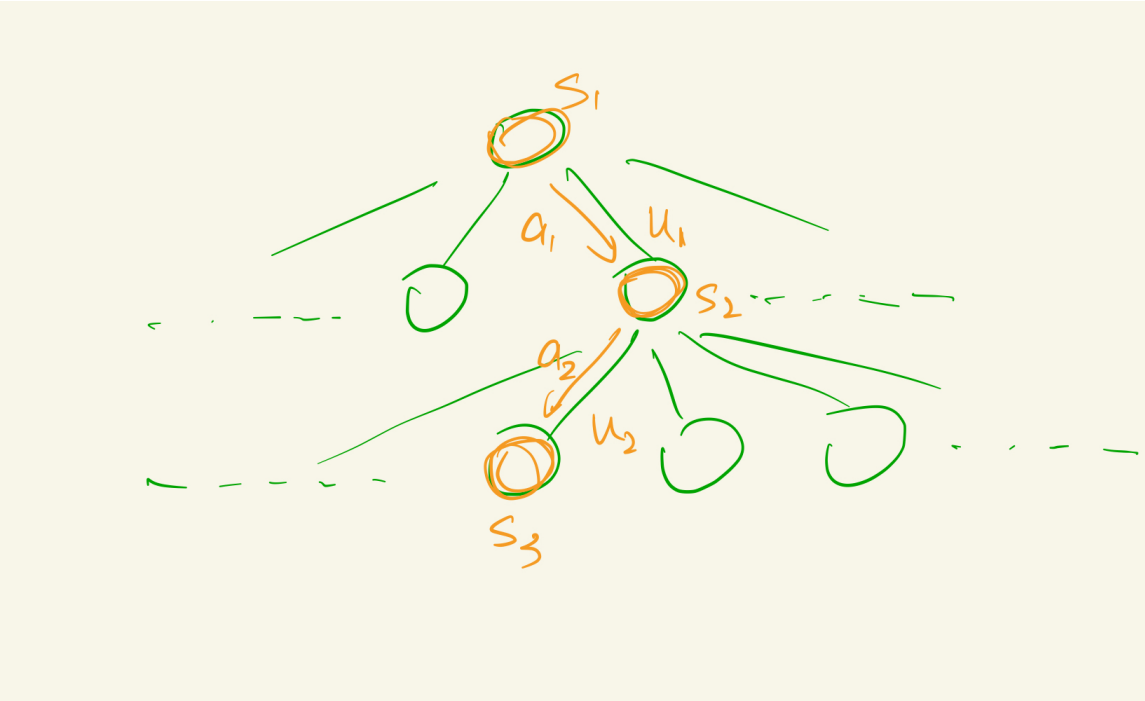
hoho_chess

credit assignment

- 方案一：直接利用MCTS进行节点select时计算的u值作为当前状态的即时奖励：



由于最后一步的奖励根据胜负平分别为1, -1，0，需要对u值进行归一化到区间[0, 1]，做法是进行模拟走子时不断更新树的最大u值与最小u值，当真实走子时通过下式进行归一化：

$$u_{normalize} = \frac{u_{select} - u_{min}}{u_{max} - u_{min}}$$

由此，MDP过程就是$\{s_1, a_1, u_1, s_2, a_2, u_2, ...s_T, a_T, r_T\}$

- 方案二

参考论文”**Hindsight Experience Replay**”（https://arxiv.org/abs/1707.01495）的实现，

通过多目标的强化学习方式，对收集的经验数据进行改造：

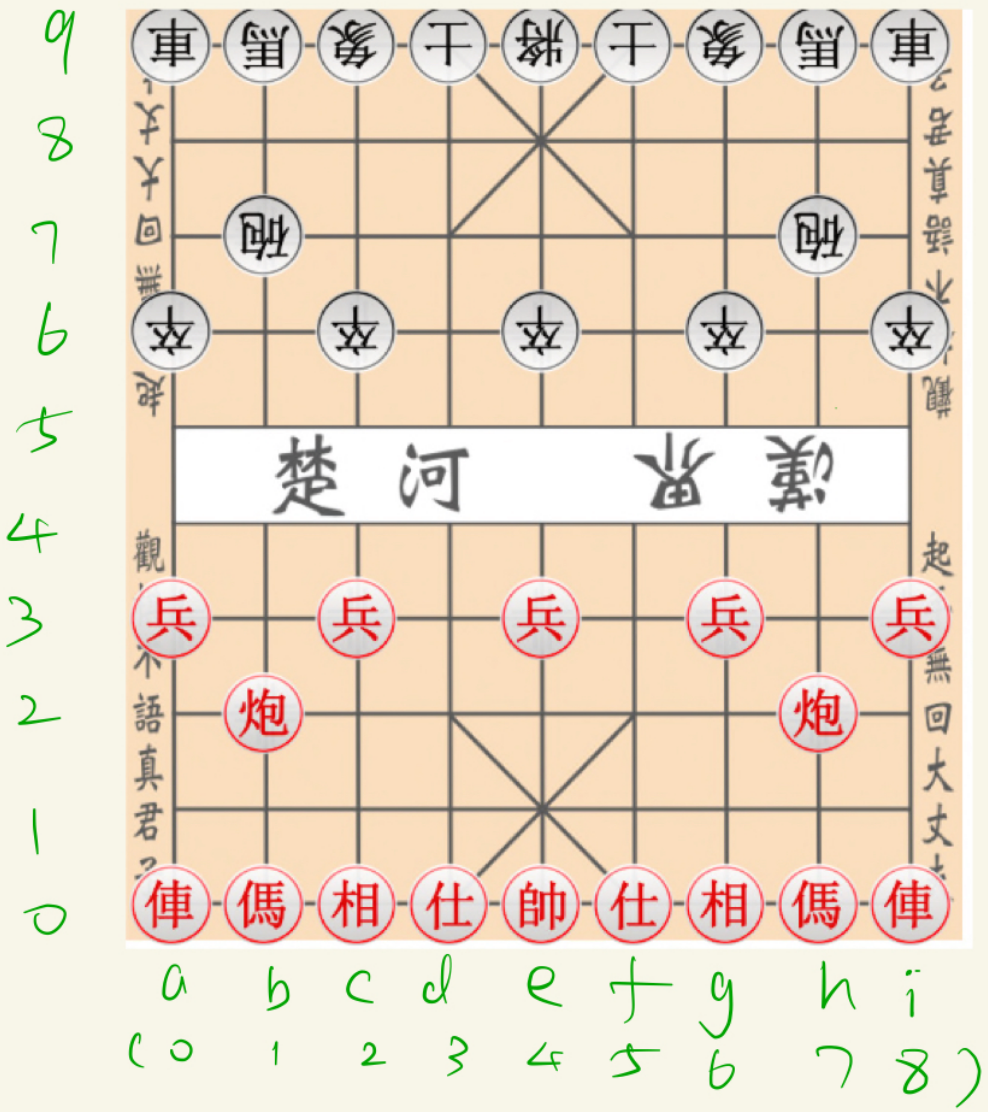
1. 对当前状态s_t，随机采样它的后面第k个状态s_{t+k}作为目标状态
2. 判断s_t的下一个状态s_{t+1}是否为s_{t+k}，是则将即时奖励改为1，否则为0

```
step_state = np.random.randint(traj.length)
state = traj.states[step_state]
step_goal = np.random.randint(step_state + 1, traj.length + 1)
goal = traj.states[step_goal]
next_state = traj.states[step_state + 1]
is_achieved = (next_state is goal)
done = True if is_achieved else False
reward = 1 if is_achieved else 0
pi = traj.policies[step_goal]
self.add(state, pi, reward)
```

关于状态表示

状态即当前棋盘的棋子分布情况。

- 棋盘现实表示如下：



- 将以上现实表示映射为10x9的字符二维数组：

```
"RNBAKABNR",
"         ",
" C     C ",
"P P P P P",
"         ",
"         ",
"p p p p p",
" c     c ",
"         ",
"rnbakabnr"
```

为了对应现实棋盘行列序号（行号从下往上递增，列号从左往右递增），将红方棋子（开局）放在二维数组上半部，黑方则在下半部。

其中字符表示棋子种类：

```
K：帅，A：仕，R：车，B：相，N：马，P：兵，C：炮/
大写红方，小写黑方
```

- 进一步，可将以上二维数组压缩为字符串表示（方便打日志和后期绘制状态图），如：

```
'RNBAKABNR/9/1C5C1/P1P1P1P1P/9/9/p1p1p1p1p/1c5c1/9/rnbakabnr'
```

其中数字表示棋子之间（或棋子与左右边界）空格的数量，斜杠“/”表示换行。

- 模型的状态输入表示

统计下来红黑双方一种14种棋子，每种棋子一个plane，所以模型输入向量$S \in \mathbb{R}^{14\times10\times9}$，棋子所在位置置为1，其余置为0。

（如果像Alpha Zero考虑前7个和后7个上下文棋局，输入向量则为$S \in \mathbb{R}^{15\times14\times10\times9}$，但为了计算简单，这里暂没实现，只考虑了当前棋局状态）。

关于动作表示

据统计，象棋棋盘一共2086种走法，所以动作向量$V \in \mathbb{R}^{2086}$

备注：每个走法为形如”a2c4”这样的字符串形式，意义为：在a列2行的棋子走到c列4行。