

Homework1 report

刘喆骐 2020013163 探微化01

1. AI 表现

1. Minmax search

在3*3的井字棋中ai能够寻找到不输策略。如下图所示。

	0	1	2
2	O	X	X
1	X	X	O
0	O	O	X

Game end. Tie

图1 人-机器

	0	1	2
2	X	O	X
1	O	O	X
0	X	X	O

Game end. Tie

图2 机器-人

2. Alpha-Beta search

在4*3的棋盘上，alpha-beta search能够较快的行棋并且找到先手必胜的策略。

	0	1	2	3
2	X	X	0	0
1	0	0	X	-
0	X	X	0	X

Game end. Winner is AlphaBetaSearchPlayer 1

图3 alpha-beta-人

3. Alpha-Beta Cutting off search

前三种搜索在大棋盘会表现处糟糕的结果，而剪枝算法能够得到较好的结果，能够较快的做出决策并且体现出智能，例如会进攻和防御。如下图所示，它能够主动进攻，防御活三、冲四。

	0	1	2	3	4	5	6	7	8
8	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-
5	-	-	-	X	0	0	0	X	-
4	-	-	-	-	X	X	-	-	-
3	-	-	-	-	X	0	0	-	-
2	-	-	-	-	-	-	X	-	-
1	-	-	-	-	-	-	-	-	-
0	-	-	-	-	-	-	-	-	-


Your move: 

图4 防御活三

	0	1	2	3	4	5	6	7	8
8	-	-	-	-	-	-	-	X	-
7	-	-	-	-	-	-	0	-	-
6	-	-	-	-	-	0	-	-	-
5	-	-	-	X	0	0	0	X	-
4	-	-	-	0	X	X	-	X	-
3	-	-	X	-	X	0	0	-	-
2	-	-	-	-	-	-	X	-	-
1	-	-	-	-	-	-	-	-	-
0	-	-	-	-	-	-	-	-	-


Your move: 

图5 防御冲四

让两个剪枝算法对抗，能够得到较好的结果，将棋盘填满，和棋。

	0	1	2	3	4	5	6	7	8
8	X	X	0	0	0	X	X	X	0
7	X	0	X	X	X	0	0	0	X
6	0	X	0	0	0	0	X	X	0
5	X	0	X	0	X	0	X	X	X
4	0	X	X	0	X	X	X	0	0
3	0	X	X	X	0	X	0	X	0
2	X	0	X	0	0	X	X	X	0
1	0	0	0	X	0	0	X	0	X
0	0	X	0	0	0	X	X	0	X

Game end. Tie

图6 剪枝对抗

4. MCTS search

MCTS搜索产生的结果主要是非常随机，并且行棋速度较慢，这是由于其随机选择节点所致。但是相较于Minxmax而言，其速度依旧较快。

MCTS和minmax对战结果如下：

	0	1	2
2	X	0	X
1	X	0	X
0	0	X	0

Game end. Tie

图7 MCTS-minmax

5. Alphazero search

结果较好，其行棋具有一定的智能，和alpha beta剪枝类似。和人类行棋有较好的结果。但是其对于斜线连珠的防御不足，此局我获胜就是使用斜线，制造了两个活三，进而获胜。而下面一局alphazero构造了活三和冲四，进而获胜。

	0	1	2	3	4	5	6	7	8
8	-	-	-	-	-	-	-	-	-
7	-	0	-	0	-	0	-	-	-
6	X	0	X	X	X	-	0	-	-
5	-	X	-	X	-	X	X	0	-
4	-	-	X	X	X	0	X	-	-
3	-	-	-	X	0	X	0	-	-
2	-	-	X	0	X	0	0	0	-
1	-	0	-	-	-	-	0	-	-
0	-	-	-	-	-	-	-	-	-

Game end. Winner is Human 1

图8 人-alphazero

	0	1	2	3	4	5	6	7	8
8	-	-	-	-	-	X	-	-	-
7	-	-	0	-	X	-	-	-	-
6	-	-	X	X	0	-	-	-	-
5	-	-	X	0	0	0	-	-	-
4	0	X	X	X	X	0	-	-	-
3	-	-	-	0	X	0	-	-	-
2	-	-	-	-	X	0	-	-	-
1	-	-	-	-	-	X	-	-	-
0	-	-	-	-	0	-	-	-	-

Game end. Winner is AlphaZeroPlayer 1

图8.1 alphazero-human

6. AI 比较

行棋较好的两个是Alphazero和Alpha_Beta_cuttingoff，将这两者进行对战，得到如下结果：

	0	1	2	3	4	5	6	7	8
8	0	X	X	0	X	0	0	X	X
7	0	X	X	0	X	X	X	0	X
6	X	X	0	X	0	X	X	X	0
5	X	0	X	X	0	0	0	X	X
4	X	0	0	0	X	X	X	0	0
3	0	X	0	X	0	0	0	X	X
2	0	X	0	0	X	0	X	0	0
1	0	0	X	0	X	X	X	X	0
0	X	0	0	X	0	0	0	0	X

Game end. Tie

图9 alphazero-剪枝

棋盘填满，最终和棋，活三等得到防御，两者都会主动进攻。程序运行时间较长，两者水平接近，但是alphazero耗时略长。

2. alpha-beta搜索和朴素minimax搜索

在4*3的棋盘中进行比较。

alpha-beta搜索耗时：4s。

朴素minimax搜索耗时：运行超过13min没有结果。

说明alpha-beta搜索更加高效。

3. 评估函数设计方案

针对不同的行棋方设定不同的评估函数。当 $p=player$ 时，分数1=活四 $\times 10000$ + 冲四 $\times 10000$ + 活三 $\times 10000$ + 冲三 $\times 50$ + 活二 $\times 10$ - 最大距离，如有两个活三，加10000分。 $p!=player$ 时，分数2=活四 $\times 10000$ + 冲四 $\times 700$ + 活三 $\times 700$ + 冲三 $\times 40$ + 活二 $\times 10$ - 最大距离，如有两个活三，加10000分。返回时进行归一化，使得返回值在 $[-1,1]$ 之间。

4. MCTS与alpha-beta搜索的对战结果与分析

MCTS的效果很差，接近于随机，不会对活三、冲四等进行防御。而alpha-beta剪枝行棋不随机，会使用活三等进攻，符合五子棋行棋规律，较为合理。

	0	1	2	3	4	5	6	7	8
8	-	-	-	-	-	-	-	-	-
7	-	-	-	-	0	-	-	-	-
6	-	X	-	-	0	-	-	-	-
5	-	-	-	-	0	-	-	-	-
4	-	-	-	X	0	-	-	-	-
3	-	-	X	-	0	-	-	-	-
2	-	-	-	-	X	-	-	-	-
1	-	-	-	X	-	-	-	-	-
0	-	-	-	-	-	-	-	-	-

Game end. Winner is CuttingOffAlphaBetaSearchPlayer 2

图10 MCTS-剪枝

5. MCTS和AlphaZero的对比

Alphazero行棋更加合理，不像MCTS那样随机。在c值较小(0.01)时，MCTS依旧较为随机，不会进攻和防御，而Alphazero行棋合理，会主动进攻。如下图，Alphazero直接连成5子获胜，而MCTS接近随机行棋。

	0	1	2	3	4	5	6	7	8
8	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-
6	-	-	-	X	-	-	-	X	-
5	-	-	-	-	-	X	-	-	-
4	-	0	0	0	0	0	X	-	-
3	-	-	-	-	-	-	-	-	-
2	-	-	-	-	X	-	-	-	-
1	-	-	-	-	-	-	-	-	-
0	-	-	-	-	-	-	-	-	-

Game end. Winner is AlphaZeroPlayer 2

图11 Alphazero-MCTS