Intelligent 2048 Agent

Kingway Liang Kelly Ndombe Calyx Liu | CS221 Final Project

Project Overview

- Build intelligent agents for the 2048 game
- Aim: Maximize largest tile achieved
- Methods: Expectimax & Deep Reinforcement Learning

Insights

Expectimax:

- When the algorithm is made to look even 3 turns ahead, the runtime is too large to justify looking at every single successor
- With our given heuristics, the agent often chooses actions differently from our human best estimates, that benefit in the longer run.

Deep Reinforcement Learning:

- Reward needs to encode the notion of desirability of the states resulting from an action
- When training set size is large, number of epoch should be small
- Having correct label of Q values to train on is difficult

Future Improvements

Expectimax:

- Continue to investigate ways to cut down on runtime without losing efficacy
- Identify optimal number of turns to look ahead
- Tune the weight each score factor is given

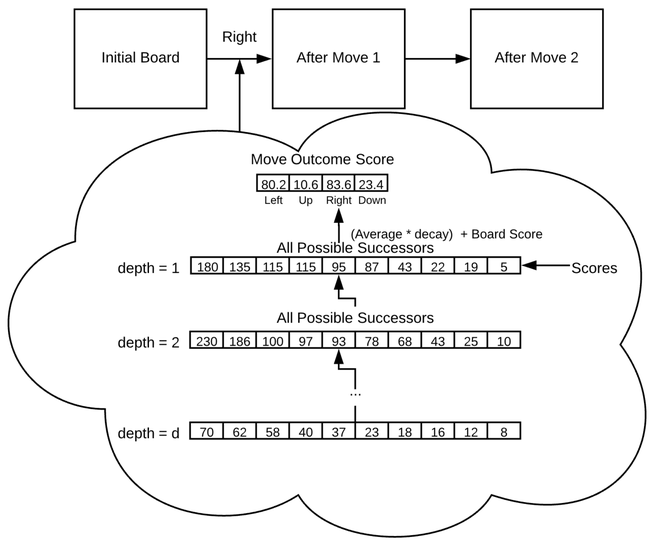
Deep Reinforcement Learning:

- Investigate other reward formulations, potentially use advanced evaluation function to label reward
- Tune neural network architectures
- Rigorously label Q values according to the known transition distribution of the states

Expectimax

- MDP based approach with states, actions, successors and an eval function
- Looks at the outcome of the next *d* turns and calculates path with largest potential
- Features that are factored into score
  - The number of possible merges
  - The number of empty tiles
  - The value of the highest valued tile
  - The sum of all the squared values of the tiles
  - Whether the largest tiles are on the bottom

Algorithm Visualization



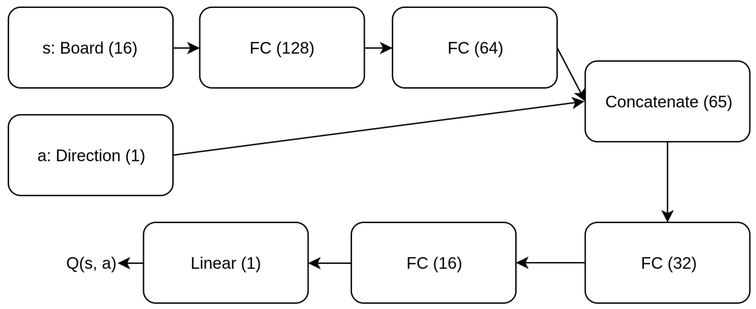
Results

- Decay >= 0.8: Max Tile 512
- Decay <= 0.7: Max Tile 512 and 1024
- Decay ~~ 0.7 MaxTile 1024 and 2048

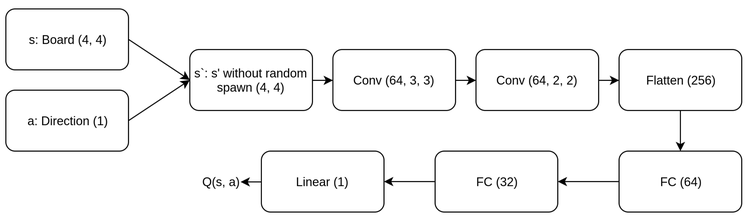
Deep Reinforcement Learning

- DQN: Use neural network for global approximation in Q-learning
- Bellman Equation: $Q^*(s,a) = \mathop{\mathrm{E}}_{s' \sim P}\left[r(s,a) + \gamma \max_{a'} Q^*(s',a')\right]$
- Reward: Number of tiles merged per move
- Training: collect and log episodes, then train offline
- Architectures: FC only with (s, a) as input & CNN with s` as input

Architectures



Fully Connected Architecture



Convolutional Architecture

Best Results

Max Tile	1024	2048
Occurrence	40%	40%