entropy	5	192	20M	**0.575661**	0.862257	**0.23014095**	3

表 2. 著手預測 (dan) 結果

ID	Pre-Activation Policy Head	Loss Function	Depth	Main Channels	訓練集	Public Top1	Public Top5	Public 加權	Private
A	N	Cross-Entropy	5	192	10M	0.554	0.862273	0.2247273	X
B	N	Focal Loss	5	192	20M	0.563545	0.867909	0.22767715	3
C	N	Cross-Entropy	5	192	20M	0.559091	**0.865727**	0.22634545	**X**
D	N	Focal Loss	7	192	10M	0.558273	0.867182	0.22628645	X
E	N	Focal Loss	7	192	20M	0.562	0.869818	0.2274818	2
F	Y	Focal Loss	5	192	10M	0.557909	0.865182	0.22599545	4
G	Y	Focal Loss	5	192	20M	**0.567909**	0.869545	**0.22893175**	1

模型在 kyu 訓練集上的 Top1 準確率優於 dan 訓練集，而在 Top5 準確率上情況相反。可能是因為 kyu 訓練集包含較多簡單的走法，使模型更容易在 Top1 準確率上取得高分；而 dan 訓練集包含更多複雜的走法，使模型在 Top5 準確率上表現更佳。加權後的結果顯示，模型在 kyu 的綜合表現優於 dan。此外，我們發現分階段的訓練方式是成功的，增加模型深度效果不顯著，Loss Function 的選擇影響性能，Pre-Activation Policy Head 對 Public 測試集有提升效果，但對 Private 測試集則不一定。

致謝

本研究感謝國科會計畫 113-2221-E-259-018-MY3, 113-2622-E-259-001, 112-2634-F-A49-004 和東華大學亮點計畫支持。

3. 參考文獻

[1] Python library for Smart Game Format (SGF) files, https://github.com/mattheww/sgfmill.

[2] David J. Wu, "Accelerating Self-Play Learning in Go", arXiv:1902.10565, 2019.

[3] CGLemon, 圍棋引擎 Sayuri 開發日誌. https://bit.ly/3QwEKC8.

[4] CGLemon, AlphaZero 之加速演算法實作. https://bit.ly/4drnI1Y.