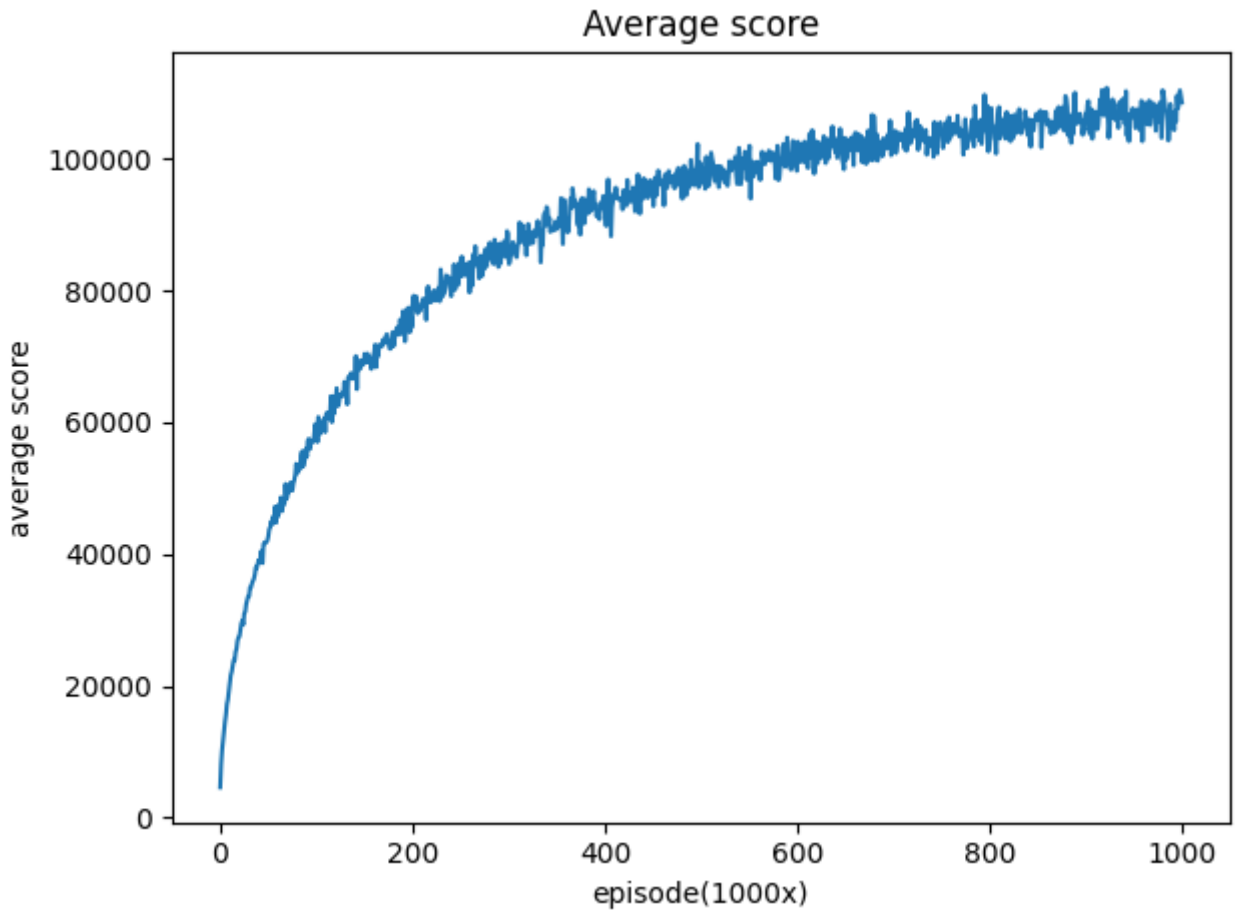
RL Lab1- 2048 TD

313551081 楊璨聰

Report

A plot shows scores (mean) of at least 100k training episodes (20%)



This figure shows average score for 1000k training episodes.

```
1000    mean = 100812   max = 288056
        128     100%    (0.3%)
        256     99.7%   (0.1%)
        512     99.6%   (1.3%)
        1024    98.3%   (3.7%)
        2048    94.6%   (17.8%)
        4096    76.8%   (27.2%)
        8192    49.6%   (49.1%)
        16384   0.5%    (0.5%)
```

This figure show the percentage above 2048 is **96.4%**

Bonus

Describe the implementation and the usage of n-tuple network. (5%)

Implementation and Usage of (n)-tuple Network

Implementation:

1. **Initialization**: Define the number of tuples (n) and initialize weights.
2. **Tuple Selection**: Select (n) tuples from the input space.
3. **Feature Extraction**: Extract features for each tuple and form unique indices.
4. **Weight Update**: Update weights based on learning rules like TD(0).

Usage:

1. **Reinforcement Learning**: Approximate value functions.
2. **Pattern Recognition**: Recognize patterns in large, sparse input spaces.
3. **Game Playing**: Evaluate board states and make decisions in games like 2048.

Explain the mechanism of TD(0). (5%)

TD(0) is a reinforcement learning algorithm that updates the value function incrementally after each action based on the observed reward and the estimated value of the next state. It calculates the temporal difference error as the difference between the current estimate and the actual reward, then updates the current state value accordingly. TD(0) combines elements of Monte Carlo methods and dynamic programming, offering efficient online learning through continuous updates during each episode. In the context of the 2048 game, it learns better board state values by playing and improving through many iterations.

Describe your implementation in detail including action selection and TD backup diagram. (10%)

- Action selection: `select_best_move()` select best move after evaluated moving up, down, left and right. When evaluating board, estimate `P(popup tile 2) = 0.9` and `P(popup tile 4) = 0.1` then calculate expected value of board S'' as estimate value of board S.
- TD backup diagram: The TD backup diagram corresponds to the update_episode method within the Learning class. This method is invoked after each episode of the game has been completed, and it adjusts the feature weights in accordance with the TD(0) update rule.

  The updates are applied in reverse order along the episode path. For each state, the temporal difference (TD) error is computed, and the corresponding feature weights are updated accordingly.