

Theory of Computer Games 2022 – Project 2+ report

109550031 李旻融

Network Design

所有可能的盤面組合太多了，所以為了計算每一種盤面的估計值，需要開一個表格來儲存，但這會使記憶體用量無法負荷，甚至有些盤面幾乎只會遇過一次，所以我們可以讓它學到小部分的特徵，而非一定要是曾經看過一模一樣的盤面，才知道那是什麼。對於 threes 的盤面來說，也能藉由學習特徵來讓相似的盤面能夠被預測出類似的潛力值。

在這個作業裡，我選擇沿用 project 2 的 features，也就是四個六格的 feature 加上兩個四格的 features，會選擇使用四格的 feature 是因為我想要避免每個 feature 過度相似。

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Improvement - Expectimax Search

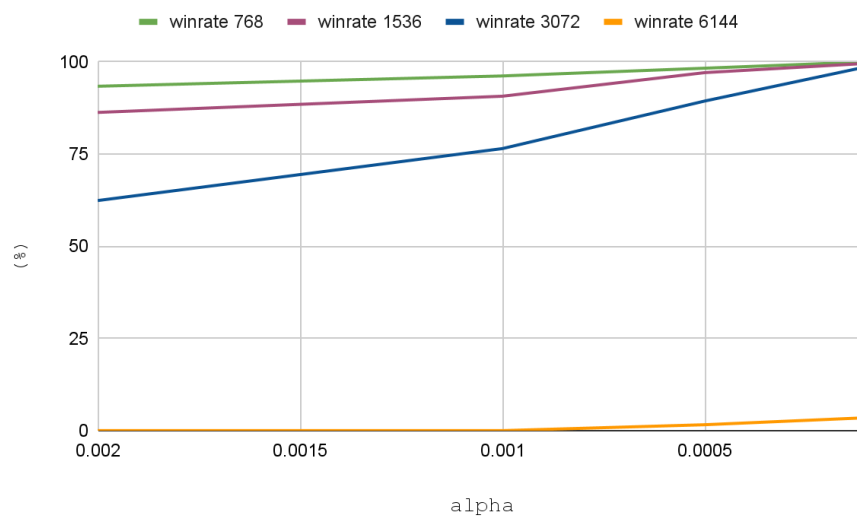
在估算每個盤面的潛力值時，以「多看一步」來取代用模型取值估計，由於系統會提示下一個方塊為何，所以只需將所有可能產生新方塊的位置列舉出來，接著計算出每種可能位置能賺最多分數下一步，並將所有可能位置得到的 reward 取平均，並決定這步該滑往哪個方向。在 Expectimax Search 中，會選擇對自己最有利的選項。接著，我在做 Expectimax Search 時，加上了一個參數 depth，可以不只多看一步，但因為搜尋的複雜度是呈指數上升的，如果在決定動作時搜尋愈多層，所花的時間就會愈多，所以最終我是使用兩層的 Expectimax Search 來訓練我的模型。

Training Process

- Expectimax Search(depth=1)

下方圖表為訓練過程，我共使用了四個 learning rate 進行十萬場的訓練。

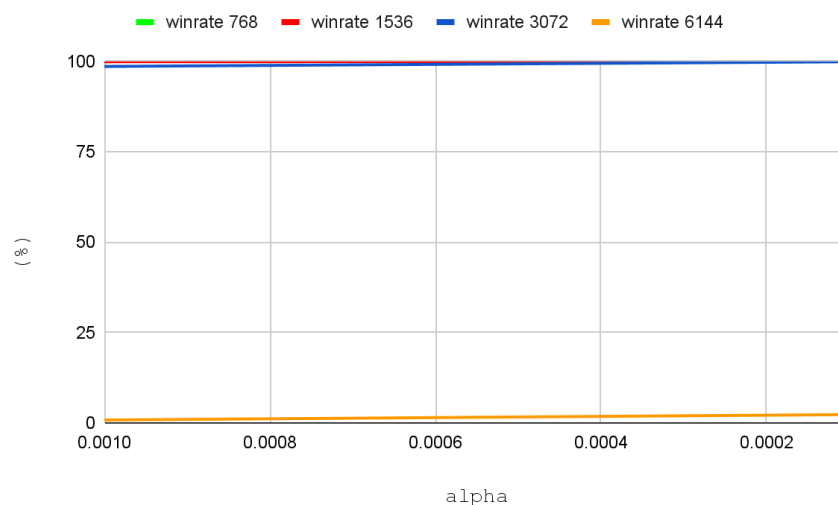
1. learning rate = $0.1/48 \doteq 0.002$ ，共訓練了三萬場。
2. learning rate = 0.001，共訓練了三萬場。
3. learning rate = 0.0005，共訓練了兩萬場。
4. learning rate = 0.0001，共訓練了兩萬場。



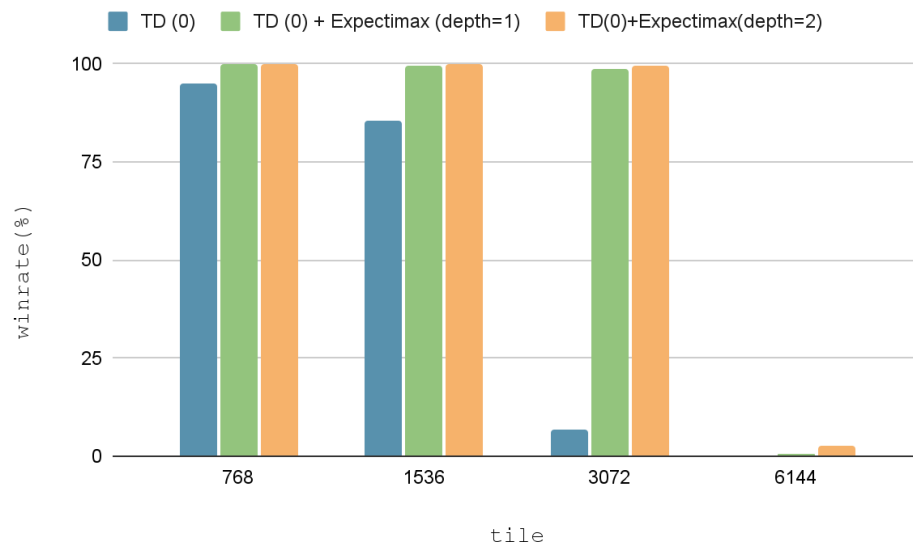
- Expectimax Search(depth=2)

下方圖表為訓練過程，我共使用了四個 learning rate 進行三萬場的訓練。

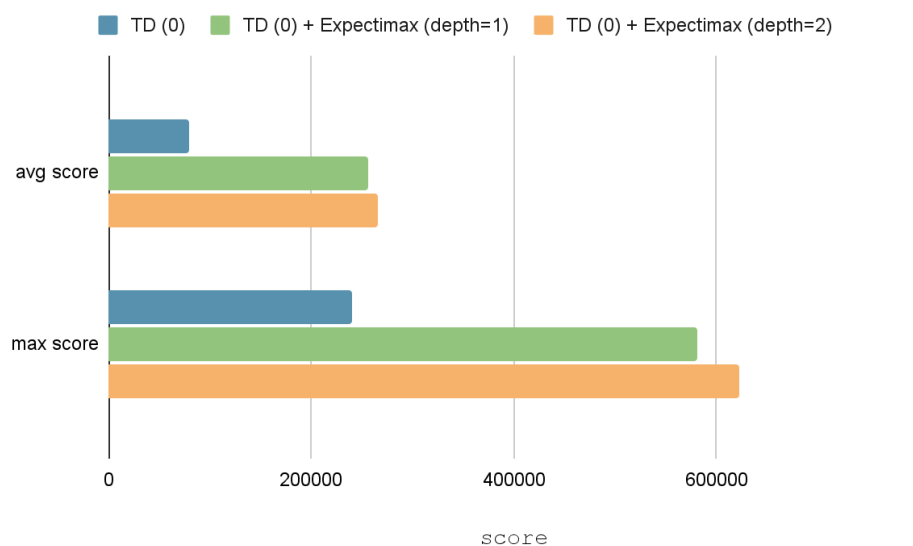
1. learning rate = 0.001，共訓練了一萬場。
2. learning rate = 0.0001，共訓練了兩萬場。



Performance



上方圖表為三個模型的比較圖，根據圖表可以發現有加上 Expectimax Search 的模型強度明顯大於沒有加上 Expectimax Search 的模型強度。接著比較一層的 Expectimax Search 和兩層的 Expectimax Search，兩者到達 tile 768、tile 1536、tile 3072 的 winrate 都快趨近於 100 %，但 tile 6144 的 winrate 卻只有不到 10 %，我認為可能的原因有兩個，一是模型本身不夠強，如果再加上一些較進階的 TD learning 方法，應該能提升 tile 6144 的 winrate，二可能是因為盤面過小，所以在到達 tile 6144 以前就已經沒位置了可以容納它了。



上方圖表為三個模型的分數比較圖，一樣可以發現有加上 Expectimax Search 的模型強度明顯大於沒有加上 Expectimax Search 的模型強度，兩層的 Expectimax Search 強度也有略高於一層的 Expectimax Search。