Theory of Computer Games 2022 – Project 2 plus

311551069 余忠旻

Network Design

我的 N-tuple network 和 project 2 一樣

可以看到下圖是依照 4 x 6-tuple network 並且有 8 種 isomorphic patterns 需要去儲存(也就是旋轉 0°, 90°, 180°, 270° 以及水平鏡射後的旋轉 0°, 90°, 180°, 270°)

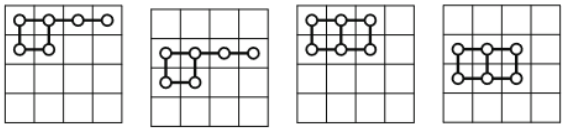

Fig. 2. The four 6-tuples by Wu et al. [11]

Method Used

Take action

跟上次不同的地方在於

Project 2: The player takes actions based on the rewards and the afterstate values, $r_t+V(s_t')$

而我這次 improved methodology 是 The player takes actions based on the rewards and the expectation values, $r_t+\mathrm{E}[r_t'+V(s_t'')]$

也就是參考 reference 2 把 expectimax 融入 TD-learning 中:

K.-H. Yeh, I-C. Wu, C.-H. Hsueh, C.-C. Chang, C.-C. Liang, and H. Chiang, “Multistage temporal difference learning for 2048-like games,”

A non-initial new tile is randomly placed at an empty cell on the puzzle border of the opposite side of the last sliding direction.

而 Board 的 hint 會顯示接下來會放的 new tile 是 1-tile, 2-tile or 3-tile

可以看到下圖我會根據 board 的 hint 去計算上述所說 non-initial new tile 可以放的位置的期望值 $\mathrm{E}[r_t'+V(s_t'')]$

```
242     float expectationEstimate(const board& after) const {
243         float expectation = 0.0;
244         int emptySpace = 0;
245 
246         std::vector<int> space = spaces[after.last()];
247         //std::shuffle(space.begin(), space.end(), engine);
248         int bag[3], num = 0;
249         for (board::cell t = 1; t <= 3; t++)
250             for (size_t i = 0; i < after.bag(t); i++)
251                 bag[num++] = t;
252         std::default_random_engine engine;
253         std::shuffle(bag, bag + num, engine);
254         board::cell tile = after.hint() ?: bag[--num];
255         board::cell hint = bag[--num];
256 
257         for (int pos : space) {
258             if (after(pos) != 0) continue;
259 
260             board b = board(after);
261             b.place(pos, tile, hint); // place 1, 2, 3
262             int bestReward = -1;
263             float bestValue = -100000;
264             for(int op : opcode){
265                 board afterstate = b;
266                 int reward = afterstate.slide(op);
267                 if(reward == -1) continue;
268                 float value = valueEstimate(afterstate);
269                 if(reward + value > bestReward + bestValue){
270                     bestReward = reward;
271                     bestValue = value;
272                     //bestOP = op;
273                     //bestAfterstate = afterstate;
274                 }
275             }
276 
277             expectation += (bestReward + bestValue);
278             emptySpace += 1;
279         }
280 
281         expectation = expectation / emptySpace;
282         return expectation;
283 
284     }
```

TD-learning

ValueExtract, ValueEstimate, ValueAdjust, training process 都與 project 2 相同

n-tuple network weights 與 project 2 也相同

Result

```
[chungminyu@tcglinux7 pj2+]$ ./threes --total=1000 --block=1000 --limit=1000 --play="alpha=0 load=weights.bin"
--save stats.txt
Threes! Demo: ./threes --total=1000 --block=1000 --limit=1000 --play=alpha=0 load=weights.bin --save stats.txt

1000    avg = 277576, max = 711885, ops = 88307 (44869|4045303)
        1536    100%    (0.1%)
        3072    99.9%   (93.3%)
        6144    6.6%    (6.6%)
```