Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/18/20230996

A review on the application of reinforcement learning to the 2048 game

Hongyu Mi

The University of Hong Kong, Hong Kong, 999077, China

u3575638@connect.hku.hk

Abstract. In light of the rapid development of Artificial Intelligence (AI), this paper pries into the application of reinforcement learning on the 2048 game which has important implications on the algorithm's ability to solve various real-life problems including but not limited to robotics, auto-driving, and financial trading. The paper summarizes the possible usage of the 2048 model in Q-learning, convolutional neural networks, genetic, and Monte Carlo tree search algorithms. Then the paper scopes into the current progress and methods used in solving the 2048 game with reinforcement learning. Starting off with Q-learning, its implementation and limitations, such as large state space, sparse rewards, limited exploration, computational complexity, and a lack of human-like intuition, are illustrated. Deep Q-learning handles the problem of large state space from Q-learning. However, most of the disadvantages that are found in Q-learning are shared by Deep Q-learning and hence should become the focus of future research on using reinforcement learning to solve 2048.

Keywords: reinforcement learning, 2048, Q-learning, deep Q-network.

1. Introduction

The 2048 game is a popular tile-matching puzzle game, the goal of which is to merge tiles with the same number to get a tile with a greater value by sliding in either of the four directions. The game requires strategic thinking and planning to reach a high score. By adopting reinforcement learning to play the game, researchers have typically applied deep neural networks to approximate the agent's policy, mapping from game states to actions. Then reinforcement learning algorithms such as Q-learning and gradient methods are used to train the agent. This research aims to summarize the current progress in using reinforcement learning to play the 2048 game and conclude whether the improvements made are effective or if further research is needed. The paper has important implications for the real-life applications of reinforcement learning algorithms provided that the methods used to train the agent can be adopted in real-life problems such as robotics, auto-driving, and financial trading.

2. Applications of the 2048 model

The 2048 game has been used as a benchmark and testing ground for various AI algorithms including but not limited to reinforcement learning, neural networks, evolutionary algorithms, and search algorithms. Some examples are given in this section.

2.1. Q-learning algorithm
Reinforcement learning is a sub-field of machine learning that involves an agent learning from its environment through trial and error. The game 2048 is a natural fit for reinforcement learning since it has a simple set of rules and a clear objective. In the study by Szubert et al. [1], the Q-learning algorithm was used to train an agent to play 2048. Q-learning is a model-free reinforcement learning algorithm that learns an optimal action-value function by iteratively updating its estimates based on the rewards obtained from the environment. The researchers achieved high scores in the game by applying Q-learning to 2048.

2.2. Convolutional neural networks
Neural networks are a class of machine learning algorithms that are loosely based on the structure and function of the human brain. They can study complex patterns and relationships from data. In the study by Kondo et al. [2], a convolutional neural network (CNN) was trained to play 2048. A CNN is a type of neural network that is particularly effective at processing images and other multidimensional data. The researchers used the game board as input to CNN and trained it to predict the next move in the game. CNN was able to achieve high scores in 2048, demonstrating the effectiveness of neural networks in solving the game.

2.3. Monte carlo tree search
Search algorithms are a class of algorithms that explore a search space to find a solution to a problem. In the study by Rodgers et al. [3], various search algorithms were compared to solve the game of 2048. They tested several algorithms, including Monte Carlo Tree Search (MCTS), which is a heuristic search algorithm that uses random simulations to guide the search. They found that MCTS was the most effective algorithm for solving the game, outperforming other algorithms such as minimax and alpha-beta pruning.

These examples demonstrate the versatility of the game of 2048 as a testing ground for various AI algorithms. Researchers have applied a range of techniques, including reinforcement learning, neural networks, evolutionary algorithms, and search algorithms, to solve the game and achieve high scores. Hence, attempts to improve the training methods for the 2048 game will serve as better foundations for various fields of study in AI.

3. Reinforcement learning applied in the game 2048

3.1. Q-Learning and its limitations
Q-learning is a popular reinforcement learning algorithm that has been applied to various problems, including the game 2048. The basic idea of Q-learning is to learn a function Q(s,a), which estimates the expected future reward of acting as in states. The algorithm updates the Q-values based on the rewards obtained from the environment and the maximum Q-value of the next state. Over time, the Q-values converge to the optimal Q-values, which represent the maximum expected future reward for each state-action pair.

One approach to using Q-learning to solve the 2048 game is to represent the state of the game as a vector of features, such as the tile values, their positions, and their distances to the edges and corners. The action space consists of four directions (up, down, left, and right), and the reward is the score obtained from merging tiles.

Szubert et al. [1] applied Q-learning to the game of 2048 using a feature-based representation of the state. They used a variant of Q-learning called State Action-Reward-State-Action (SARSA), which updates the Q-values based on the next state-action pair taken by the agent, rather than the maximum Q-value of the next state. The researchers used a decaying epsilon-greedy policy, that is, a random action with probability epsilon and the maximum Q-value action otherwise, to balance exploration and exploitation.

Another study by Kaundinya et al. [4] also applied Q-learning to the 2048 game using a CNN to represent the state. CNN took a 4x4x16 matrix as input, where each channel represents the log2 value of the tiles. The Q-values were estimated using a fully connected layer followed by a softmax function, which converts the Q-values into a probability distribution over the actions. The researchers used a decaying epsilon-greedy policy and experience replay to improve the stability of the Q-learning algorithm.

The results showed that their Q-learning agent was able to achieve high scores in 2048, outperforming a random agent and achieving scores comparable to those of human players. The researchers also compared their approach to other reinforcement learning algorithms, such as SARSA with eligibility traces and temporal difference (TD) learning, and found that their approach performed the best.

However, there are several limitations to using Q-learning to solve the game of 2048, and they have been discussed in several studies.

3.1.1. Large state space. The game of 2048 has a large state space, with many possible configurations of tiles on the board. This makes it difficult for the Q-learning algorithm to explore all possible states and learn an accurate Q-function. For example, in the study by Kurek et al. [5], the authors noted that the large state space and sparse rewards in 2048 make it difficult for the Q-learning algorithm to learn an accurate Q-function. They proposed to address this issue by using a hierarchical Q-learning algorithm, which learns a Q-function at different levels of abstraction.

3.1.2. Sparse rewards. The rewards in 2048 are sparse, meaning that the agent only receives a positive reward when it merges tiles. This makes it difficult for the algorithm to learn which actions lead to higher scores, especially in the early stages of the game.

3.1.3. Limited exploration. The Q-learning algorithm may become stuck in local optima, where it chooses suboptimal actions that lead to a dead end. This can be mitigated by using exploration strategies such as epsilon-greedy, but these can also lead to suboptimal performance.

3.1.4. Computational complexity. Q-learning can be computationally expensive, especially when using neural network-based representations of the state. This can limit the scalability of the algorithm to larger and more complex games.

3.1.5. Lack of human-like intuition. Q-learning is a model-free algorithm, which means that it learns solely based on the rewards obtained from the environment. As a result, it may not capture human-like intuition or strategies that are not explicitly represented in the reward function.

To conclude, while Q-learning is a powerful algorithm for solving the game of 2048, it also has some limitations and challenges that need to be addressed. Future research can overcome these challenges and extend the algorithm to more complex tasks, such as playing video games and controlling robots.

3.2. Deep Q-Learning and its limitations

Deep Q-learning is a variant of Q-learning that uses a neural network to represent the Q-function, allowing it to handle high-dimensional and continuous state spaces. Deep Q-learning has been applied to the game of 2048 with promising results, as demonstrated by several studies in the literature.

One example is a study by Zeng et al. [1], in which the authors used a deep Q-network (DQN) to learn a policy for the game of 2048. The DQN took a 4x4x16 matrix as input, where each channel represented the log2 value of the tiles. The output of the DQN was a Q-value for each action, which was used to select the next action to take. The authors used a replay buffer and a target network to improve the stability of the DQN. Similarly, in a study by Matsuzaki et al. [6], the authors used a multi-scale convolutional neural network (MSCNN) to represent the state of the game. The MSCNN took a 5x5x16 matrix as input, where each channel represented the log2 value of the tiles and their neighboring tiles.

The output of the MSCNN was a Q-value for each action, which was used to select the next action to take. The authors used a decaying epsilon-greedy policy and experience replay to improve the stability of the DQN.

The results showed that their DQN agent was able to achieve higher scores in 2048 than the previous state-of-the-art algorithm based on MCTS. The authors also analyzed the behavior of their agent and found that it learned to avoid bad moves and focus on merging large tiles, which is a key strategy in 2048.

In summary, deep Q-learning has been shown to be an even more effective algorithm for solving the game of 2048 compared to Q-learning, using both neural network-based representations of the state. The algorithm has been extended with various techniques, such as replay buffers, target networks, multi-scale CNNs, experience replay, and epsilon-greedy policies, to improve its performance and stability. While DQN solves the inefficiency of Q-Learning when dealing with large state space, it remains volatile to other limitations that apply to Q-Learning in general including sparse rewards, limited exploration, computational complexity, and a lack of human-like intuition.

4. Conclusion

In conclusion, the game of 2048 has become a popular benchmark for testing and evaluating various AI algorithms, including reinforcement learning, neural networks, evolutionary algorithms, and search algorithms. In this paper, the progress made in using reinforcement learning algorithms, particularly Q-learning and deep Q-learning, were reviewed to train agents to play 2048. The studies discussed in this paper have shown that both Q-learning and deep Q-learning can be effective in solving the game, achieving high scores and outperforming other algorithms.

However, both Q-learning and deep Q-learning have limitations and challenges that need to be addressed. These include the large state space, sparse rewards, limited exploration, computational complexity, and a lack of human-like intuition. Researchers have proposed various techniques to address these challenges, including hierarchical Q-learning, double Q-learning, replay buffers, target networks, multi-scale CNNs, experience replay, and epsilon-greedy policies.

The advancements made in using reinforcement learning algorithms to play 2048 have important implications for real-life applications, such as robotics, auto-driving, and financial trading. The methods used to train agents in the game of 2048 can be adopted and extended to solve more complex tasks.

The game of 2048 has proven to be a valuable benchmark for evaluating reinforcement learning algorithms. While Q-learning and deep Q-learning have shown promising results, there is still room for improvements and further research in addressing the limitations and challenges of these algorithms. Future research in this field will benefit from a more integrated approach that combines the strengths of various reinforcement learning algorithms and neural networks, thereby achieving human-like performance in solving complex tasks.

References

[1] Szubert, M. G. and Jaśkowski, W. (2014). Temporal difference learning of N-tuple networks for the game 2048. 2014 IEEE Conference on Computational Intelligence and Games, 1-8. https://doi.org/10.1109/cig.2014.6932907.

[2] Kondo, N. and Matsuzaki, K. (2019). Playing Game 2048 with Deep Convolutional Neural Networks Trained by Supervised Learning. J. Inf. Process., 27, 340-347. https://doi.org/10.2197/ipsjjip.27.340.

[3] Rodgers, P. and Levine, J. (2014). An investigation into 2048 AI strategies. 2014 IEEE Conference on Computational Intelligence and Games, 1-2. https://doi.org/10.1109/cig.2014.6932920.

[4] Kaundinya, V., Jain, S., Saligram, S., Vanamala, C.K. and B, A. (2018). Game Playing Agent for 2048 using Deep Reinforcement Learning. NCICCNDA. https://doi.org/10.21467/proceedings.1.57.

[5] Kurek, M. and Jaśkowski, W. (2016). Heterogeneous team deep q-learning in low-dimensional

multi-agent environments. 2016 IEEE Conference on Computational Intelligence and Games (CIG), 1-8. https://doi.org/10.1109/cig.2016.7860413.

[6] Matsuzaki, K. and Teramura, M. (2018). Interpreting Neural-Network Players for Game 2048. 2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI), 136-141. https://doi.org/10.1109/taai.2018.00038.