Reinforcement Learning　Lab1 report　　姓名:陳品翰

學號:313552042

Q:Describe the implementation and the usage of　n-tuple network

在這次的作業中，n-tuple 的實作是藉由取板面上固定位子(e.g (0,1,2,3,4))的值，並且把板面右旋轉、左旋轉、上下顛倒、左右顛倒等等方法，讓每一種 tuple 可以順利地得到同構(isomorphic)的板面，以利於減少 train 的複雜度。

每種 tuple 有 8 種同構，這個作業對每個板面有取 4 種不同的 tuple，每種 tuple 的同構會有自己的 weight 可以查表(1 種 tuple 因為有 8 個同構，所以有 8 個 weight table)，另外同構的板面所查到的期望值會是相同的(因為都是原本的版面做右旋轉、左旋轉、上下顛倒、左右顛倒)。

在做 update weight 的時候，會把要更新的值平均分給 4 個 tuple，而每個 tuple 因為有八個同構，所以會再平分成八份，用來更新各自的 weight。

當有板面傳進來時，會以 tuple 的 pattern 來取板面上相對的位子的值，之後做查表，並且把 8 種同構對於這個板面的值都相加，之後把每個 tuple 的期望值做相加，即可得該板面的期望值。

Q:Explain the mechanism of TD(0).

為了簡化描述，以下敘述的板面都為每個 state 的 before state。
TD Learning 的機制，以這個遊戲來舉例，一開始由於對於每個板面的期望值沒有數據，所以先讓系統隨意亂玩，先玩過一遍以後，可以得到每個板面做完一個動作後得到的分數(reward)，以及最後一個板面(定為 S')和最後的得分，我們可以從後往前看，倒數第二個板面(定為 S)的期望值應該更新為，S 的期望值 +alpha *(S 的 reward +(S'的期望值 - S 的期望值))，寫成數學式子的話為 V(S) ←V(S) + alpha * (S.reward + (V(S') – V(S)))，alpha 為 learning rate，概念上為如果 V(S')很高，則因為 S 可以做一個動作後得到 S'，所以 S 的期望值應該也要調高，以利於之後選擇動作時可以順利得到 S 板面，之後再做一個動作後得到 S' 板面。

簡單來說，如果 S'的期望值很高，代表之後玩遊戲時，要盡量做一些可以得到 S'板面的動作，而要得到 S'板面以前要先得到 S 板面，所以代表 S 板面的期望值也要調高，讓之後的遊戲可以先做些可以得到 S 板面的動作，得到 S 板面後，再做一個動作即可得到 S'，也就可以得到高分。

上述更新期望值的動作要從後往前做，所以要從最後一個板面一直更新到第一個板面，每次的更新都是以目前的板面和後一個板面的期望值做運算。

更新完成後，再進行下一輪的遊戲，直到之後每次玩遊戲時最後的分數都差不多，即為收斂的結果。

Q: Describe your implementation in detail including action selection and TD-backup diagram.

TD backup 的方式就如同上述所示，在這個作業中，以下一個的 state 的 before board(即為剛 pop 出數字的板面)的期望值(V(S'))，減掉目前 state 的 before board 的期望值(V(S))，之後再加上目前 state 做完 action 的 reward，整個的結果再乘上 learning rate(alpha)之後加上 V(S)，數學式子為:V(S)←V(S) + alpha * (reward + (V(S') – V(S)))，以此來更新每個 state 的 before board 的期望值。

Action selection 的部分，由於更新的是 before board 的期望值，又因為做完動作後 POP 的數字是在隨機的位子，且 2 和 4 的機率為 0.9 和 0.1，所以選擇的方法為:假設 before board 做完 action 後，先去計算出 POP 後所有可能的板面，且計算每個板面的期望值，之後相加，由此可以得到做完目前這個 action 後，所有可能的板面的期望值總和，比較上、下、左、右每個動作得到的期望值總和，選擇最高的，即為目前要選擇的動作。

下圖 X 軸為 episode，以 1000 為一個單位，Y 軸為 1000 個 episode 的 mean