烟台大学
YANTAI UNIVERSITY

基于强化学习的黑白棋的设计与实现

Design and Implementation of Othello Based on Reinforcement Learning

Zhao Qian

Adviser: Jerry Wang

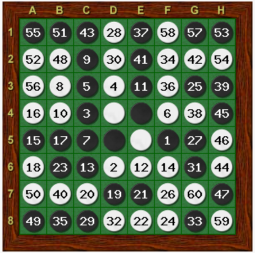
Yantai University
School of Computer and Control Engineering

2019.05.30

Outline

Why Reversi and Prior Work

Why Reversi?



- 以往算法的设计中大都使用博弈树搜索的方法
- 2017 年 AlphaGo Zero 出现了
- 打算将这种思路扩展到黑白棋中

Prior Work

Minimax Tree Search + Alpha-Beta pruning

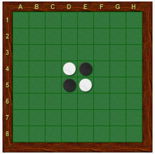
- 搜索空间巨大（随搜索层数指数级增加）
- 棋力受限于搜索的层数（理想时间内很难提高）
- 棋力受限于设计者的能力（如估值函数的设计）

Monte Carlo Tree Search

- 无需任何领域知识便可工作
- 非对称式增长，算法会频繁地访问“更感兴趣”的节点，并聚焦搜索空间于更加相关树的部分
- 算法可在任何时间终止，并返回当前最优的估计

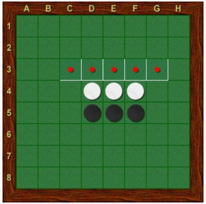
Reversi: Basic Rules

How to play

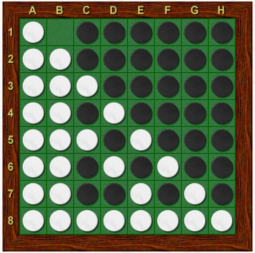


- 棋盘 8×8 大小
- E4、D5 为黑棋
- D4、E5 为白棋
- 执黑先手

- 落子必须在一空位
- 落子必须造成翻转
- 无子可走则本轮 pass
- 有子可走则必须走棋



How to win



获胜条件：

1. 双方都无子可下
2. 棋子数目多者获胜，若相等为平局

Reinforcement Learning

Sample Text......
Sample Text

Structure Breakdown

Sample Text
Sample Text

Neural Policy and Value Network

Sample Text
Sample Text

Monte Carlo Tree Search for Policy Improvement

Sample Text
Sample Text

Experiments and Analysis

Sample Text
Sample Text

Conclusions and Prospect

Sequence Tagging Loss

$$\mathcal{L}_p = -\sum_{i=1}^{S}\sum_{j=1}^{N} p_{i,j}\log(\hat{p}_{i,j})$$

Language Classifier Loss

$$\mathcal{L}_a = -\sum_{i=1}^{S} l_i \log(\hat{l}_i)$$

Bidirectional Language Model Loss

$$\mathcal{L}_l = -\sum_{i=1}^{S}\sum_{j=1}^{N} \log(P(w_{j+1}|f_j)) + \log(P(w_{j-1}|b_j))$$

Acknowledgement

- 感谢王建华老师在毕设期间的指导
- 感谢 ACM 实验室卢云宏老师、周世平老师、封玮老师
- 感谢班主任毕远伟老师
- 感谢四年里遇到的各位老师和同学

请多提宝贵意见