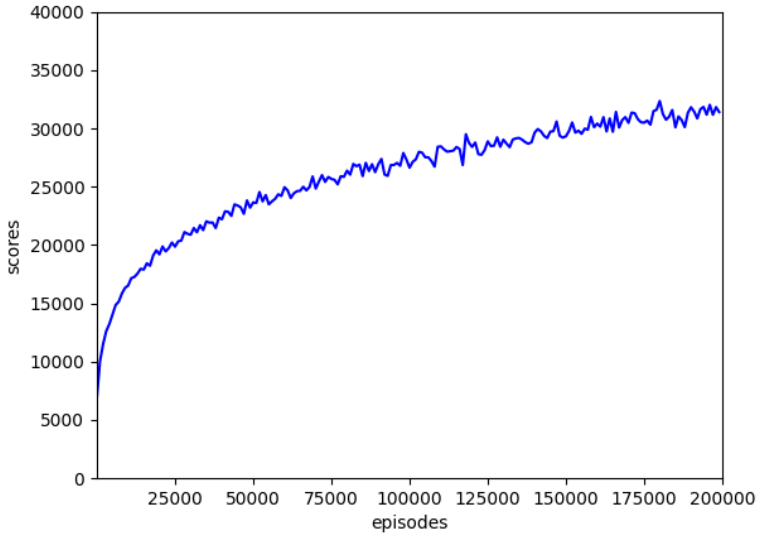

1. 使用 6 tuple network，共四個 feature，分別為{0,1,2,3,4,5}、{4,5,6,7,8,9}、{0,1,2,4,5,6}、{4,5,6,8,9,10}，皆以 class pattern 儲存。在 constructor 中 feature(1 << p.size() * 4) 建立 weight table，用來查詢 8 個 isomorphic 所代表的權重，查表時使用 indexof()獲得該六個位置(e.g 0,1,2,3,4,5)所對應的數值(e.g 0x012314)，再利用 operator[]()獲得該數值代表的權重。
2. TD(0)代表每一個 state 只考慮下一個狀態，也就是更新當下 state 的估計值時，只需要考慮 reward 和下一狀態的估計值，若是 TD(X)，表示考慮整個過程的 reward 和估計值。
3. 在 select_best_move()中，有上右下左四個動作用到 board b 上，每做完一個動作，就記錄他的 before state 和 after state 和 reward 在變數 move 中，經過 assign()確定該動作是有效的後，觀察 after state 的狀況，找出所有 2 或 4 可能會出現的情況，將這些情況的盤面做估計(e.g. 呼叫 estimate())，乘上對應的機率(e.g. 0.9 or 0.1)，所有估計值相加後的值和 reward 去更新該盤面的 value。上下左右都做完後，找出最大值就視為 best move。TD backup 是在跑完一個 episode 後，從倒數第二個狀態往回更新目標值，因為最後一個 state 已經是 final state，已沒有目標值，所以不須更新。從倒數第二個狀態開始，使用 reward + 下一狀態的目標值，減去此狀態原先的估計值，即可得到 TD error。因此將 error 和學習率相乘，去更新此狀態的估計值，如此做下去即完成更新一個 episode。