

**Gomoku Deep**

李佳陽, 王泰淞, 張騏岳, 江尚紘, 曾士桓

***Symposium on Digital Life Technologies*, 2019**

Speaker：曾士桓

1

## Introduction

* Self-play reinforcement learning on Gomoku
* Experiment results
* Conclusion

### AlphaGo[1] 讓人工智慧再度受到重視

* + 結合強化學習(Reinforcement learning)與深度學習

(Deep learning)的技術

### AlphaGo Zero[2] 有更好的表現

* + 僅使用深度強化學習(Deep reinforcement learning)
  + 以Self-play reinforcement learning稱之

### 目的：實作Self-play reinforcement learning在五子棋對奕上

* + 五子棋的規則相較於圍棋簡單
  + 訓練時間也相較少

1. David Silver et al., ‘‘Mastering the game of Go with deep neural networks and tree search’’,

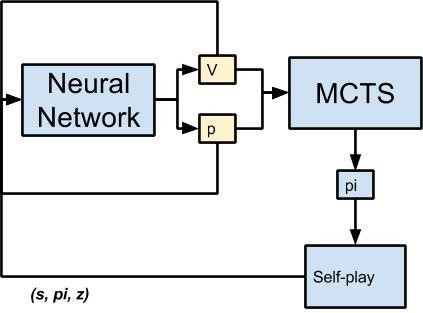
*Nature*, 2016.

## AlphaGo Zero 結構圖[1]

𝑃: 盤面上每個位置的落子概率

𝑉: 目前盤面評分

Monte Carlo Tree Search

𝑠:目前盤面

𝑧:下完的輸贏結果

盤面上每個位置的落子概率



Self-play reinforcement learning on Gomoku



## 模型的評估方式

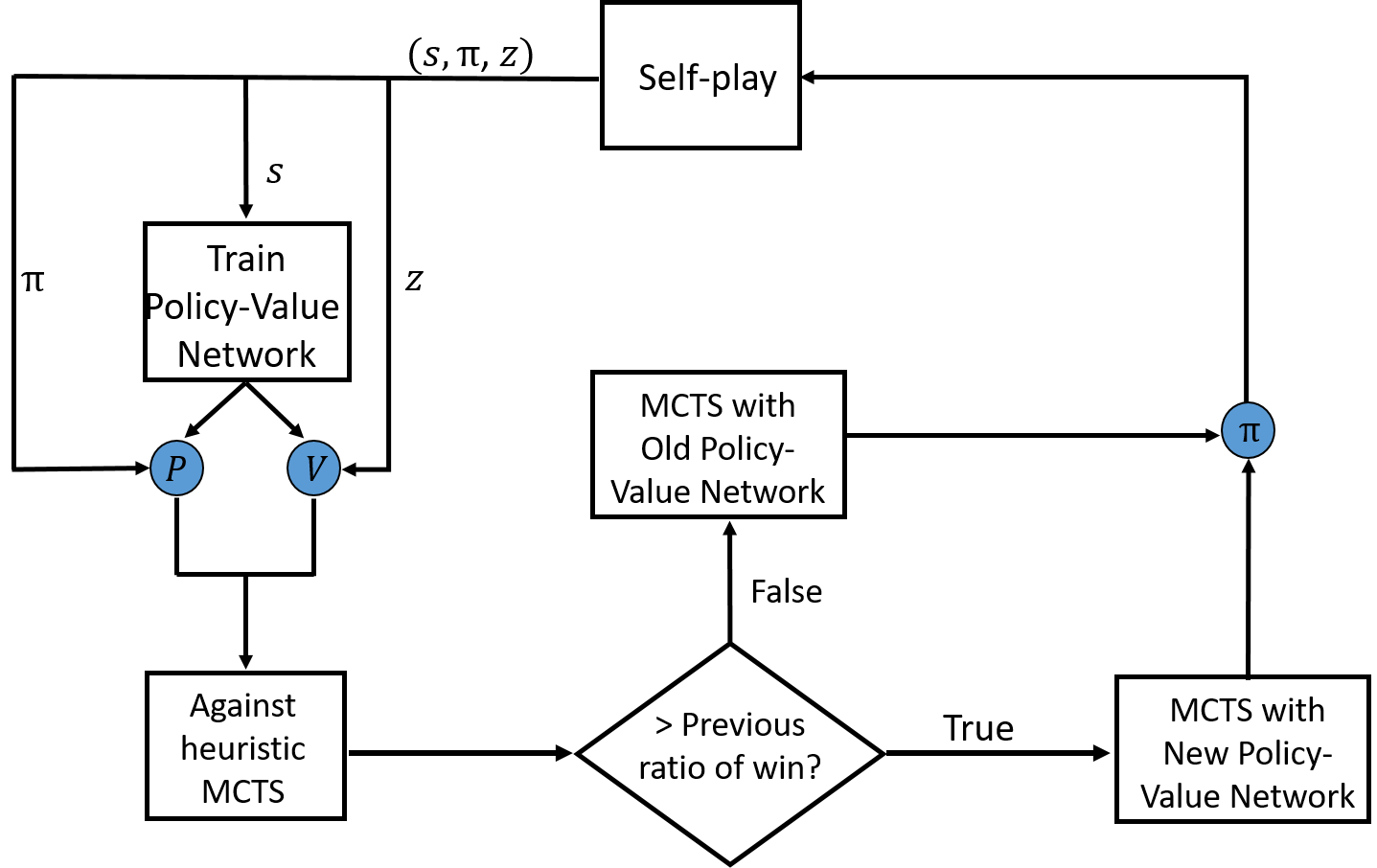
* + - AlphaGo Zero
      * 新的模型與舊的模型對戰，有如自己去和與自己實力 相當的人對戰
      * 收斂速度很慢
    - Gomoku Deep
      * 新的模型與啟發式的MCTS對戰，有如自己去跟會下 棋的人對戰
      * 收斂速度較快

## 系統架構

𝑠:目前盤面

π: MCTS輸出盤面上每個位置的落子概率

𝑧:下完的輸贏結果

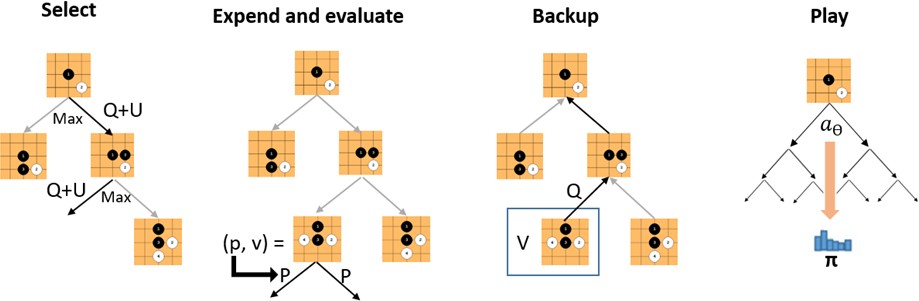


𝑃:盤面上每個位置的落子概率

𝑉:目前盤面評分

## Monte-Carlo tree search (MCTS)

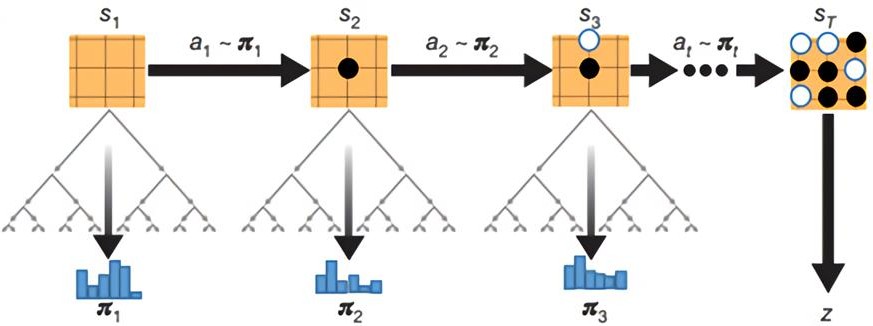
* + - 對抗式遊戲常用的啟發式搜尋演算法



# Self-play reinforcement learning on Gomoku



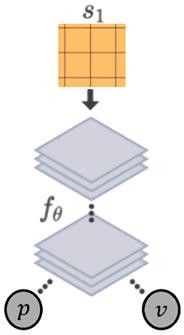
## Self-play for training

* + - 自行產生 training data
    - 生成data的多樣性：加入exploration的方法
    - Data 保存和擴充：因對稱性，可旋轉和鏡像

# Self-play reinforcement learning on Gomoku



## 策略價值網路(policy-value network) 𝑓𝜃

* + - 輸入：盤面 𝑠
    - 輸出： = 𝑓𝜃(𝑠)

𝑝, 𝑣

* + - * 盤面每個位置的機率 𝑝
      * 盤面的評分值 𝑣
    - 卷積層：3
      * 32、64、128個 3x3 filter
    - 策略(Policy)
      * 4個1x1 filter 降維
      * 1個全連接層，透過softmax輸出
    - 價值(Value)
      * 2個1x1 filter 降維

示意圖

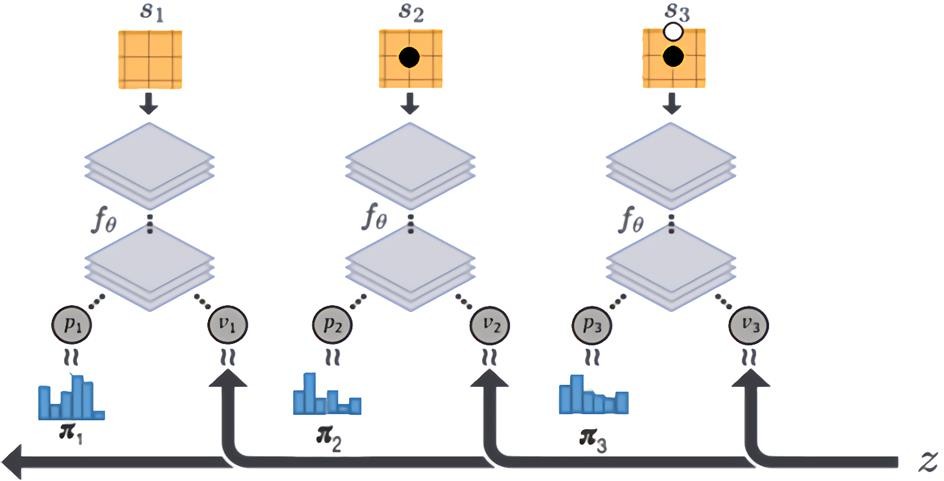
# Self-play reinforcement learning on Gomoku



## 策略價值網路訓練

* + - 機率 𝑝 接近 MCTS的機率𝜋
    - 評分 𝑣 接近實際結果 𝑧

Loss function [1]：



𝑙 = 𝑧 − 𝑣 2 − 𝝅𝑇𝑙𝑜𝑔𝒑 + 𝑐 𝜃 2

1. David Silver et al., ‘‘ Mastering the Game of Go without Human Knowledge’’,

# Self-play reinforcement learning on Gomoku



## 策略價值網路的評估方式

* + - 每50次的self-play，做一次評估
    - MCTS+新的策略價值網路 vs. MCTS + heuristic function
      * 每次評估對戰10局
      * 逐次增加 MCTS + heuristic function 的模擬次數，以提高其強度
  + 總結訓練過程
    - self-play 提供 data 訓練策略價值網路
    - 評估後，好的留下；壞的捨棄
    - self-play 重新產生新 data，構成訓練的循環

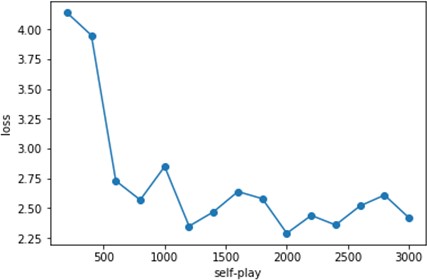
## 實驗設定

|  |  |
| --- | --- |
| **CPU** | **Intel I7 8700k** |
| GPU | GTX 1080 |
| RAM | DDR4 16GB |
| Chess board size | 9x9 |

* + 訓練時間
    - 3000 rounds in 3 days

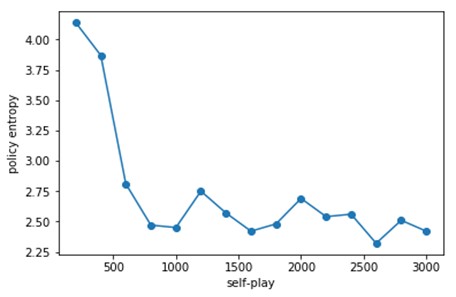
## Loss 函數變化

* + - 收斂速度較快



## 輸出落子機率分布(策略)的entropy變化

* + - 慢慢學會在不同的局面下哪些位置應該有更大的落子概率



14

### 實作Self-play reinforcement learning方法在五子棋對弈

* + - Monte Carlo Tree Search
    - Policy-Value Neural Network

### 修改策略價值網路評估方式

* + - 因為縮短了訓練時間，個人電腦也能實現Self-play reinforcement learning方法

### 實驗結果顯示本論文提出的修改方法不僅縮短了訓練時間，也確實強化對弈能力

* + 對於棋面邊角的位置較缺乏對抗性，未來將會這部分於Training改進與修改



Thank you for listening