

Lab1: Temporal Difference Learning

Lab Objective:

In this lab, you will learn temporal difference learning (TD) algorithm by solving the 2048 game using an n -tuple network.

Important Date:

1. Submission Deadline: 9/21 (Sun) 23:59

2. Demo date: 9/22 (Mon)

Turn in:

1. Experiment report (.pdf)

2. Source code (.cpp) [NOT including model weights]

Notice: zip all files with name “RL_LAB1_StudentId_Name.zip”,
e.g.: 「RL_LAB1_313551126_李謙蓉.zip」

Lab Description:

- Understand the concept of (before-)state and after-state.
- Learn to construct and design an n -tuple network.
- Understand TD algorithm.
- Understand Q-learning network training.

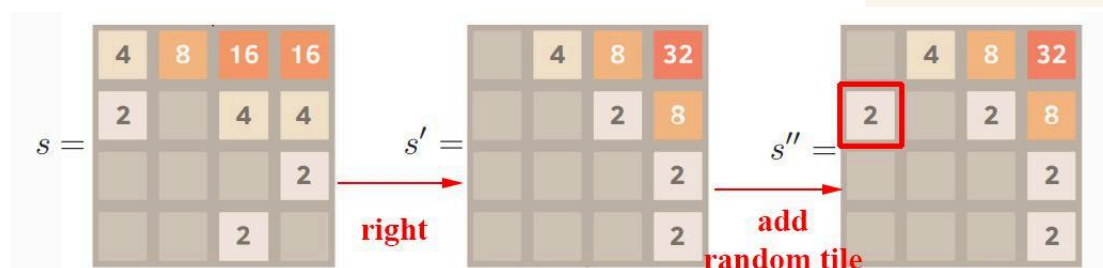
Requirements:

- Implement TD(0) algorithm
 - Construct an n -tuple network.
 - Action selection according to the n -tuple network.
 - Calculate TD-target and TD-error.
 - Update $V(\text{state})$, not $V(\text{after-state})$.
 - ◆ That is, you need to use the information of $P(\text{popup tile 2}) = 0.9$ and $P(\text{popup tile 4}) = 0.1$ ~~$P(\text{popup tile 2}) = 0.9$ and $P(\text{popup tile 4}) = 0.1$~~ and $P(\text{popup tile 2}) = 0.8$ and $P(\text{popup tile 4}) = 0.2$ in your code.
 - Understand temporal difference learning mechanisms.

Game Environment – 2048:

- Introduction: 2048 is a single-player sliding block puzzle game. The game's objective is to slide numbered tiles on a grid to combine them to create a tile with the number 2048.
- Actions: **Up, Down, Left, Right**
- Reward: The score is the value of new tile when two tiles are combined.
- A sample of two-step state transition

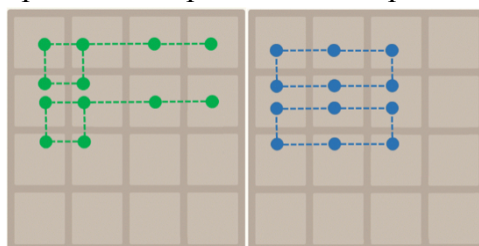
		2	4
		4	8
	2	16	32
	2	2	16



Implementation Details:

Network Architecture

- n -tuple patterns: 4×6-tuples with all possible isomorphisms



Training Arguments

- Learning rate: 0.1
 - Learning rate for features of n -tuple network with m features: $0.1 \div m$
- Train the network at least 100k episodes

Algorithm:**A pseudocode of the game engine and training.** (modified backward training method)

```

function PLAY GAME
  score ← 0
  s ← INITIALIZE GAME STATE
  while IS NOT TERMINAL STATE(s) do
    a ← EVALUATE(s, a')
    r, s', s'' ← MAKE MOVE(s, a)
    SAVE RECORD(s, a, r, s', s'')
  score ← score + r
  s ← s''
  for (s, a, r, s', s'') FROM TERMINAL DOWNT0 INITIAL do
    LEARN EVALUATION(s, a, r, s', s'')
  return score

function MAKE MOVE(s, a)
  s', r ← COMPUTE AFTERSTATE(s, a)
  s'' ← ADD RANDOM TILE(s')
  return (r, s', s'')

```

TD-state

```

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

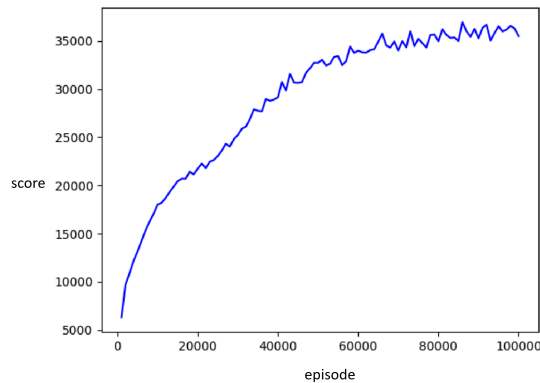
function LEARN EVALUATION(s, a, r, s', s'')
   $V(s) \leftarrow V(s) + \alpha(r + V(s'') - V(s))$ 

```

Scoring Criteria:

Show your work, otherwise no credit will be granted.

- Report (20% + Bonus 20%)
 - A plot shows scores (mean) of at least 100k training episodes (20%)
E.g.



- Bonus: (20%)
 - ◆ Describe the implementation and the usage of n -tuple network. (5%)
 - ◆ Explain the mechanism of TD(0). (5%)
 - ◆ Describe your implementation in detail including action selection and TD-backup diagram. (10%)
- Demo Performance (80%)
 - The 2048-tile win rate in 1000 games, $[winrate_{2048}]$. (60%)

E.g.

1000	mean = 21355.2	max = 64492
128	100%	(0.1%)
256	99.9%	(1.4%)
512	98.5%	(11.6%)
1024	86.9%	(51.2%)
2048	35.7%	(34.6%)
4096	1.1%	(1.1%)

- Questions. (20%)

References:

- [1] Szubert, Marcin, and Wojciech Jaśkowski. "Temporal difference learning of N-tuple networks for the game 2048." 2014 IEEE Conference on Computational Intelligence and Games. IEEE, 2014.
- [2] Kun-Hao Yeh, I-Chen Wu, Chu-Hsuan Hsueh, Chia-Chuan Chang, Chao-Chin Liang, and Han Chiang, Multi-Stage Temporal Difference Learning for 2048-like Games, accepted by IEEE Transactions on Computational Intelligence and AI in Games (SCI), doi: 10.1109/TCIAIG.2016.2593710, 2016.
- [3] Oka, Kazuto, and Kiminori Matsuzaki. "Systematic selection of n-tuple networks for 2048." International Conference on Computers and Games. Springer International Publishing, 2016.
- [4] moporgic. "Basic implementation of 2048 in Python." Retrieved from Github: <https://github.com/moporgic/2048-Demo-Python>.
- [5] moporgic. "Temporal Difference Learning for Game 2048 (Demo)." Retrieved from Github: <https://github.com/moporgic/TDL2048-Demo>.
- [6] lukewayne123. "2048-Framework" Retrieved from Github: <https://github.com/lukewayne123/2048-Framework>.