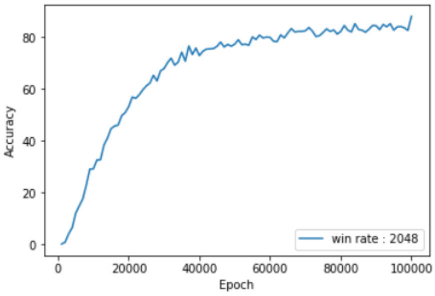
Lab7 : Temporal Difference Learning

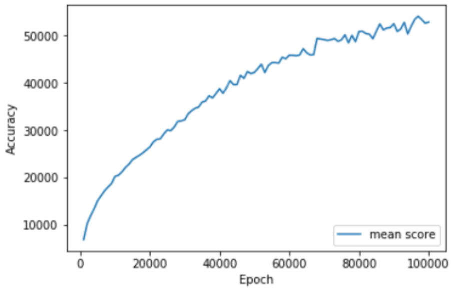
0516069 翁英傑

1. Score Plot :

Win rate of 2048 tile



mean score



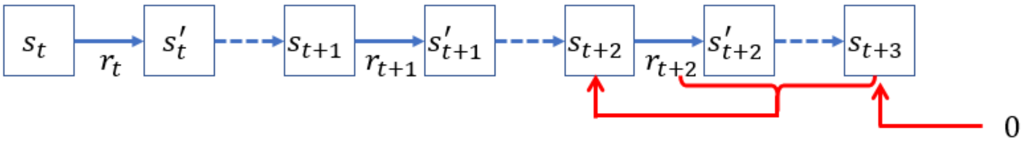
2. Mechanism of TD :

TD learning is to merge the distance of two state to the reward it gets with the action. The formula looks like this:

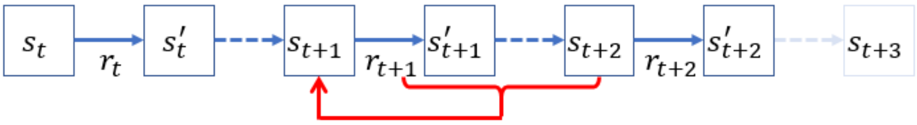
$$V(s) \leftarrow V(s) + \alpha(\overbrace{r + \gamma V(s')}^{\text{The TD target}} - V(s))$$

3. V(state) :

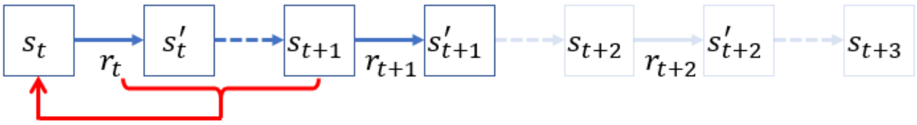
- Step 1: after game over (s_{t+3}), update the last state ($s^{\square}_{t+2}$)



- Step 2: update the previous *afterstate* ($s^{\square}_{t+1}$)

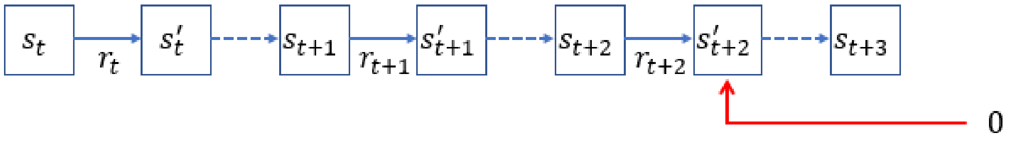


- Step 3: update the previous *afterstate* ($s^{\square}_{t}$)

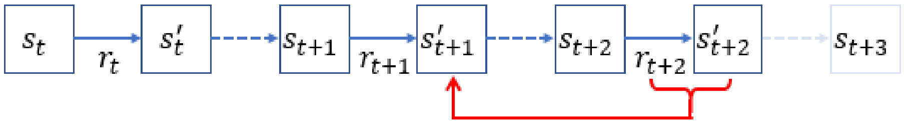


4. V(after-state) :

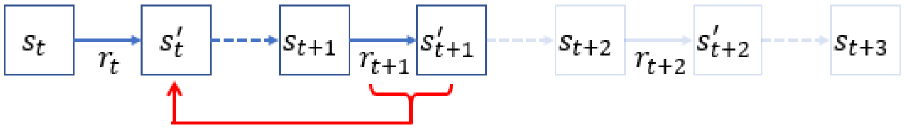
- Step 1: after game over (s_{t+3}), update the last state (s'_{t+2})



- Step 2: update the previous *afterstate* (s'_{t+1})



- Step 3: update the previous *afterstate* (s'_t)



5. Code :

The code is divided into five classes:

- Board: functions of moving the board with up, down, right and left four actions. Also contain the status of the board.
- Pattern: the function for computing the index for storing value of the board with certain pattern.
- Feature: a virtual class for pattern, has the same function as pattern.
- State: contains the information of each state in a play, which contains what action was taken, what reward was gained, what state turned to what state…etc.
- Learning: main methods of td-learning, including determine function for best action of each state, backward update function…etc.

A pseudocode of a game engine and training (modified backward training method)

```
function PLAY GAME
    $score \leftarrow 0$
    $s \leftarrow$ INITIALIZE GAME STATE
    while IS NOT TERMINAL STATE($s$) do
        $a \leftarrow \operatorname{argmax}_{a' \in A(s)}$ EVALUATE($s, a'$)
        $r, s', s'' \leftarrow$ MAKE MOVE($s, a$)
        SAVE RECORD($s, a, r, s', s''$)
        $score \leftarrow score + r$
        $s \leftarrow s''$
    for ($s, a, r, s', s''$) FROM TERMINAL DOWNTO INITIAL do
        LEARN EVALUATION($s, a, r, s', s''$)
    return $score$

function MAKE MOVE($s, a$)
    $s', r \leftarrow$ COMPUTE AFTERSTATE($s, a$)
    $s'' \leftarrow$ ADD RANDOM TILE($s'$)
    return ($r, s', s''$)
```

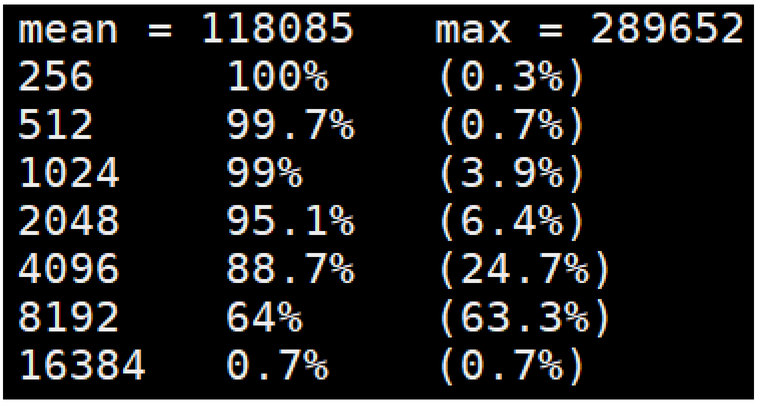
TD(0)-state

```
function EVALUATE($s, a$)
    $s', r \leftarrow$ COMPUTE AFTERSTATE($s, a$)
    $S'' \leftarrow$ ALL POSSIBLE NEXT STATES($s'$)
    return $r + \Sigma_{s'' \in S''} P(s, a, s'') V(s'')$

function LEARN EVALUATION($s, a, r, s', s''$)
    $V(s) \leftarrow V(s) + \alpha(r + V(s'') - V(s))$
```

TD(0)-afterstate

```
function EVALUATE($s, a$)
    $s', r \leftarrow$ COMPUTE AFTERSTATE($s, a$)
    return $r + V(s')$

function LEARN EVALUATION($s, a, r, s', s''$)
    $a_{next} \leftarrow \operatorname{argmax}_{a' \in A(s'')} EVALUATE(s'', a')$
    $s'_{next}, r_{next} \leftarrow COMPUTE\ \ AFTERSTATE(s'', a_{next})$
    $V(s') \leftarrow V(s') + \alpha(r_{next} + V(s'_{next}) - V(s'))$
```

6. Result



```
mean = 118085   max = 289652
256      100%     (0.3%)
512      99.7%    (0.7%)
1024     99%      (3.9%)
2048     95.1%    (6.4%)
4096     88.7%    (24.7%)
8192     64%      (63.3%)
16384    0.7%     (0.7%)
```

7. Discussion

We can see that the value of before-state contains a probability of each possible of after-state, this may cause the score estimation to add an extra variance and leads to a worse performance.