人工智慧期末專題報告

A.I. Final Project Proposal

教授: 林傑森

Beta Gobang - 基於強化學習(Reinforcement Learning)的五子棋

第 8 組

洪郁修 4108056036

江尚軒 4108056005

王思正 4108056004

導論

五子棋遊戲 A.I. 我們預計經由 RL 方式訓練來訓練五子棋 A.I. 模型 - Beta GoBang，成品的命名參考著名的圍棋 A.I. 阿爾法 Go (AlphaGO)，透過合適的訓練演算法、優化器與強化學習模型訓練 A.I.，令其能夠下五子棋並與玩家對弈，在最後將遊戲透過 PyGame 或其他套件完成渲染並製作成 App 發布或是以網頁的形式展現我們的作品。

背景

現在人工智慧在實際應用上已有不少的案例，比如影像辨識領域、自然語言處理等，這些背後是許多機器學習研究的成果，其中之一就是強化學習 (Reinforcement learning)，強化學習是基於動態規劃與馬可夫決策過程 (Markov decision process)，其獎勵函數會考慮過去的樣本，因此能有效解決動態的最佳化問題。強化學習訓練出來的模型本質上是對一個動態問題的策略 (Policy)，可以用來解決相似的問題而不需要重新再訓練或計算，具有通用性，我們預計會參考 AlphaGO 的設計理念與相關的演算法(蒙地卡羅搜尋等)來實作我們的 Beta Gobang。

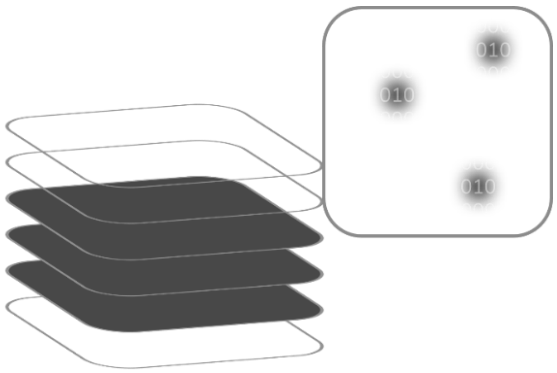
在這次期末專題中，我們想要嘗試使用強化學習 (Reinforcement Learning) 技術，其中較為人知的應用就是在遊戲上，比如說 AlphaGo，在實戰中一步一步將柯潔等數位人類棋王逼入絕境，更可怕的是 AlphaGo 還會藉由學習變得越來越強，這也促進了強化學習的發展，也因此誕生出更新更強的圍棋 A.I.，比如 AlphaGo Zero 等。

動機

我們查到許多關於 Reinforcement Learning 的遊戲 A.I. 與相關實作分享，但大多都過於複雜，而且訓練資料難以取得，加上又必須在短時間內拿出成品，於是我們選擇相對較少人做過，複雜度又相對簡易的五子棋遊戲作為環境來訓練我們的 A.I.，在強化學習的框架與遊戲環境互動的訓練下，打造出屬於自己的五子棋遊戲 A.I. - Beta Gobang。

如此一來，以後自己一個人感到無聊時，也可以和 Beta Gobang 來場戰況膠著的五子棋了!

實作方法



1. **五子棋遊戲環境**

五子棋的規則與勝利條件:

* 棋盤大小為 9*9
* 黑子先下，再換白子
* 五顆相同顏色的棋子連成一線(對角亦算)則執棋方勝利
* 環境輸出的 observation 如右圖，以自己和對手去 3 步落子的特徵平面加上己方顏色特徵平面，大小為 9x9x(3+3+1)
* 使用 OpenAI/gym 的框架實作

2. **RL 演算法**

A. 智慧體

* 參考 AlphaGo Zero 論文使用 Policy-Value Network
* Policy Network 輸出當前 observation 下動作機率分布
* Value Network 輸出當前 observation 的 Reward 估值
* 使用蒙地卡羅搜尋樹輔助

B. 訓練演算法

* 以蒙地卡羅搜尋搜尋樹來讓智慧體對我們的環境進行採樣
* 訓練採用 self-play 方式，自己生成資料自己訓練

C. 實作方式:

* 使用 stable-baselines3 的套件來實踐訓練演算法
* 使用 PyTorch 套件來實作 Policy-Value Network

3. **視覺化遊戲界面**

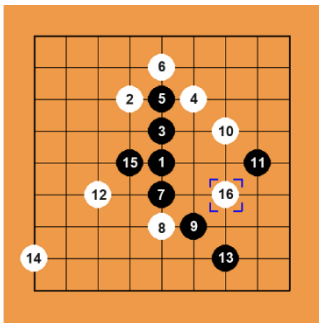
* 內部亦有實作遊戲規則
* 串接玩家與 Beta Gobang
* 使用 PyGame 進行實作

主要使用到的開發環境與程式語言/套件

* Python : 程式語言
* PyGame：視覺化界面
* PyTorch：模型訓練、神經網路建立
* OpenAI: 強化學習框架，包含 gym、baseline3 等
* VScode: 輕量化編譯器

成果

模型強度分析



900 回合

1. 訓練約 10 回合:

=> 尚對遊戲環境不熟，學習規則中

=> 落子策略較為隨機

=> 會落子在非法的地方導致智慧體輸掉

2. 訓練約 900 回合:

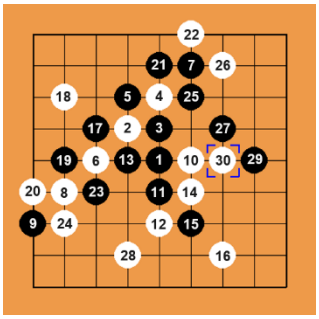
=> 最初幾子會傾向往中間地方下

=> 開始學會基礎的防守(EX:擋住活三等)

3. 訓練約 3100 回合:

=> 人類(我們)開始打不過

=> 策略較傾向保守，除非人類故意露出破綻



3100 回合

4. 訓練約 6700 回合:

=> 勝率開始逼近 alpha-beta pruning(~48.4%)

效能分析

1. 使用自定義環境、預設 RL 模型:

=> 訓練速度快、模型效果差

2. 使用自定義環境、參考 AlphaGo Zero 框架:

=> 訓練速度慢、模型效果極佳

3. 使用自定義環境、參考 AlphaGo Zero 框架，使用 GPU 加速:

=> 訓練速度稍微改善、模型效果極佳

4. 結論: 訓練速度慢的原因在於智慧體需要與環境互動(self-play)來產生資料集來自我訓練，但 GPU 加速只對神經網路的訓練有比較好的加速效果，對於資料集的產生速度助益較少

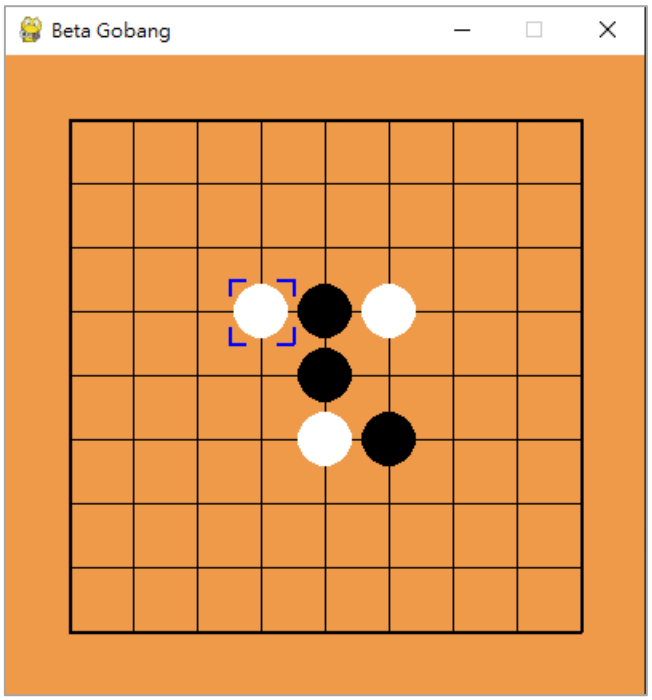
比較(與 alpha-beta pruning)

alpha-beta pruning 會計算未來棋局的所有可能性，並選擇對自己最有利的落子位置。這個方法理論上是**最佳解**，但因為需要計算所有可能性，因此越到終局**等待時間較久，效能較差**。相較之下，強化學習的方法雖然可能不是找到最佳解，但效能比較好，但在 AlphaGo Zero 框架幫助下，勝負表現已相差無幾

結論

最終成果

如預期實作出 Beta Gobang，並以 PyGame 渲染成一個小品遊戲



過程中遇到的困難

1. 一開始採用預設模型(PPO、DQN)的效果不佳
2. 訓練模型成本曠日費時
3. 前端與遊戲環境內部邏輯不一致，導致串接不順利
4. 套件版本問題衝突

解決方式

1. 參考 AlphaGo Zero 論文的模型及框架，改用 Policy-Value Network 並以蒙地卡羅搜尋樹輔助決策
2. 同學分別升級 RTX 3060ti 及 RTX 3050 提升神經網路訓練速度，花了大把錢
3. 成品採用雙環境(PyGame 與 gym)表示，提供 API 互相串接
4. 使用 virtualenv 在虛擬環境下開發

參考資料

【機器學習 2021】概述增強式學習 (Reinforcement Learning, RL) https://www.youtube.com/watch?v=XWukX-ayIrs&list=PLJV_el3uVTsMhtt7_Y6sgTHGHp1Vb2P2J&index=30&ab_channel=Hung-yiLee 2022/10/15

強化學習 五子棋演算法 https://www.gushiciku.cn/pl/p3jQ/zh-tw 2022/10/15

AlphaGo Zero 論文 https://www.nature.com/articles/nature24270.epdf?author_access_token=VJXbVjaSHxFoctQQ4p2k4tRgN0jAjWel9jnR3ZoTv0PVW4gB86EEpGqTRDtpIz-2rmo8-KG06gqVobU5NSCFeHILHcVFUeMsbvwS-lxjqQGg98faovwjxeTUgZAUMnRQ

OpenAI Gym 開發文件 https://www.gymlibrary.dev/

stable-baseline3 開發文件 https://www.gymlibrary.dev/

PyGame 開發文件 https://www.pygame.org/news

https://www.cnblogs.com/zhiyiYo/p/14683450.html 2022/10/15

GitHub - Alpha Gobang Zero https://github.com/zhiyiYo/Alpha-Gobang-Zero 2022/10/15