

# 我預判你的預判：中學生與AI圍棋理論的對戰歷程

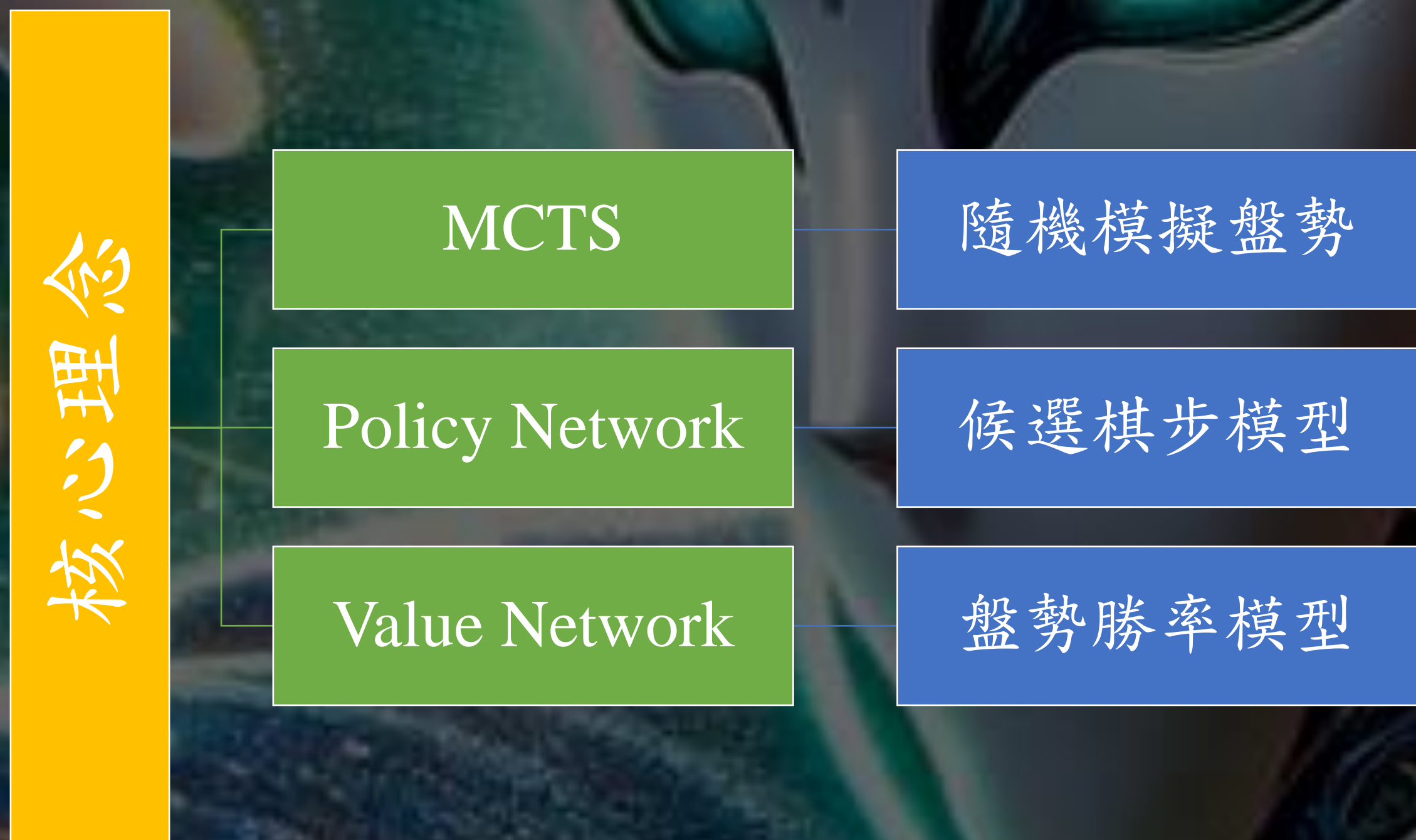
姓名: 黃鈺翔 學校: 成功高中  
EMAIL: tudo.huang@gmail.com

## 研究動機與目的:

- 八年前，AlphaGO橫空出世，擊敗了人類九段棋士。
- 八年後，高一的我，加入了電研社，從零開始了解程式、慢慢地往打造自己的圍棋機器人:SigmaGO前進。
- 想要希望向與會成員分享這段打造此機器人的實作歷程。

## 研究方法與過程:

- 研究書籍:《深度學習與圍棋》、AlphaGO論文



為了完成SigmaGO的資料收集，我使用了Pyaq這一開源專案的資料集，共34,572局的sgf檔案，並將其轉換為9x9矩陣，其中(1為白;-1為黑;0為空)

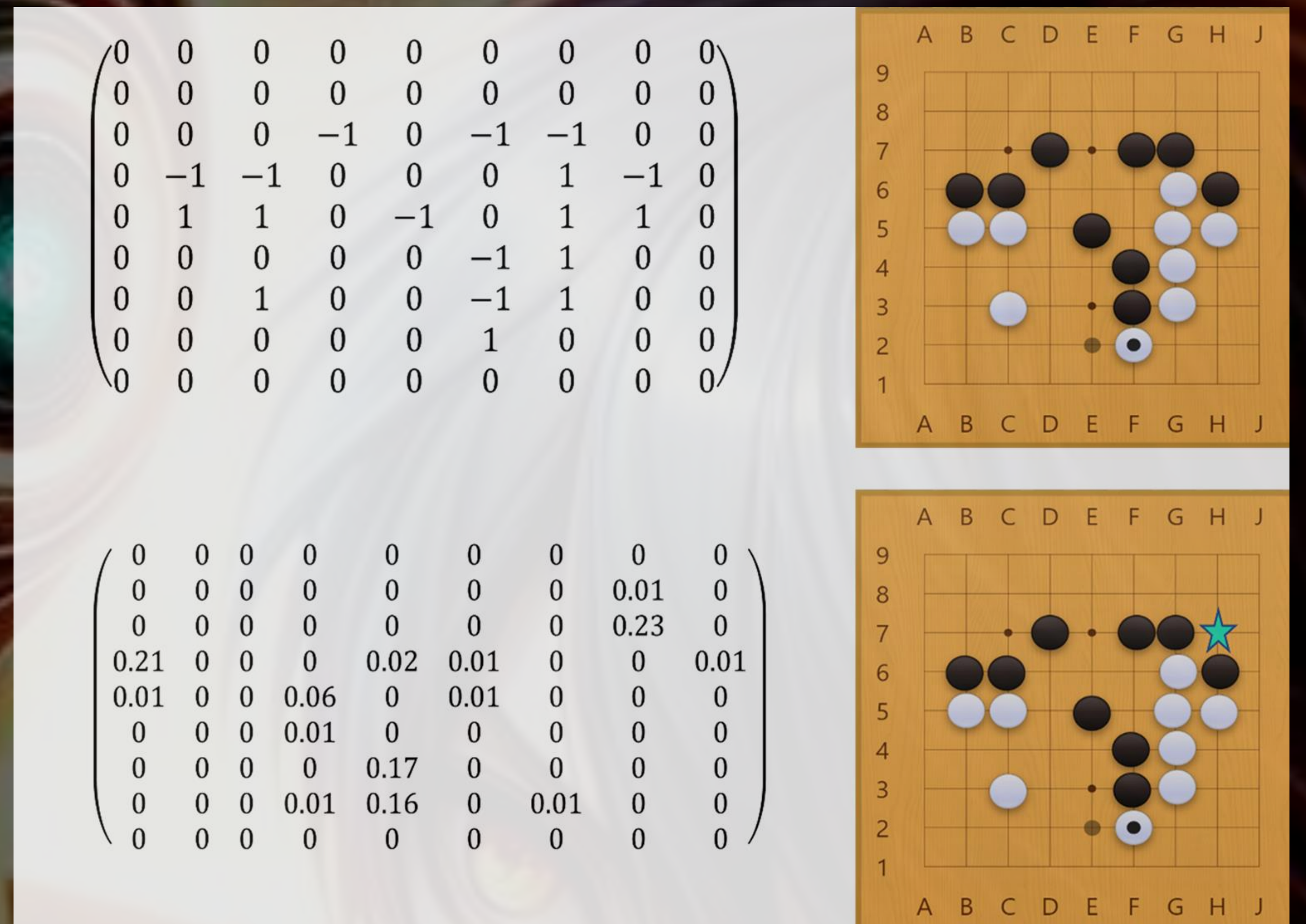
```
B[ee];W[cf];B[cd];
W[eg];B[fg];W[ef];
B[ff];W[fe];B[ed];
W[ge];B[eh];W[dh];
B[fh];W[fc]
```

```
[0., 0., 0., -1., 0., 0., -1., 0., 0.],
[0., 0., 0., 0., 0., 0., -1., 0., 0.],
[0., 1., 0., 0., 1., 0., 0., 0., 0.],
[0., 0., 0., 1., -1., 1., 0., 0., 0.],
[-1., 0., 0., 1., -1., -1., -1., 0., 0.],
[0., 0., 0., -1., 0., 0., 1., 0., 0.],
[0., 0., 0., 0., 0., 0., -1., 0., 0.],
[0., 0., 1., 0., 0., 0., 1., 0., 1.],
[0., 0., 0., 0., 0., 0., 1., 0., 0.]
```

接著，我用Pytorch訓練了一個卷積神經網路。這個模型的輸入是一個代表棋盤狀態的9x9矩陣，輸出則是一個長度為81的向量，表示在當前盤面下，每個位置下一步棋的可能性。下表為模型架構圖。

Layer(type)	Output Shape	Param #
Conv2d-1	32x9x9	320
Dropout-2	32x9x9	0
Conv2d-3	64x9x9	18,496
MaxPool2d-4	64x4x4	0
Dropout-5	64x4x4	0
Linear-6	1x128	131,200
Dropout-7	1x128	0
Linear-8	1x81	10,449

然而模型只是單純的預測，並不了解真正的規則。例如「打劫」以及「貼目」。我利用Gym\_Go模組，來得知給定的盤面中的合法棋步。合併這項資訊，系統得以找出滿足規則且機率最大的棋步。接著利用GTP串接圍棋使用者介面Sabaki，並利用圍棋程式GNUGo，來評估本系統的棋力。如下圖，綠星是表示機率顯示最高的位置(0.23)。



## 研究結果與討論:

我設置不同棋力的GNUGo (1~9)與SigmaGo的自動對戰。以SigmaGo執黑或執白與各下1000盤後求取平均勝率。如圖三所示，可以看出，隨著GNUGo的棋力調升，SigmaGo的勝率曲線顯示出下降趨勢。在Level 4到5之間，兩者的勝率接近。因此可以得知SigmaGo的棋力約為GNUGo的第四等級。而曲線呈現出一定程度的波動。我推論這個波動可能來自於對戰棋局數不足，或許能用更多的對戰棋局來評估棋力。



實際對局圖，GNUGo為黑贏2.5目；  
對戰不同棋力的GNUGo時SigmaGo的勝率變化圖

## 結論應用及展望:

隨著知識慢慢累積，不敢相信自己正漸漸接近那曾經遙不可及的AlphaGO。在本文撰寫的當下，我完成了Policy網路、使用者介面的串接以及棋力評估的統計。根據我的實驗結果，SigmaGo與GNUGo level 4的對戰棋力在伯仲之間。未來的研究中，我希望能將再實現MCTS法與Value網路，甚至引入強化式學習法，讓SigmaGo的能力，隨著這些實作過程漸漸成長，邁向資訊未來的偉大航道。