Tetris using Reinforcement Learning

Ravindra Kumar
Ayush Raj
Utsav Anand

Overview

- ❏ Project Description
- ❏ Features used
- ❏ Approaches
- ❏ Analysis
- ❏ Conclusions
- ❏ Challenges
- ❏ Further Aspects

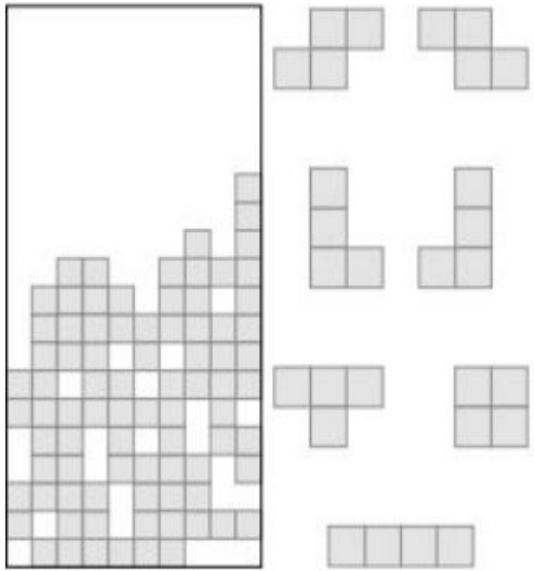
Tetris : A Basic Overview

The objective of the game is to ensure no block touches the upper boundary of the board until a maximum possible score is achieved

The different shapes available are given on the side.

We can rotate and translate them along the x-axis and drop.

AIM : To make as many full rows as possible to lower the height of Tetris



WHAT & WHY

What problem are we solving?

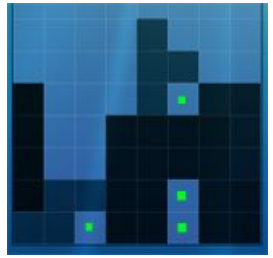
We are making an AI agent which plays the game of Tetris optimally.

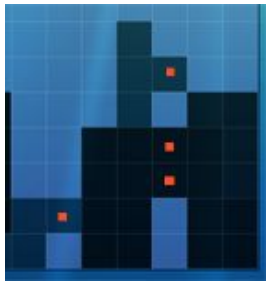
Why solve Tetris?

- AI players : Great challenge for ML community
- Introduction to RL and Genetic Algorithms
- Approximate global optimal solutions needed due to large state space - (2^10-1)^20

Features Used

- ❏ Holes
- ❏ Blockades
- ❏ Maximum Height
- ❏ Average Height
- ❏ Max difference in heights
- ❏ Sum of adjacent differences in heights
- ❏ Row clears





Approaches I:

Genetic Algorithm

- Initialise the weights with random values
- Use 16 different weight vectors and calculate rewards from each one of them.
- Pick up 20% best weight vectors(ones giving better rewards)
- Mate the best 50% weight vectors randomly among themselves and make up remaining more weight vectors.
- With small probability add mutation.

Repeat this till the values get converged.

Approach II

Least Squares Policy Iteration algorithm

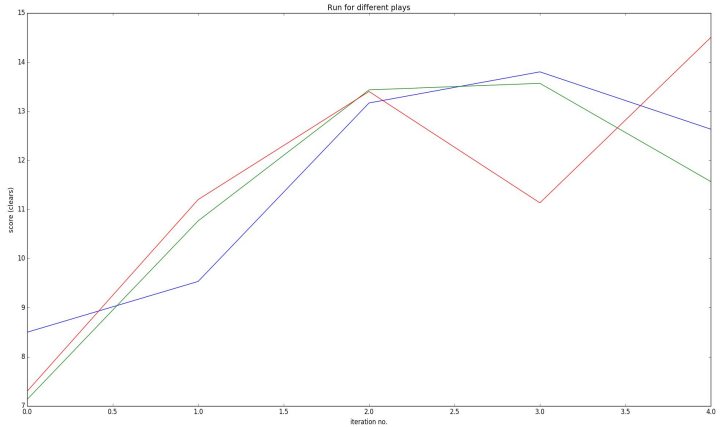
It is a reinforcement algorithm that uses value function approximation to cope with large state spaces

- ➢ Randomly initialise weight vector
- ➢ Generate random board orientations and train the blocks on these boards.
- ➢ Repeat this till convergence or upto a max number of iterations.

This step is time-taking. Although the values do not converge, the results at the end of 200 iterations are quite satisfactory.

Analysis :

- LSPI is generally better than Genetic Algorithm as our learner manages to converge to a relatively good policy using only a few iterations of learning
- Since LSPI does not rely on the mutation chance, we believe that it is more deterministic than GA.
- GA also converges in most of the cases. But due to random mating and mutations, it may not converge
- But we can see that score increases with time



Challenges:

The choice of the features for a particular algorithm was the bottleneck of the project. An optimal subset of features had to be taken in each of the two approaches to get the best solution and a higher probability of convergence.

Coming up with new features was also a difficult part. Needed to look upon some references for new features.

An interesting research question would be to come up with algorithms to learn new features rather than learning just the weights of the features.

References:

- Michail G. Lagoudakis (2003)
  Least-Squares Policy Iteration
  Duke University
- Amine Boumaza (2011)
  How to design good Tetris players
  Universite de Lorraine, LORIA
- Basic Tetris Implementation
- Idea for LSPI

Thank You