Machine Learning Agent Training for Tetris Gameplay: A Comparative Study of Reinforcement Learning Approaches

Jakub Szymczyk
Index: 198134

Krzysztof Taraszkiewicz
Index: [tutaj wpisz numer]

***Abstract*—This paper presents a comprehensive study on training machine learning agents to play Tetris using reinforcement learning techniques. Despite extensive existing research in this domain, our project revealed significant challenges in achieving effective agent performance. Initially, we implemented a classical Deep Q-Network (DQN) agent with actions mirroring keyboard inputs, which proved to be a suboptimal approach that reached performance plateaus rapidly, even with optimizations such as mixed-precision FP16 training. Subsequently, we developed a value network-based agent utilizing grouped actions that lead to specific game states, which are evaluated and ranked by the network to select optimal moves. This approach demonstrated superior performance and faster convergence compared to traditional Q-networks. Additionally, we investigated the impact of different observation methods on agent performance, comparing screen-based input versus direct game state representation. Finally, we benchmarked all developed models against a custom-designed heuristic algorithm to evaluate their relative effectiveness.**

***Index Terms*—reinforcement learning, deep Q-network, value network, Tetris, game AI, machine learning**

I. Introduction

Tetris represents a particularly compelling domain for machine learning analysis due to its unique combination of strategic complexity and temporal decision-making requirements. The game presents an enormous state space with approximately 2^{200} possible board configurations in a standard 10×20 grid, making it computationally intractable to solve through exhaustive search methods.

The fundamental challenge in applying machine learning to Tetris lies in its strategic nature, where optimal play requires long-term planning and the ability to handle delayed rewards. Unlike games with immediate feedback, Tetris rewards are sparse and often significantly delayed, creating a complex temporal credit assignment problem. Players must balance immediate line clearing opportunities against maintaining favorable board configurations for future pieces.

This research addresses the problem of developing effective machine learning approaches for such environments where strategic thinking is paramount and reward signals are temporally displaced. Our primary objective was to collect and compare various reinforcement learning models, evaluating their performance against each other and against traditional heuristic approaches. The scope of this work encompasses the development and analysis of DQN-based agents, value network implementations, and comparative studies of different observation representation methods, as outlined in our methodology section.

II. Background

Reinforcement Learning (RL) provides a mathematical framework for decision-making problems where an agent learns to maximize cumulative rewards through interaction with an environment. In the RL paradigm, an agent observes states $s \in S$, takes actions $a \in A$, and receives rewards r according to the environment dynamics $P(s', r|s, a)$.

The Q-Network approach, fundamental to our implementation, learns an action-value function $Q(s, a)$ that estimates the expected cumulative reward for taking action a in state s and following the optimal policy thereafter. Deep Q-Networks (DQN) extend this concept by using neural networks to approximate the Q-function, enabling application to high-dimensional state spaces.

A DQN agent employs experience replay and target networks to stabilize training. Experience replay stores transitions (s, a, r, s') in a buffer and samples mini-batches for training, breaking correlation between consecutive experiences. The target network, a periodically updated copy of the main network, provides stable target values during training.

Value-based agents, another approach explored in this work, focus on learning state values $V(s)$ rather than action values. In our implementation, the value network evaluates potential future states achievable through grouped actions, selecting the action sequence leading to the highest-valued state.

Our implementation utilizes PyTorch as the deep learning framework and a custom Tetris environment developed in Pygame. This setup provides full control over the game mechanics and enables easy extraction of both visual and structural game state information for agent training.

III. Related Work

Our research encountered challenges similar to those documented in previous studies, particularly the work by Stanford researchers [1]. The Stanford study highlighted the difficulty of training effective Tetris agents using standard DQN approaches, noting issues with sparse rewards and the need for long-term strategic planning.

Like the Stanford researchers, we initially attempted direct keyboard action mapping, which proved problematic due to the

large action space and poor sample efficiency. The Stanford work also explored various reward shaping techniques and state representations, findings that influenced our decision to investigate value-based approaches and alternative observation methods.

The existing literature consistently identifies Tetris as a challenging domain for reinforcement learning due to its delayed reward structure and the requirement for strategic lookahead. Our work builds upon these insights by implementing and comparing multiple approaches within a single experimental framework.

IV. Novelty

This section will present the complete project description including experimental results, performance charts, and detailed analysis of each implemented approach. The novelty of our work lies in the comprehensive comparison of different agent architectures and observation methods within a controlled experimental setting.

V. Methodology

Our methodology encompasses three primary experimental approaches, each addressing different aspects of the Tetris learning problem.

The first approach implemented a classical DQN agent with actions directly corresponding to keyboard inputs (move left, move right, rotate, drop). This agent received game state observations and learned to map states to keyboard actions through experience replay and temporal difference learning. Despite implementing various optimizations including mixed-precision FP16 training for improved computational efficiency, this approach consistently reached performance plateaus early in training.

The second approach utilized a value network architecture where actions represented grouped sequences of moves leading to specific board states. Rather than learning individual key presses, the agent evaluated potential landing positions for the current piece and selected the position with the highest estimated value. This approach significantly reduced the action space complexity and provided more meaningful decision points for the learning algorithm.

The third experimental dimension investigated the impact of observation representation on learning performance. We compared two input modalities: raw screen pixel data versus structured game state information (board configuration, current piece type, next piece preview). This comparison aimed to quantify the effect of input representation on learning efficiency and final performance.

All models were trained using identical hyperparameters where applicable and evaluated against a custom heuristic algorithm that considers factors such as line clearing potential, board height, and hole creation to establish baseline performance benchmarks.

VI. Evaluation

This section will present comprehensive performance comparisons between all developed models and the baseline heuristic algorithm. Evaluation metrics will include average score, lines cleared, game duration, and learning convergence rates.

VII. Conclusion

This research demonstrates the inherent challenges in applying reinforcement learning to strategic games with delayed rewards like Tetris. Our comparative study reveals that value network approaches significantly outperform traditional DQN methods in this domain, likely due to the more appropriate action space representation and reduced temporal credit assignment complexity.

The investigation of observation methods provides insights into the trade-offs between raw sensory input and structured state representations in game learning scenarios. Future work should explore hybrid approaches combining the benefits of both methodologies while addressing the scalability challenges identified in our experiments.

VIII. References

References

[1] CS231n: Convolutional Neural Networks for Visual Recognition, Stanford University, "Playing Tetris with Deep Reinforcement Learning," 2016. [Online]. Available: https://cs231n.stanford.edu/reports/2016/pdfs/121_Report.pdf