# LAB 2

TA 鄭紹雄

**Deadline: 2021/04/13(Tue) 12:00**

**Demo: 2021/04/13(Tue)**

In this lab,

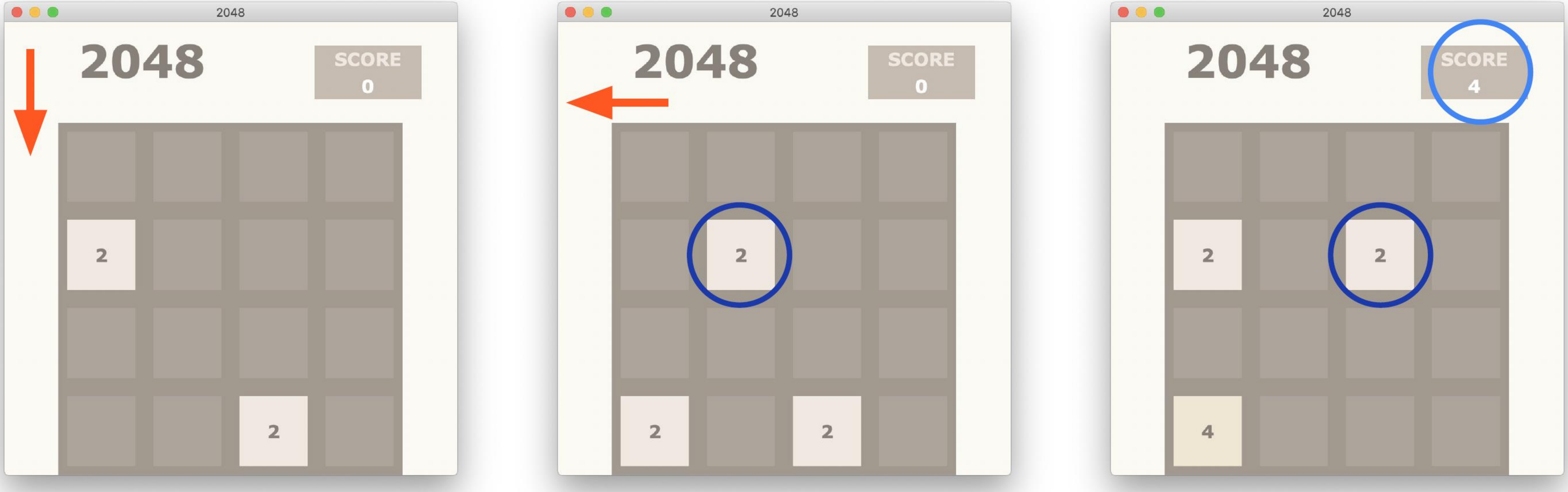
**Must use sample code, otherwise no credit.**

# Outline

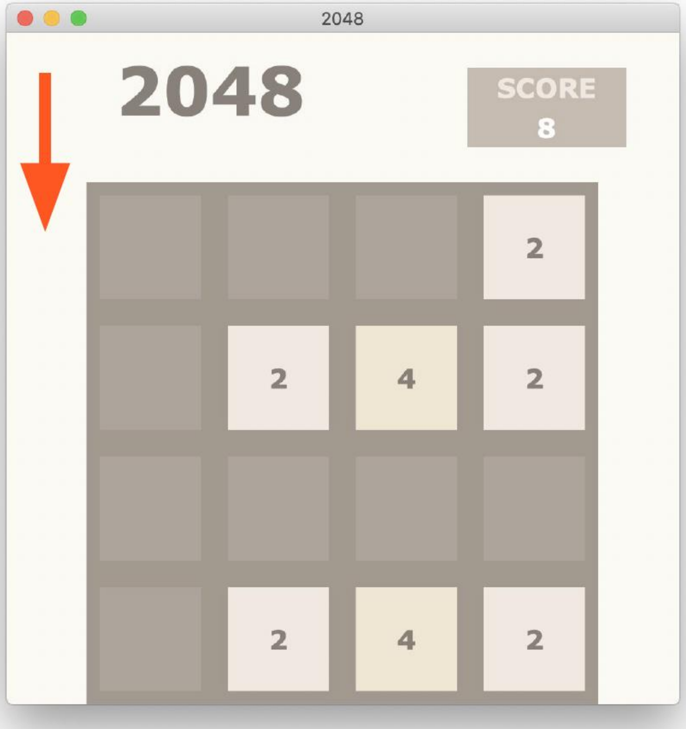
1. 2048 Game Rule
2. Game State
3. Temporal Difference Learning
4. n-tuple Network
5. Modify and Run Sample Code
6. Scoring Criteria
7. Reminders

# 2048 Game Rules (1/2)

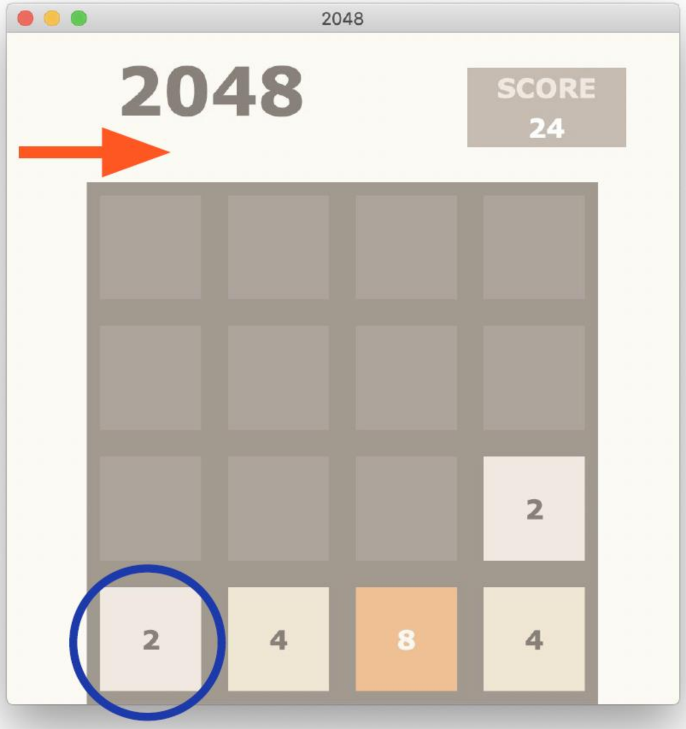
popup: **2** (90%), **4** (10%)



# 2048 Game Rules (2/2)

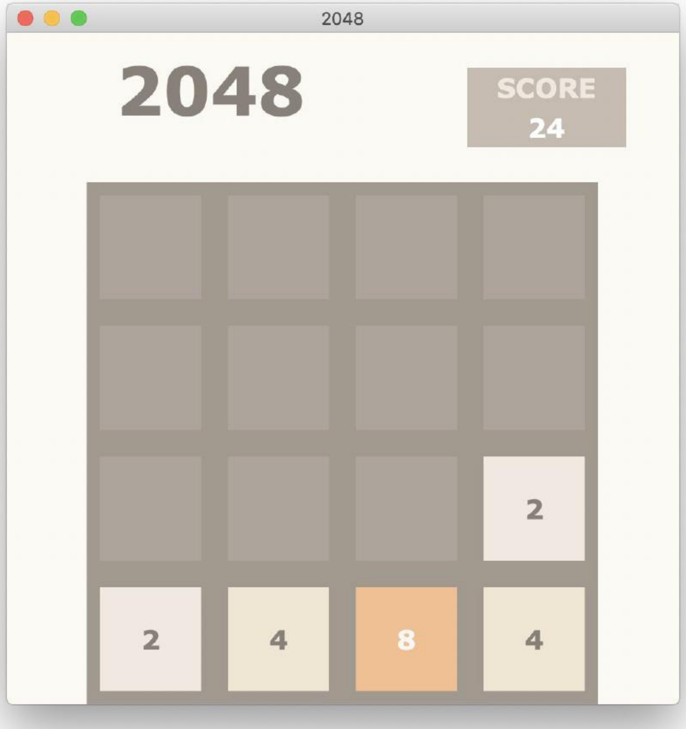


| | | | |
|---|---|---|---|
| | | | 2 |
| | 2 | 4 | 2 |
| | | | |
| | 2 | 4 | 2 |



| | | | |
|---|---|---|---|
| | | | |
| | | | |
| | | | 2 |
| 2 | 4 | 8 | 4 |

does not popup



| | | | |
|---|---|---|---|
| | | | |
| | | | |
| | | | 2 |
| 2 | 4 | 8 | 4 |

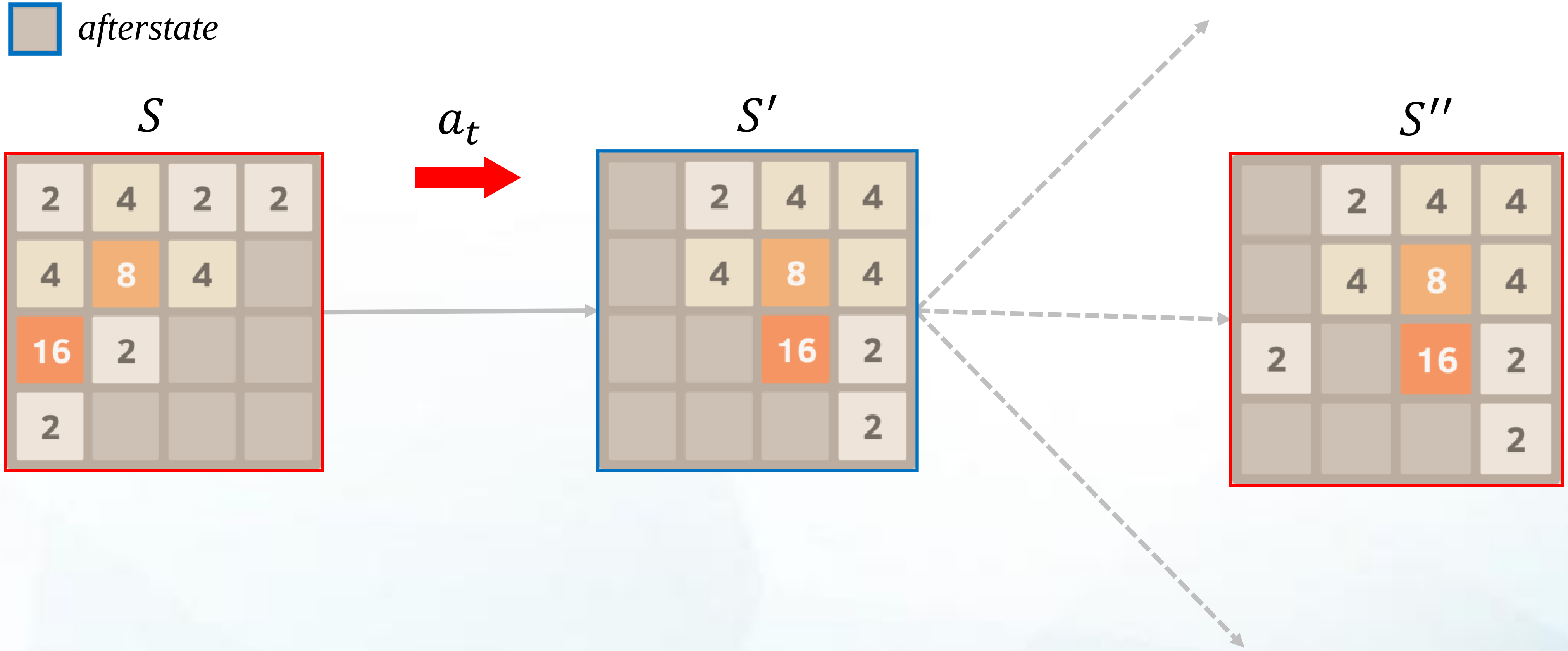
# Game State

→ perform action

⇢ popup a random tile

🟥 *beforestate*

🟦 *afterstate*



$S$

| | | | |
|---|---|---|---|
| 2 | 4 | 2 | 2 |
| 4 | 8 | 4 | |
| 16 | 2 | | |
| 2 | | | |

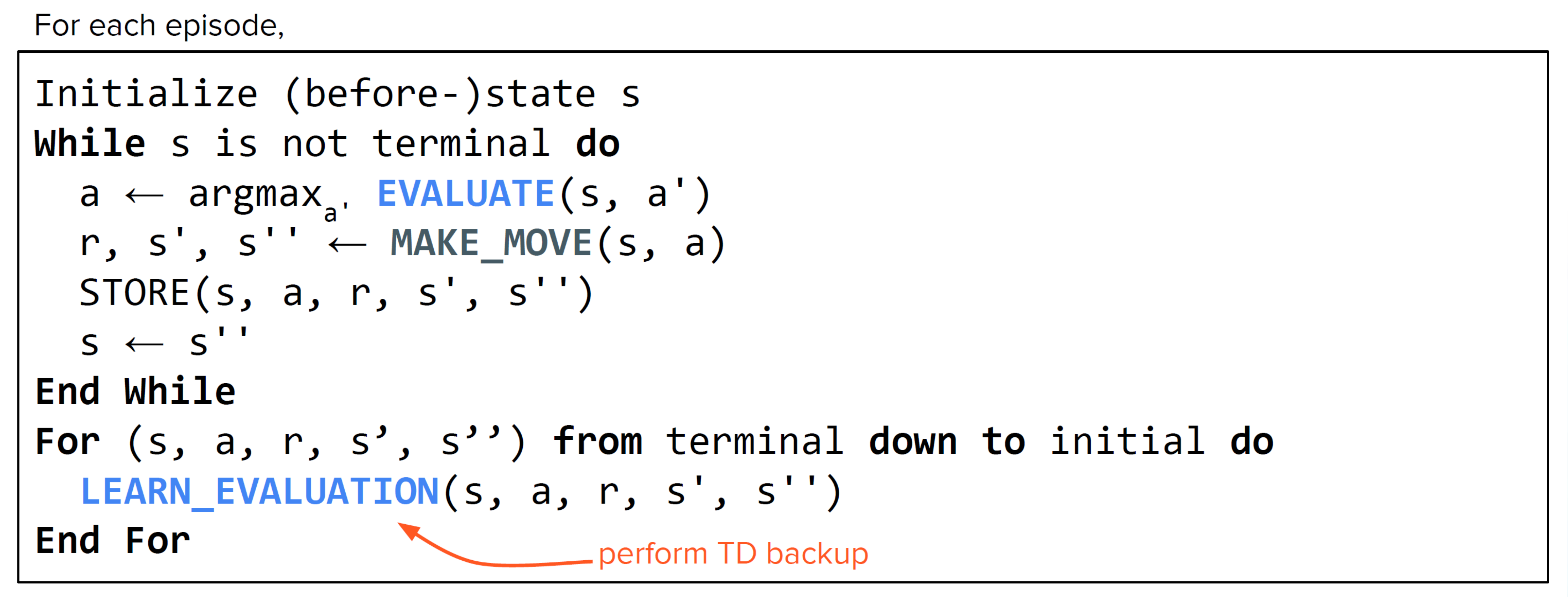
$S'$

| | | | |
|---|---|---|---|
| | 2 | 4 | 4 |
| | 4 | 8 | 4 |
| | | 16 | 2 |
| | | | 2 |

$S''$

| | | | |
|---|---|---|---|
| | 2 | 4 | 4 |
| | 4 | 8 | 4 |
| 2 | | 16 | 2 |
| | | | 2 |

# Temporal Difference Learning (TD)



For each episode,

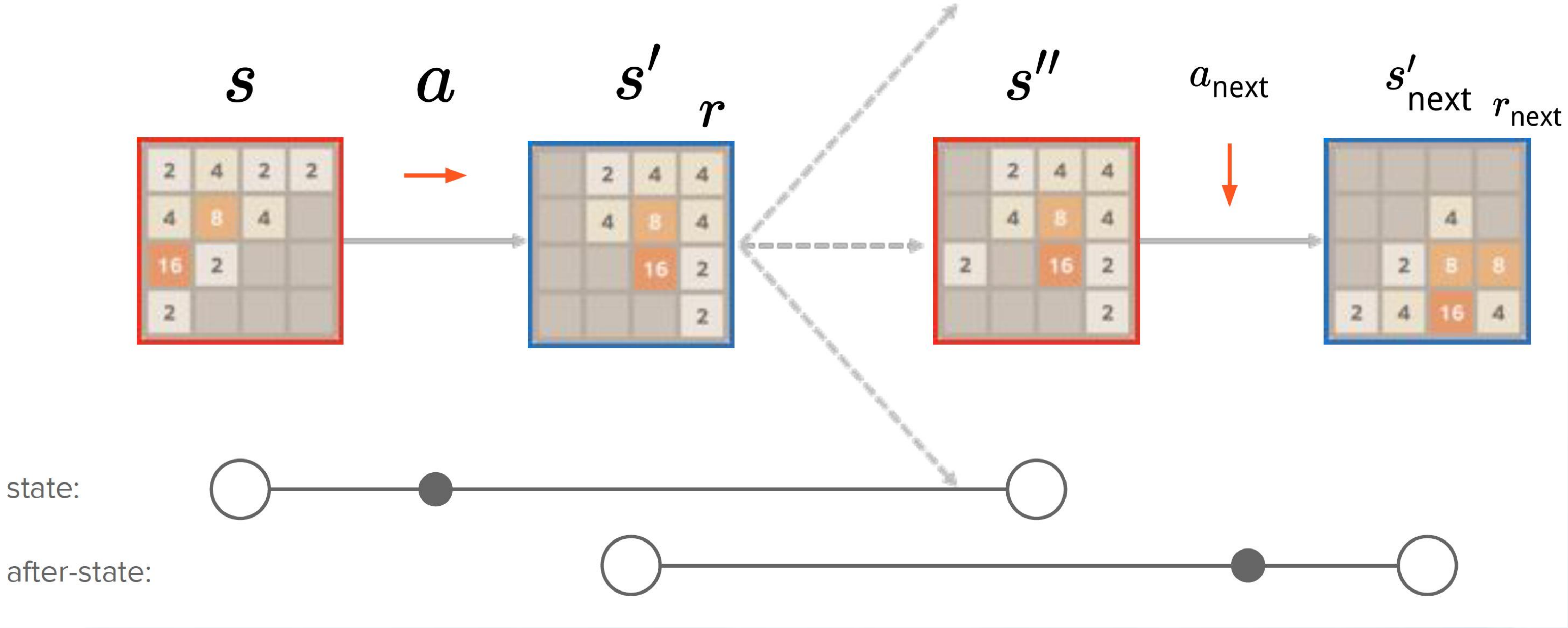
```
Initialize (before-)state s
While s is not terminal do
   a ← $\text{argmax}_{a'}$ EVALUATE(s, a')
   r, s', s'' ← MAKE_MOVE(s, a)
   STORE(s, a, r, s', s'')
   s ← s''
End While
For (s, a, r, s’, s’’) from terminal down to initial do
   LEARN_EVALUATION(s, a, r, s', s'')
End For
```

perform TD backup

# TD Backup Diagram

🟥 *beforestate*

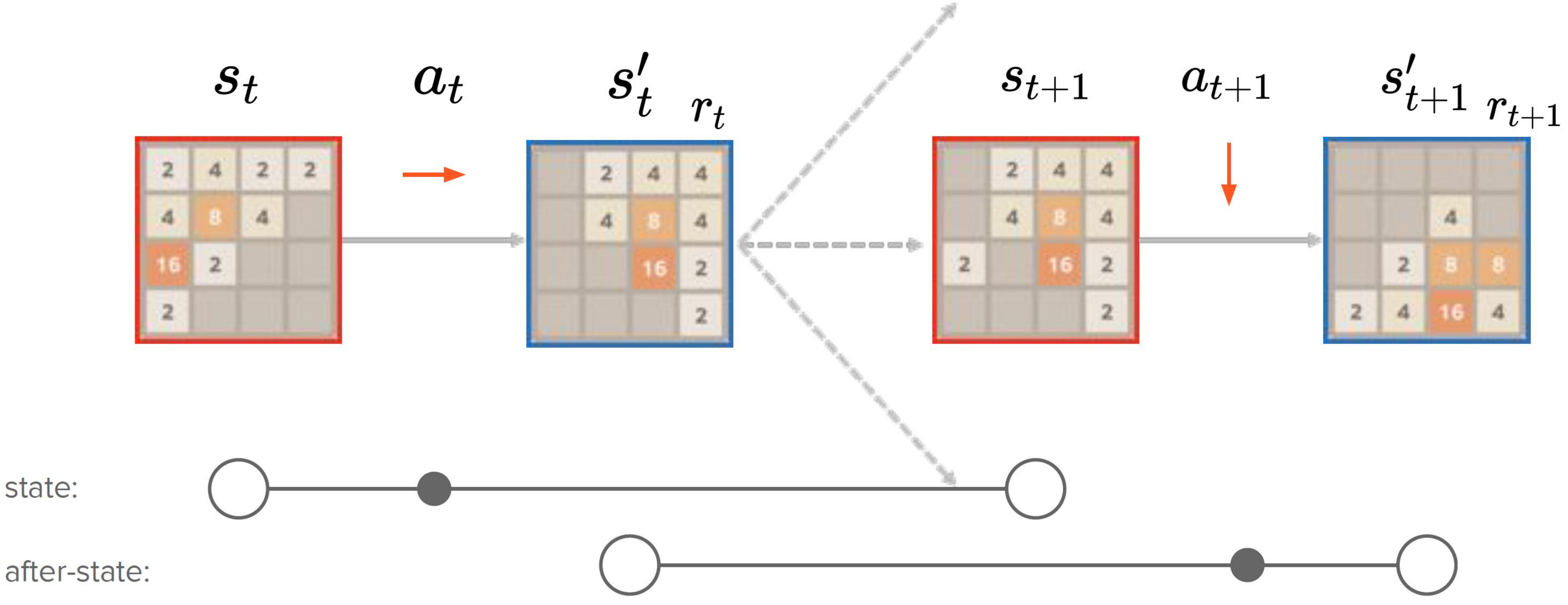
🟦 *afterstate*



# TD Backup Diagram

🟥 *beforestate*

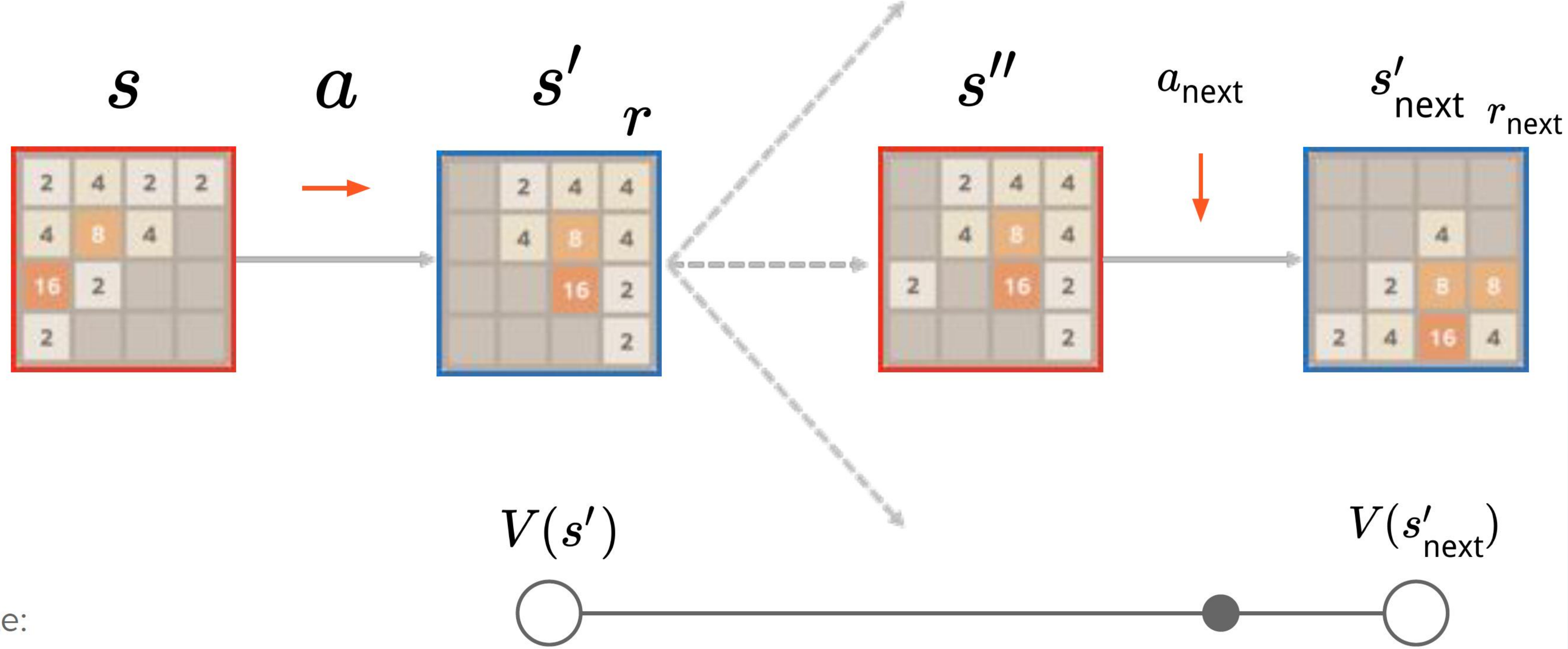
🟦 *afterstate*



# TD Backup: After-State

🟥 *beforestate*

🟦 *afterstate*



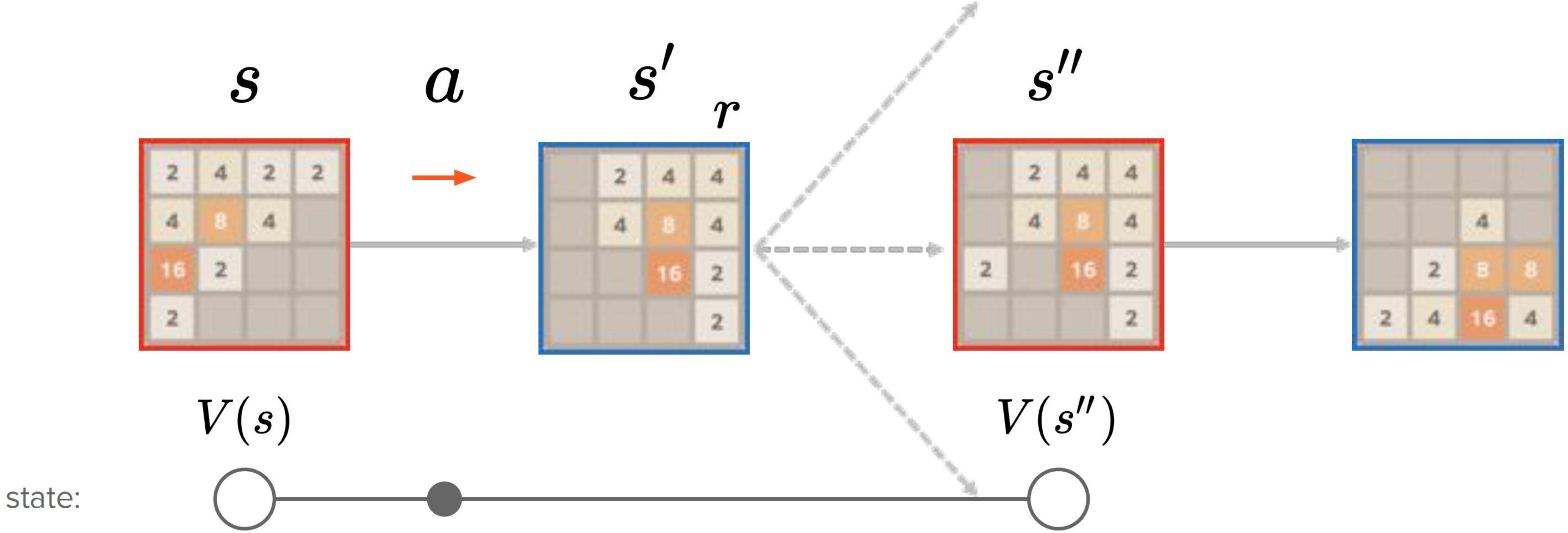
after-state:

$$V(s') \leftarrow V(s') + \alpha(r_{\text{next}} + V(s'_{\text{next}}) - V(s'))$$

# TD Backup: State

🟥 *beforestate*

🟦 *afterstate*



$$V(s) \leftarrow V(s) + \alpha(r + V(s'') - V(s))$$

# Why use n-tuple network?

- The expected score/return $G\_t$ from a board $S$

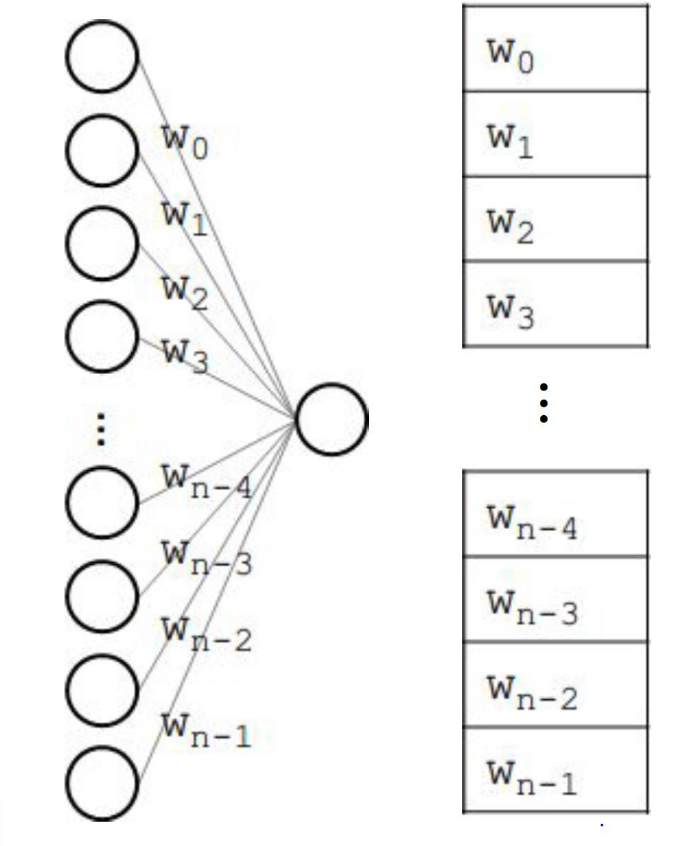
- But, #states is huge
  - About $17^{16}$ (=$10^{20}$).
    - Empty (→ 0), 2 (=$2^1$ → 1), 4 (=$2^2$ → 2), 8 (=$2^3$ → 3), …, 65536 (=$2^{16}$ → 16).

- Need to use value function approximator.

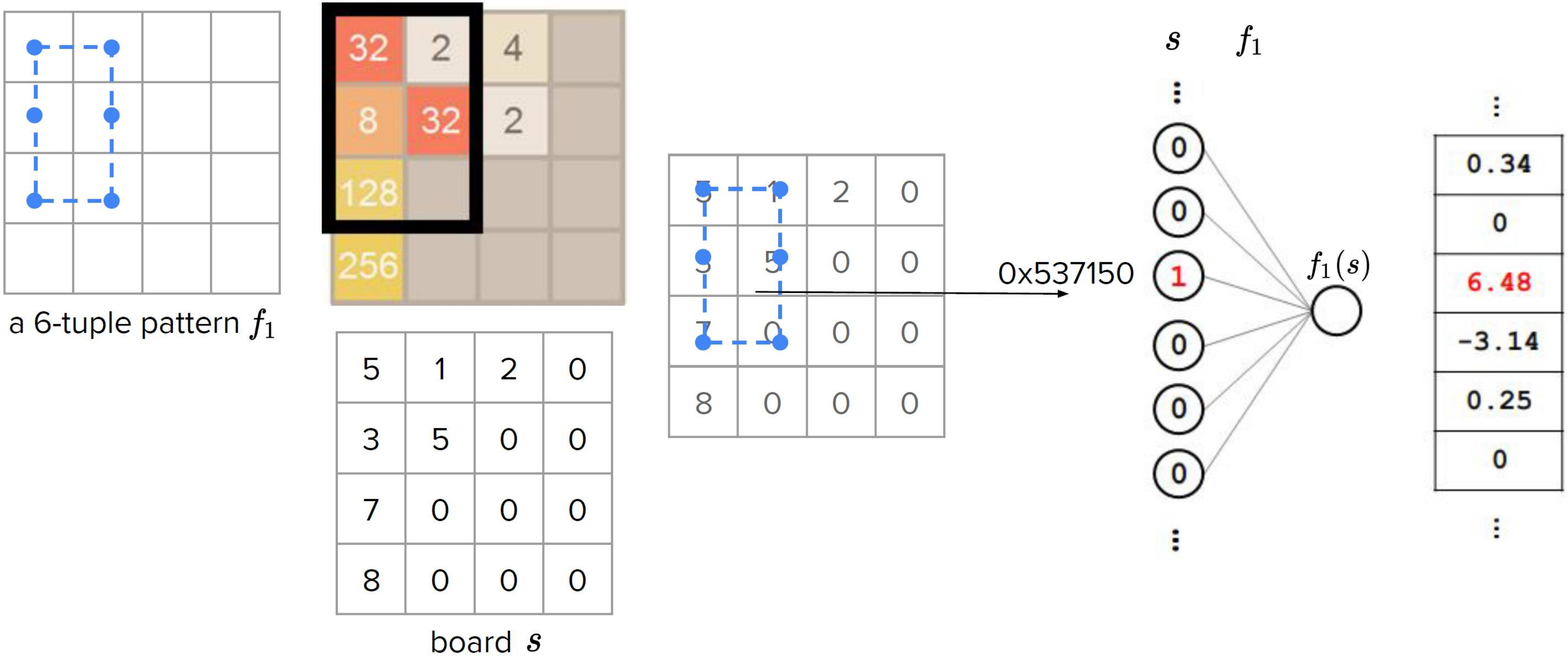
# n-tuple network

n-tuple network (a.k.a. RAM-based neural network) is a type of artificial neural network.

- A large number of input nodes.
  - Input values are either 1 or 0.
  - Input is a sparse vector.
- No hidden layers.
- Only 1 output node.



# Example: 2048 with n-tuple network



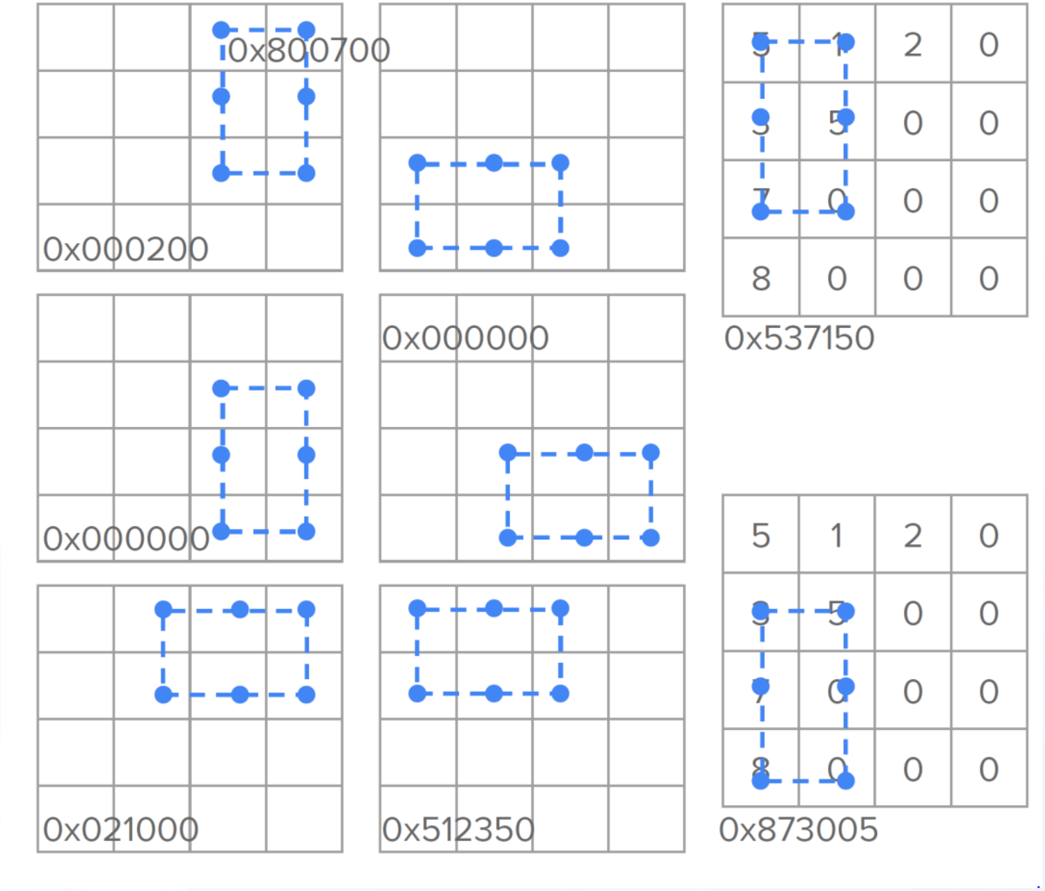
a 6-tuple pattern $f_1$

| | | | |
|---|---|---|---|
| 5 | 1 | 2 | 0 |
| 3 | 5 | 0 | 0 |
| 7 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 |

board $s$

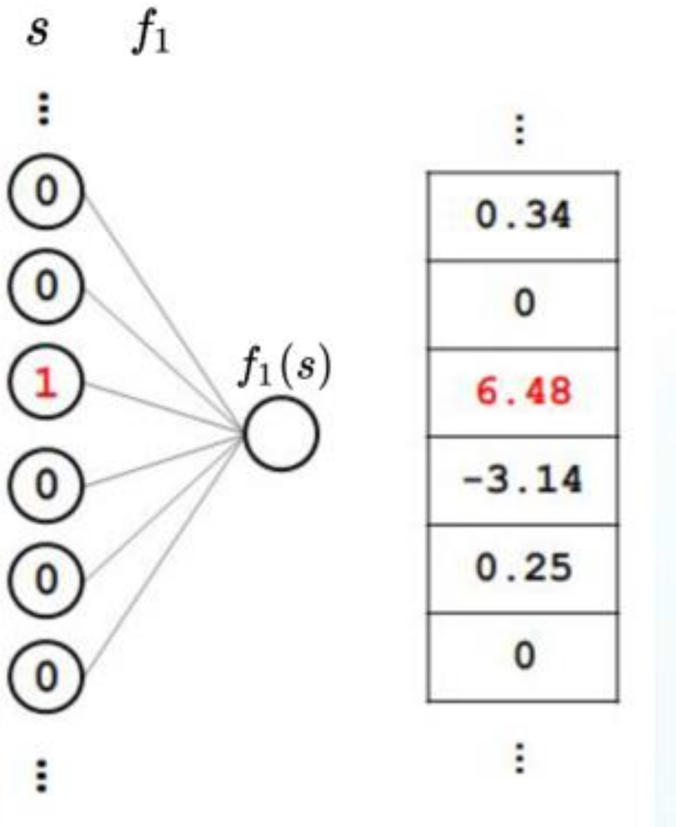
# All Isomorphism

- Rotations and Reflections
- The sum of the eight values can represents the board.



| | | | |
|---|---|---|---|
| 5 | 1 | 2 | 0 |
| 3 | 5 | 0 | 0 |
| 7 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 |

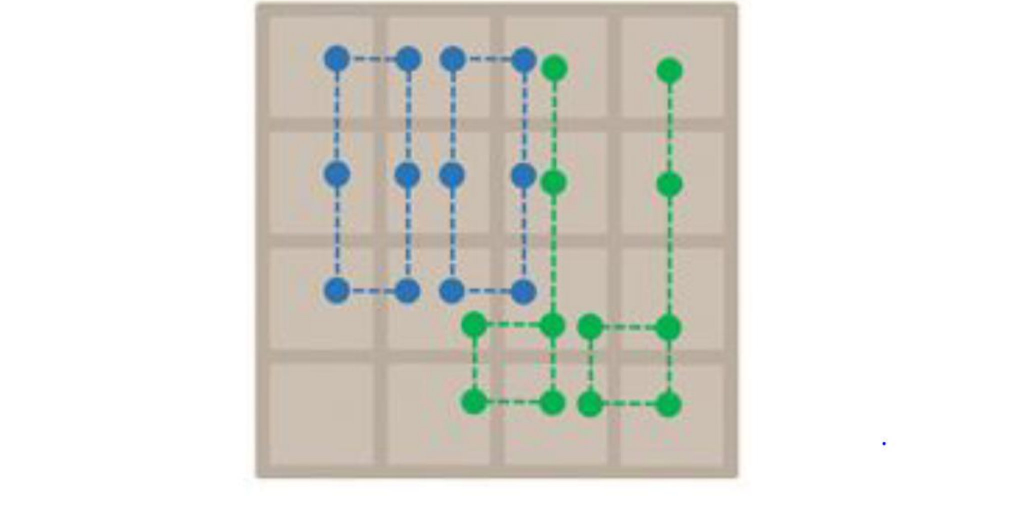
| | | | |
|---|---|---|---|
| 5 | 1 | 2 | 0 |
| 3 | 5 | 0 | 0 |
| 7 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 |

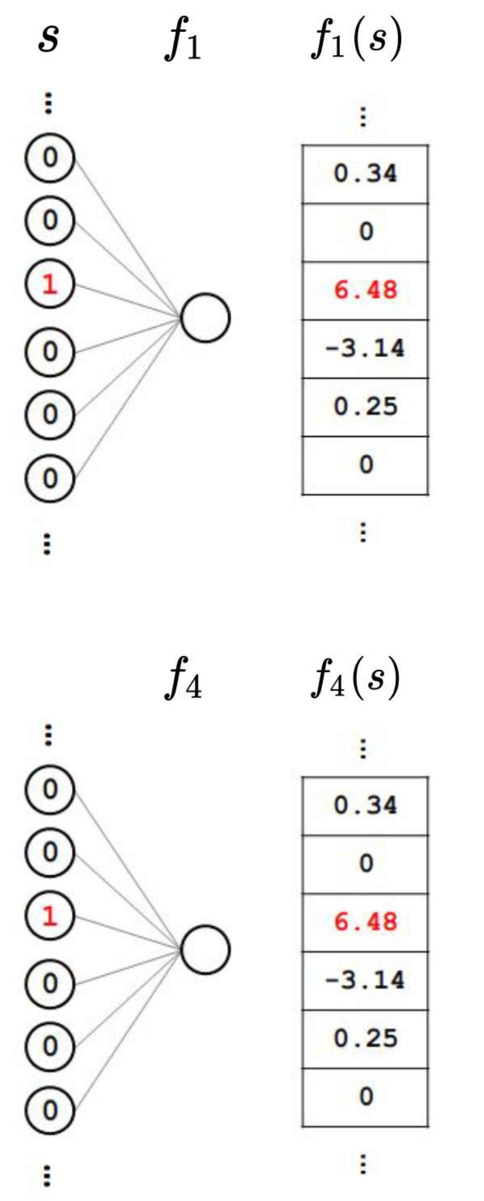


# Multiple n-tuple

- Example: 4 kinds of 6-tuple.
- Calculate (use int):
  - Size: 4* $15^6$ * 4 byte

$$V(s) = f_1(s) + f_2(s) + f_3(s) + f_4(s)$$





# Sample Code

- Implement V(state)
  - Compile with C++11 support
  - ex: g++ -std=c++11 -O3 -o 2048 2048.cpp
- Run and Train

# Scoring Criteria

Show your work, otherwise no credit will be granted.

- Report (60%)
  - (DO explain; do not only copy and paste your codes.)
- Performance (40%)
  - The 2048-tile win rate in 1000 games,$\lceil \mathrm{winrate}_{2048} \rceil$.(20%)
  - Questions. (20%)

# Reminders

- You can design your n-tuple.
- You should avoid using CNN in this lab.
- 2048-tile should appear within 10,000 episodes.

# 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