AI Plays 2048, Fall 2016

Introduction,goal,approaches

- With the 4x4 grid,we can swipe it down, up, left of right and adjacent numbers which are equal and aligning in the swiping direction will merge to form new numbers.
- In the project, our goal is to use machine learning algorithm to maximize the total score we get and try to get 2048 or even higher number tiles as ofter as possible.

- We used 3 methods to do the job:
    - MiniMax (pruned)
    - ExpectiMax (pruned)
    - Reinforcement learning

MiniMax

- The math equation of our algorithm:

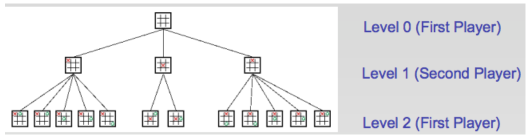
$$V_{\max,\min}(s,d)=\begin{cases} totalscore(s), IsEnd(s) \\ Eval(s)=currentScore, d=0 \\ \min_{a\in Actions(s_1)} V_{\max,\min}(Succ(s_1,a),d), Players(s)=computer \\ \max_{a\in Actions(s)} V_{\max,\min}(Succ(s,a),d-1), Players(s)=human \end{cases}$$

in which

$$s \in \{up, down, left, right\}$$

$$s_1 \in \{2,4 \xrightarrow{fill} empty\_positions\}$$

The search tree is shown as below.



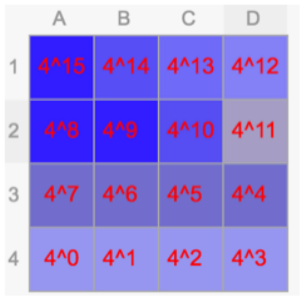
ExpectiMax

- We used 2 types of expects-max algorithms:

1. When depth = 0, we used a weight matrix to multiply the current board and do the summation as following.

$$score=\sum_{i=0}^{4}\sum_{j=0}^{4} weight[i][j]*board[i][j]$$

Here is the weight matrix we use for this step.



	A	B	C	D
1	4^15	4^14	4^13	4^12
2	4^8	4^9	4^10	4^11
3	4^7	4^6	4^5	4^4
4	4^0	4^1	4^2	4^3

2. When depth = 0, we used the score got above to multiply the actual score when at depth = 0.

Score = score * currentScore

Pruning

- To improve running time, we pruned search trees so that AI does not consider all empty tiles. In particular, at depth higher than 3, the number of empty tiles we examine is:

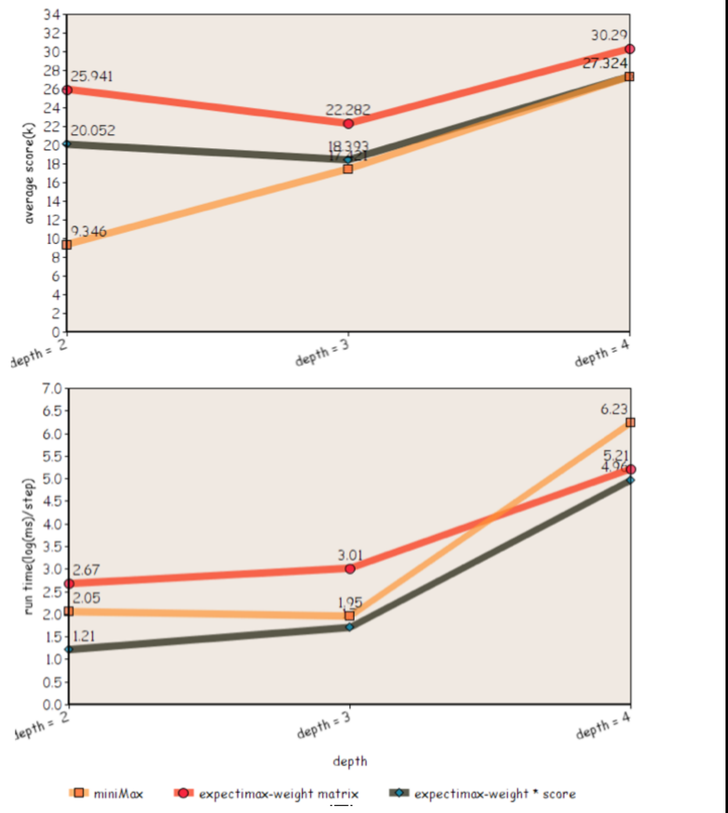
min(min(depth, 4), num of empty tiles) or
min(min(depth, 6), num of empty tiles)

- We consider only the "most important" tiles according to weight matrix.

Result Analysis

average score & run time v.s. depth

(pruning level 4 when depth >=3)



Reinforcement Learning

We generated MDP and discretized board states by:

1) tiles monotonicity
2) number of empty tiles

and implemented value iteration.

Current training has not achieved result as good as Minimax, and we are working to improve it by leveraging Q-learning.