Tetris

Xiyu Chen, Zhengrong Gu, Tianxing Jiang

Introduction

The primary motivation for this project was a fun application of neural networks, tetris. Tetris is a well-known game loved by many. In this tile-matching video game, players complete lines by moving differently shaped pieces (tetrominoes), which descend onto the playing field. The completed lines disappear and grant the player points. The game ends when the playing field is filled.

Method

We used deep reinforcement learning to train an Artificial Intelligence to play tetris. We used Deep Q-learning to learn the quality of actions telling an agent what action to take under what circumstances.

Train

Tetris Python:
A great simulator of Tetris is not found in PyGame libraries. Though there's an OpenAI Gym environment for Tetris on the Nintendo Entertainment System (NES) based on the nes-py emulator, it lacks documentation and very hard to use. So, we referenced some other github repos and made our own simulator and rendered with OpenCV.

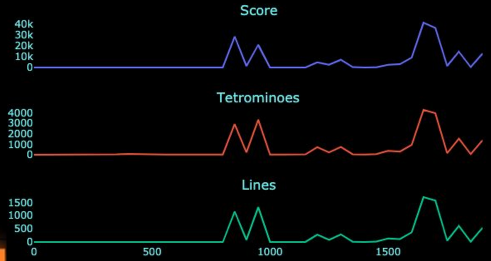
States:
Four states are input in the model.
Lines: The lines will be cleared at the state.
Holes: The holes in the board.
Bumpiness: The gap between columns.
Height: The total heights of all columns.

Reward:
Reward = 1+width*num_cleared_lines^2
Reward = -1 if game over

Agent:
Agent is a simple TensorFlow network. It is trained every time the game ends and has over 1000 states in memory.

Results

We have limited time to try different kinds of rewards and penalties, e.g., penalize the bumpiness, heights or holes of the state. The agent can play the game very well after lots of training. We save the model every 50 trainings and plot the scores, number of pieces and number of cleared lines.



Reference

https://github.com/uvipen/Tetris-deep-Q-learning-pytorch
https://pypi.org/project/gym-tetris/
https://deepai.org/publication/the-game-of-tetris-in-machine-learning