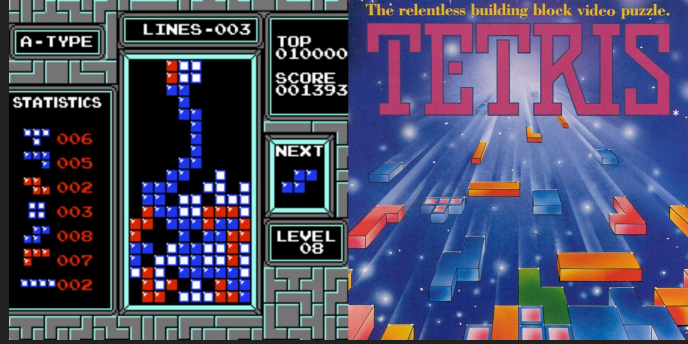
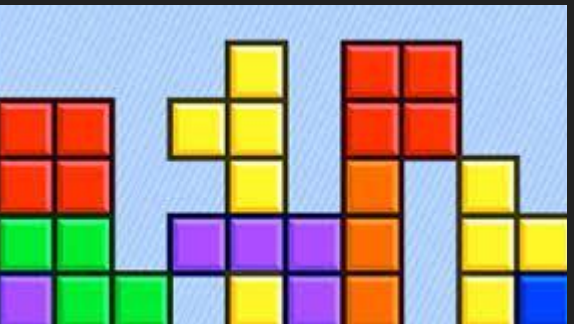


Applying Different Learning Rates to a Self Learning Tetris Agent to Clear 1000 Lines as Fast as Possible

Quinn Ormond



Summary

Project Goal

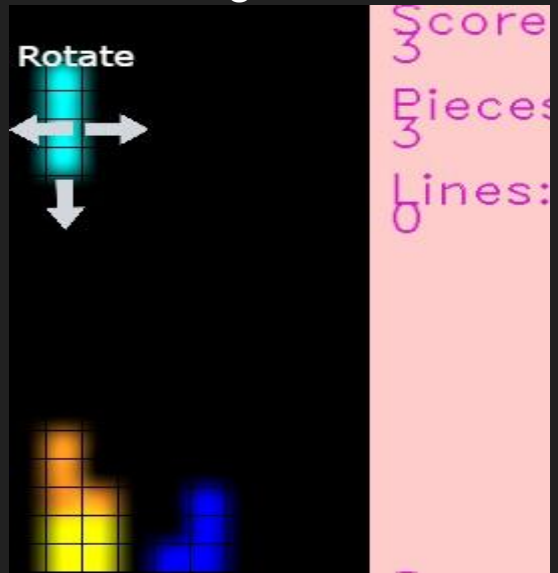
- An agent capable of clearing 1000 lines in Tetris
- Find the learning rate that averagely clears 1000 lines the quickest

The need of the work was to expand my knowledge of reinforcement learning through something I am passionate about.

To the right is an output video from the tetris agent learning

The initial approach taken

- Provide a positive reward to the tetris agent if it cleared a line
- Provide a negative reward for the overall height of the board
- Let the agent figure out where it should move or if it should rotate at any point
- Resulted in extremely slow learning



The final approach taken

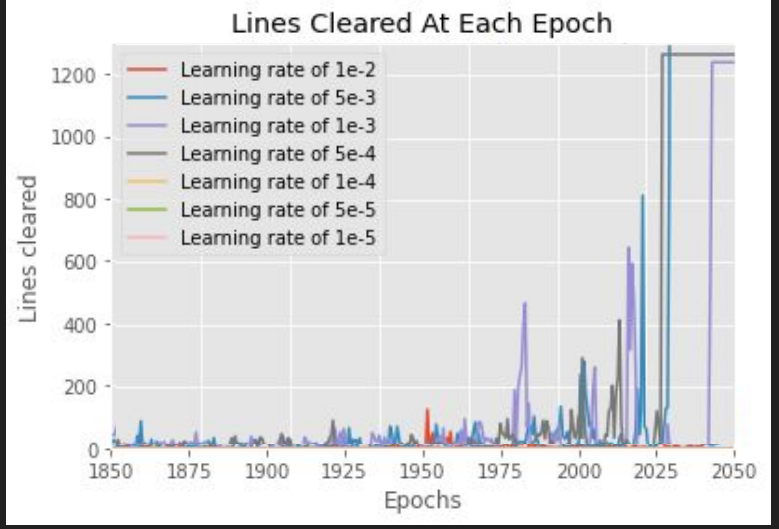
- Implemented Q-learning
- Calculated the efficiency of every piece dropped based on current and future rewards
- Decided where to drop by applying positive or negative weights to those calculations

These images are examples of the different places this I piece can be set. The location is then given a weight based upon current and future rewards of the placement

This sped learning up drastically

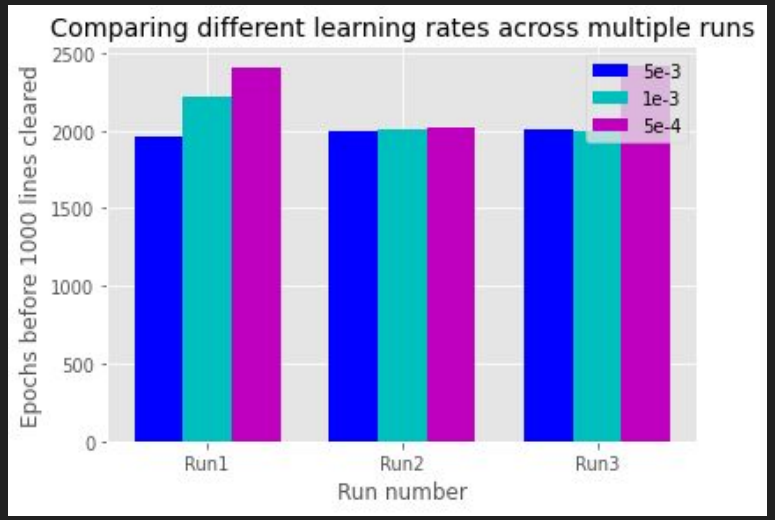
Results

This graph displays the learning rates and how fast they each achieved 1000 lines cleared



From the previous graphs information each one of the fastest three learning rates is run three times to find the average speed they achieve 1000

Average for 5e-3 was: 1990
Average for 1e-3 was: 2077.33
Average for 5e-4 was: 2282.66



Conclusion

- Do your research before just jumping into a big project

- Multiple sources pointed to this means of Q-learning for tracking piece placement and it should’ve been considered before creating an agent

- Given the size of a tetris board you should utilize a learning rate near 5e-3 for the quickest means of achieving 1000 lines cleared