讲给朋友们的AlphaGo

王志鹏 博士

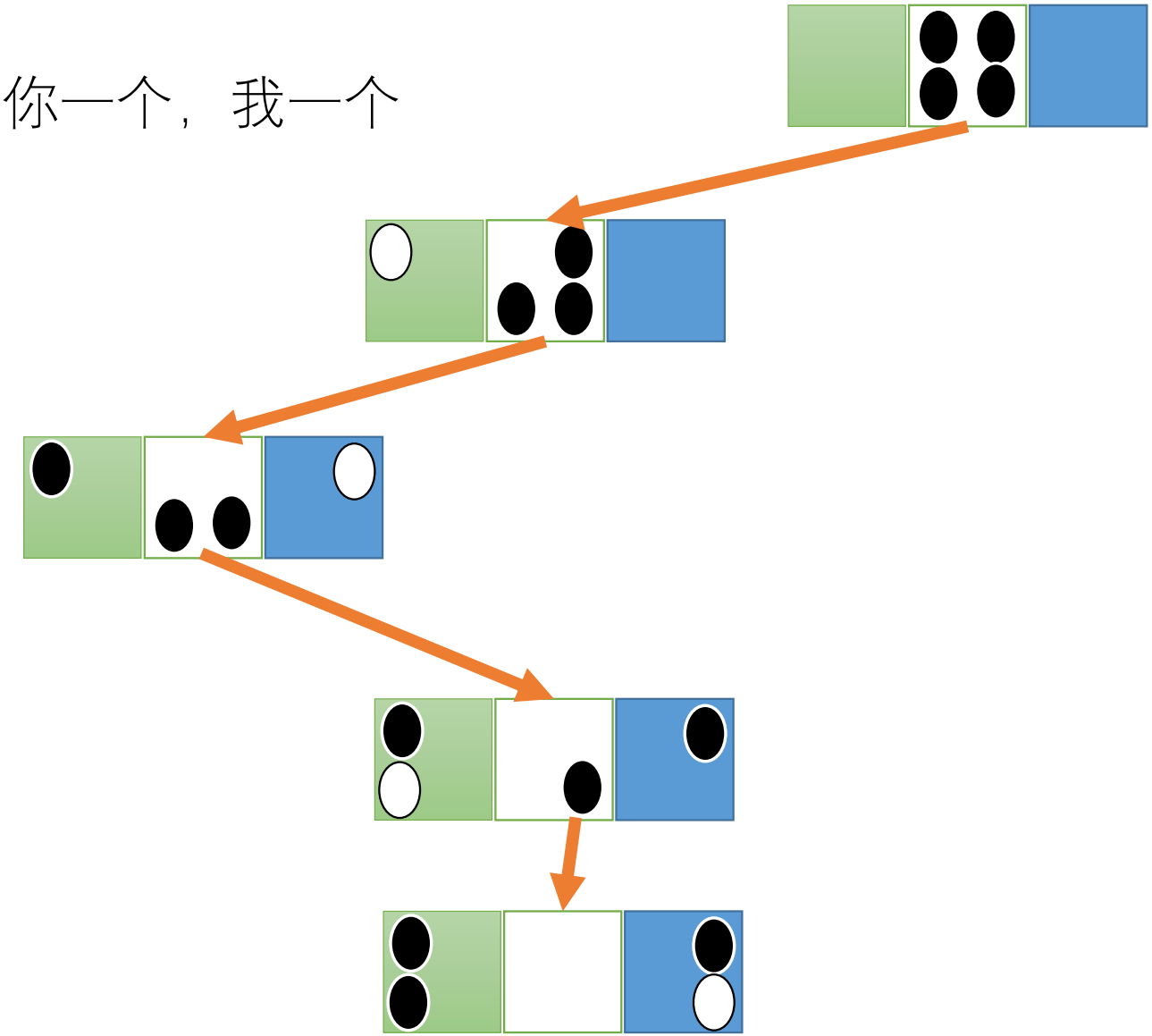
2018.8.4

四蛋博弈（Game）

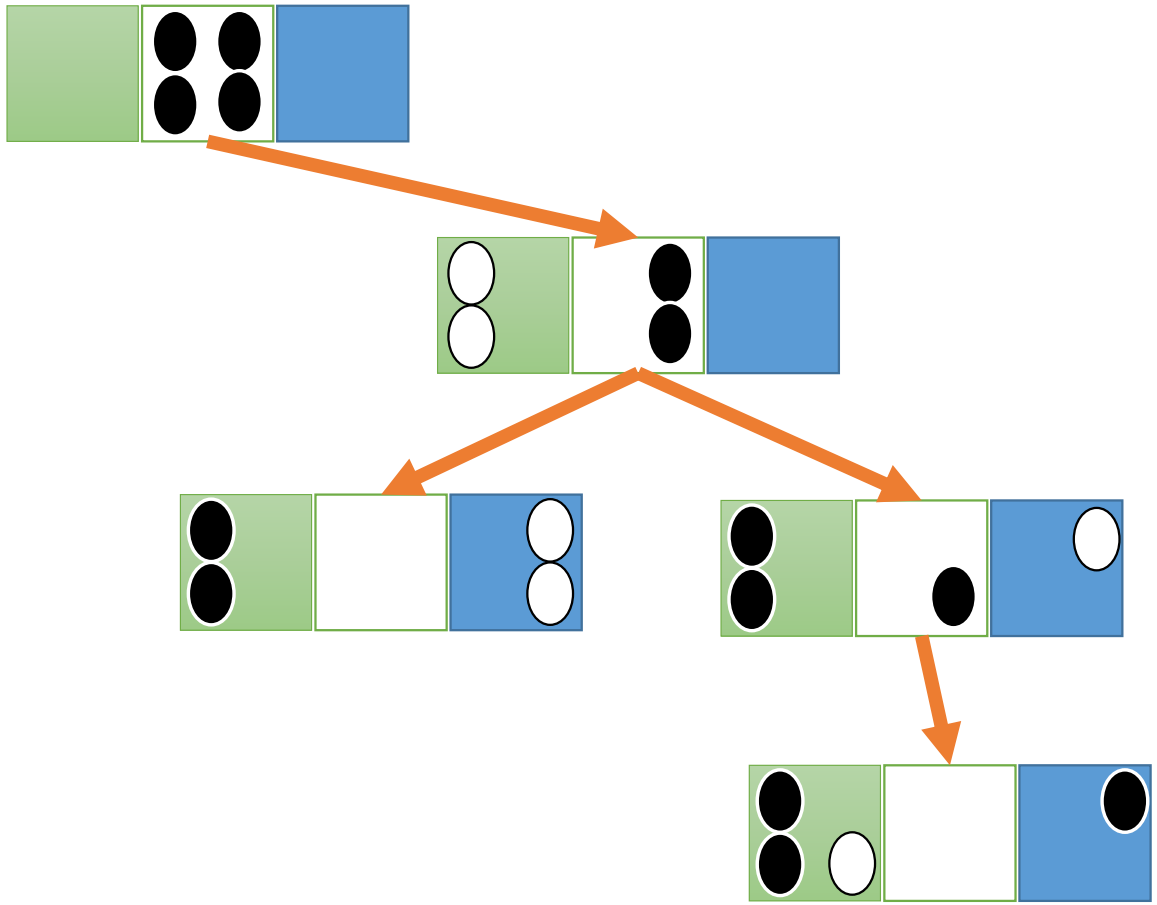
请忘记“凑三”的解析解！

- 桌子上有四颗蛋
- 甲（绿色）和乙（蓝色）每轮须从桌子上拿一颗或两颗蛋
- 甲先拿
- 拿到最后一颗蛋为胜利

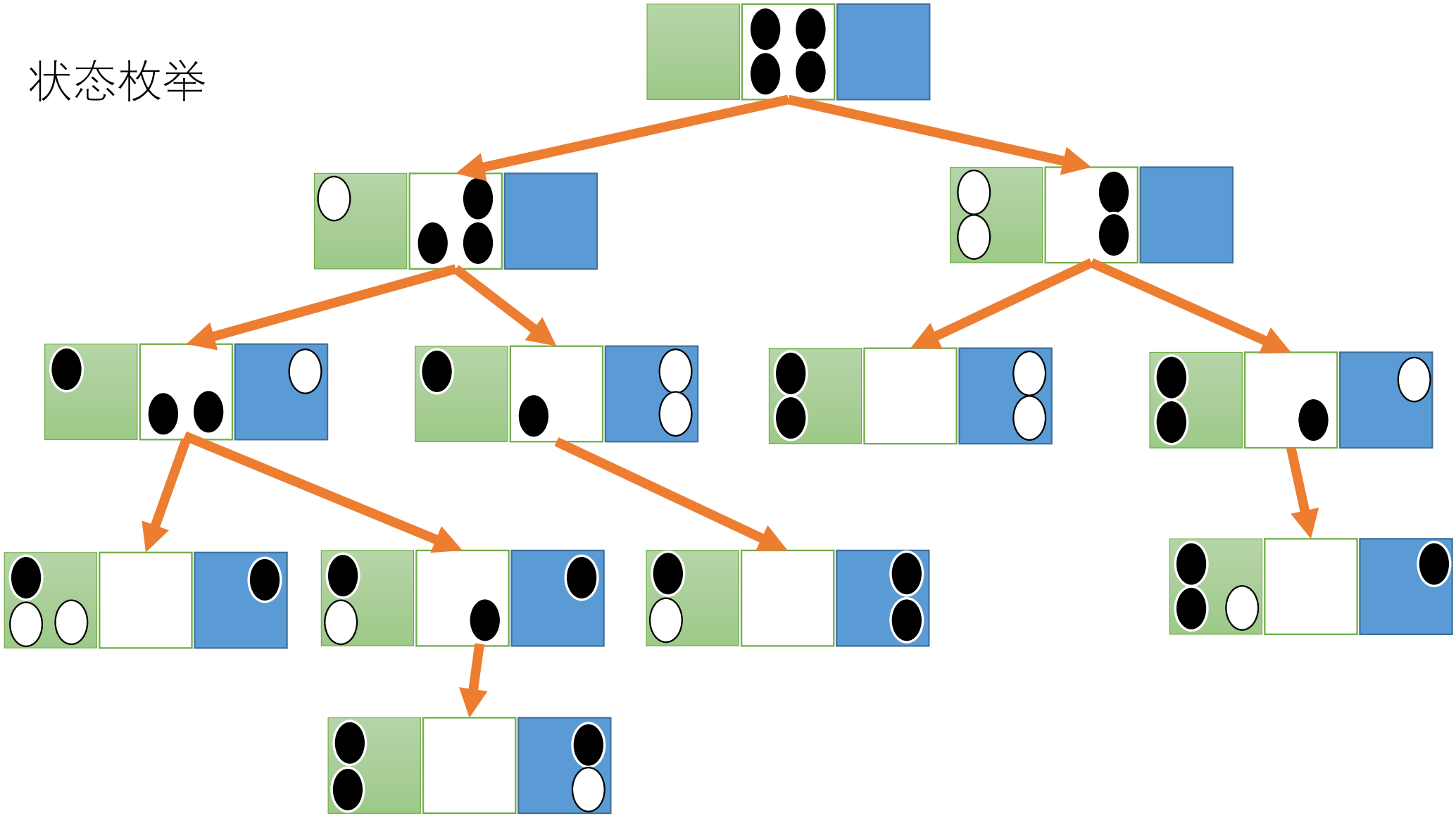
你一个，我一个



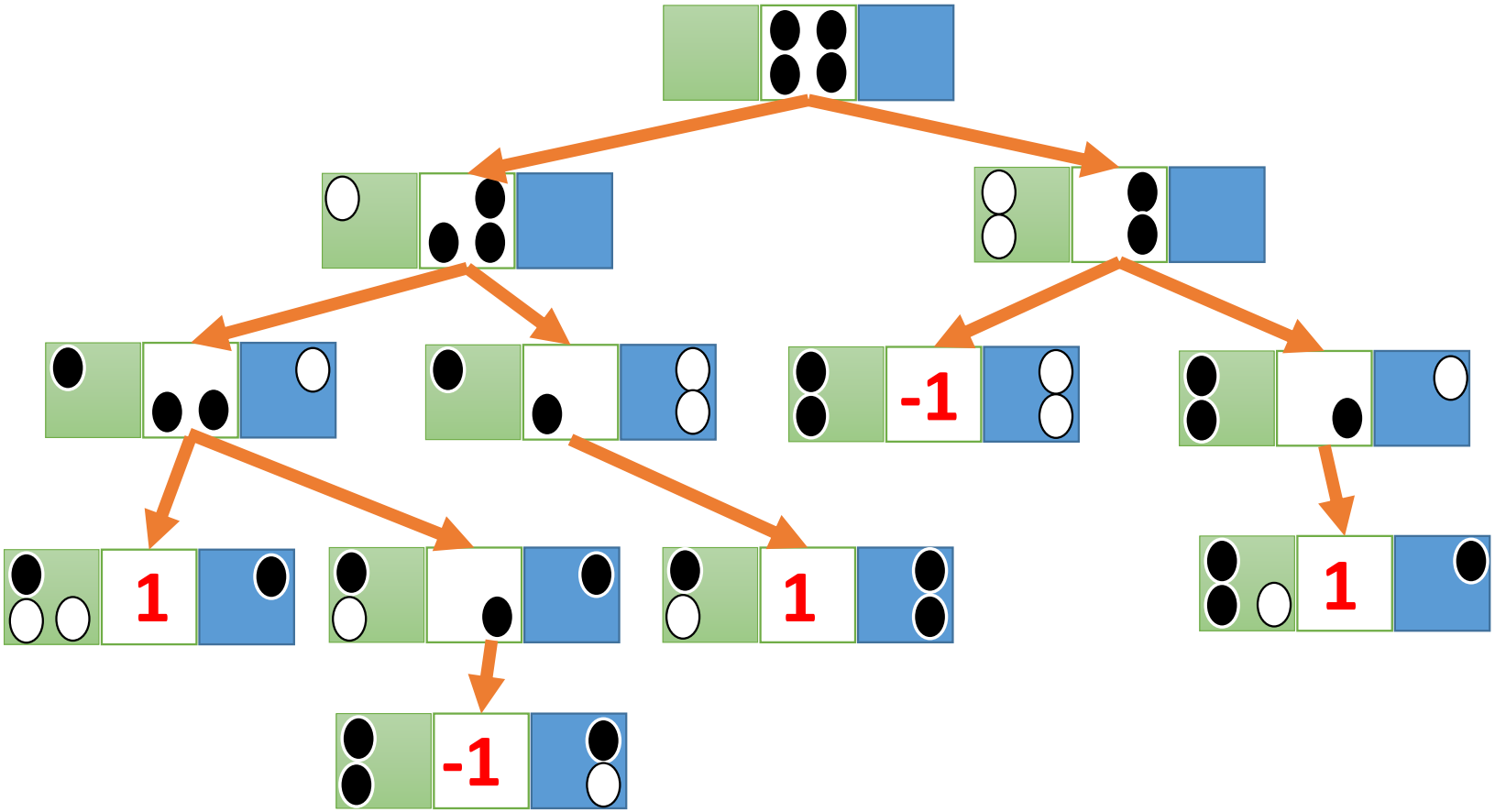
我两个，你随意



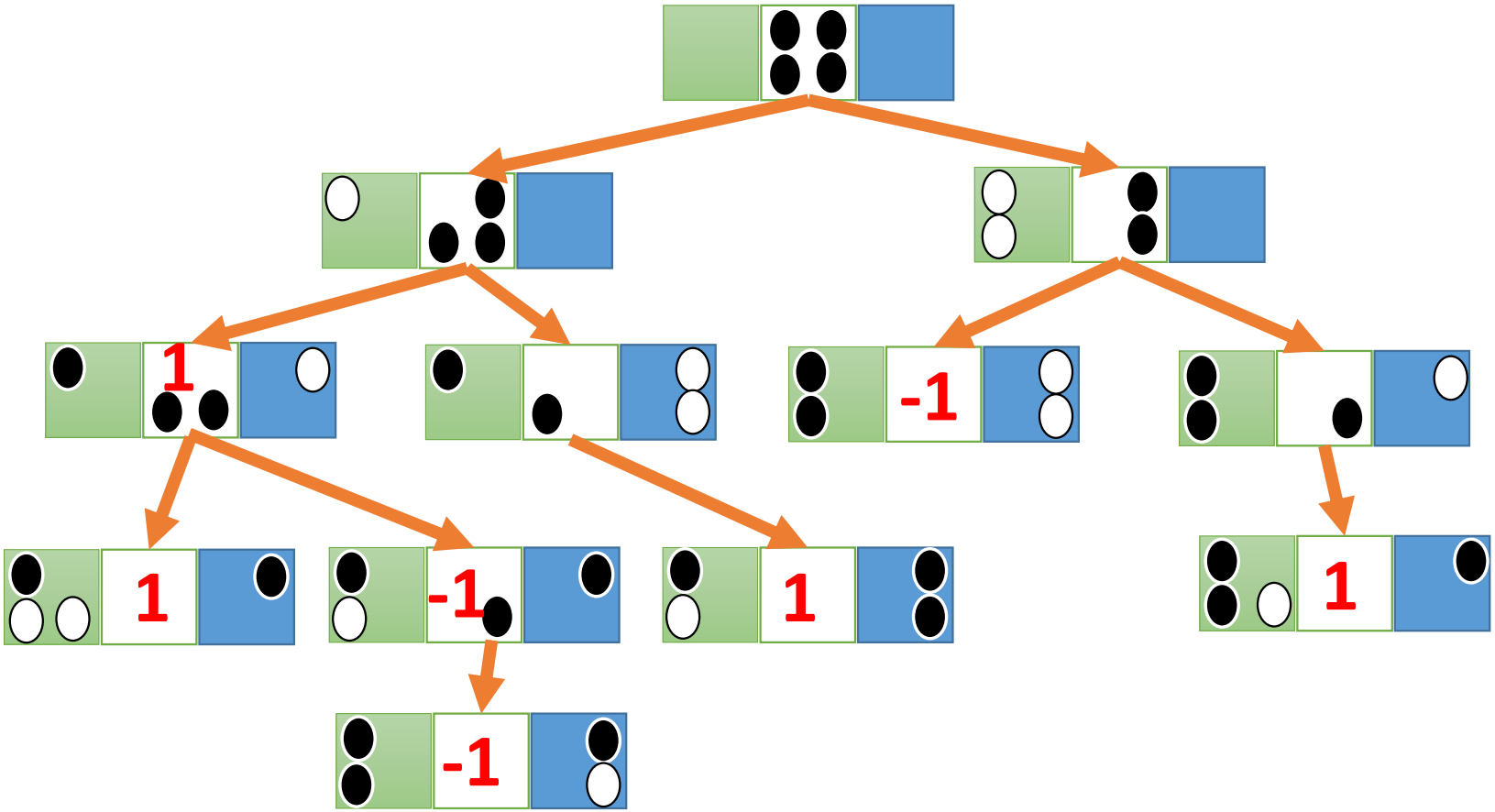
状态枚举



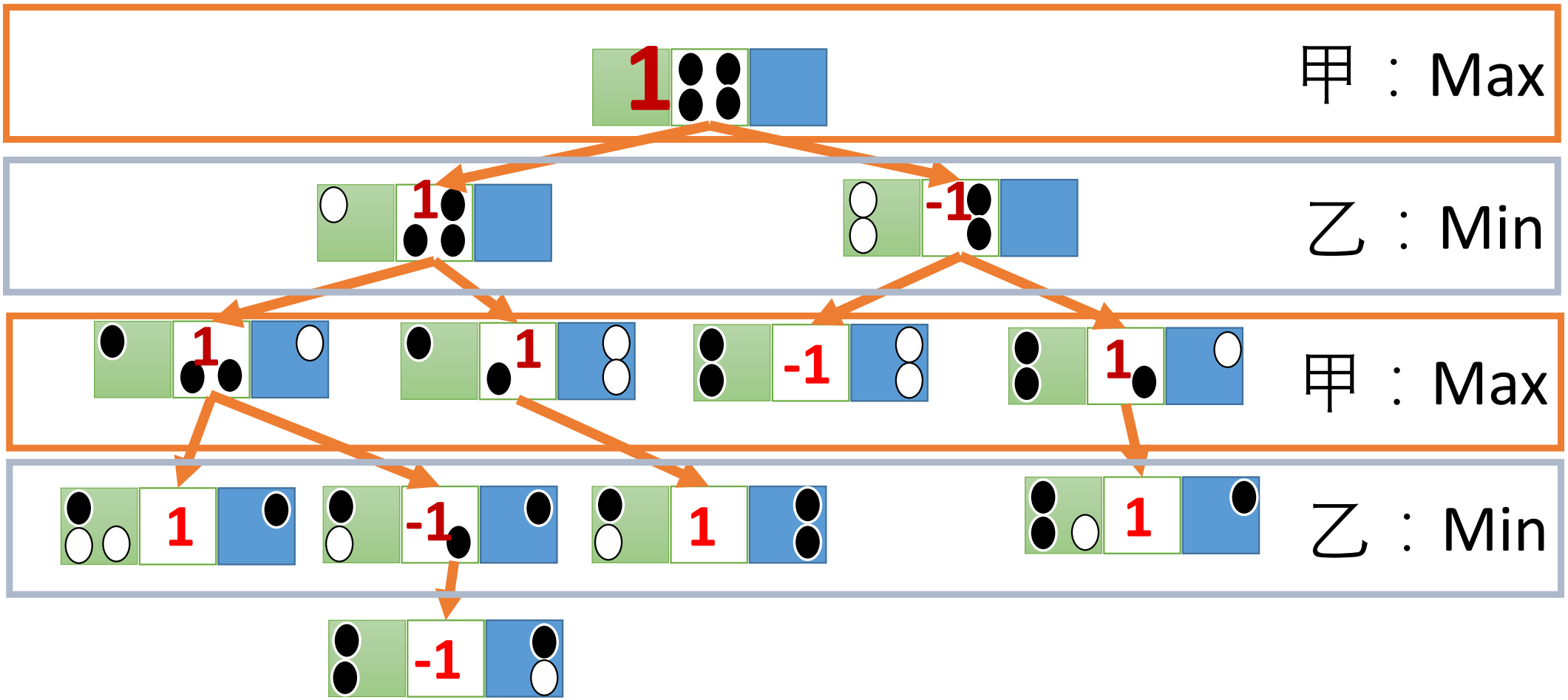
对甲（绿）来说赢是1，输是-1，游戏玩完见胜负！



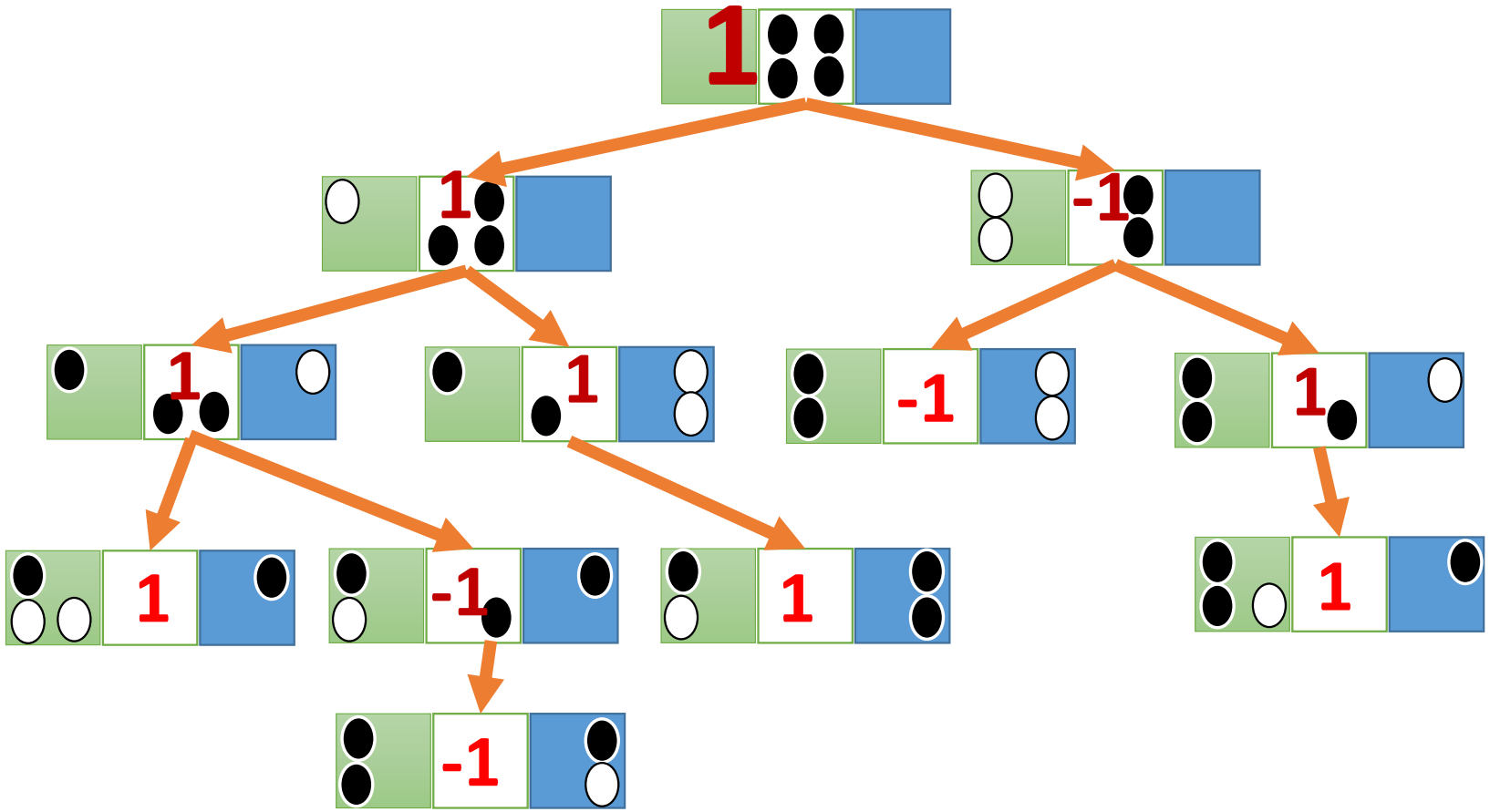
还没玩完，怎么看胜负（理性玩家）？



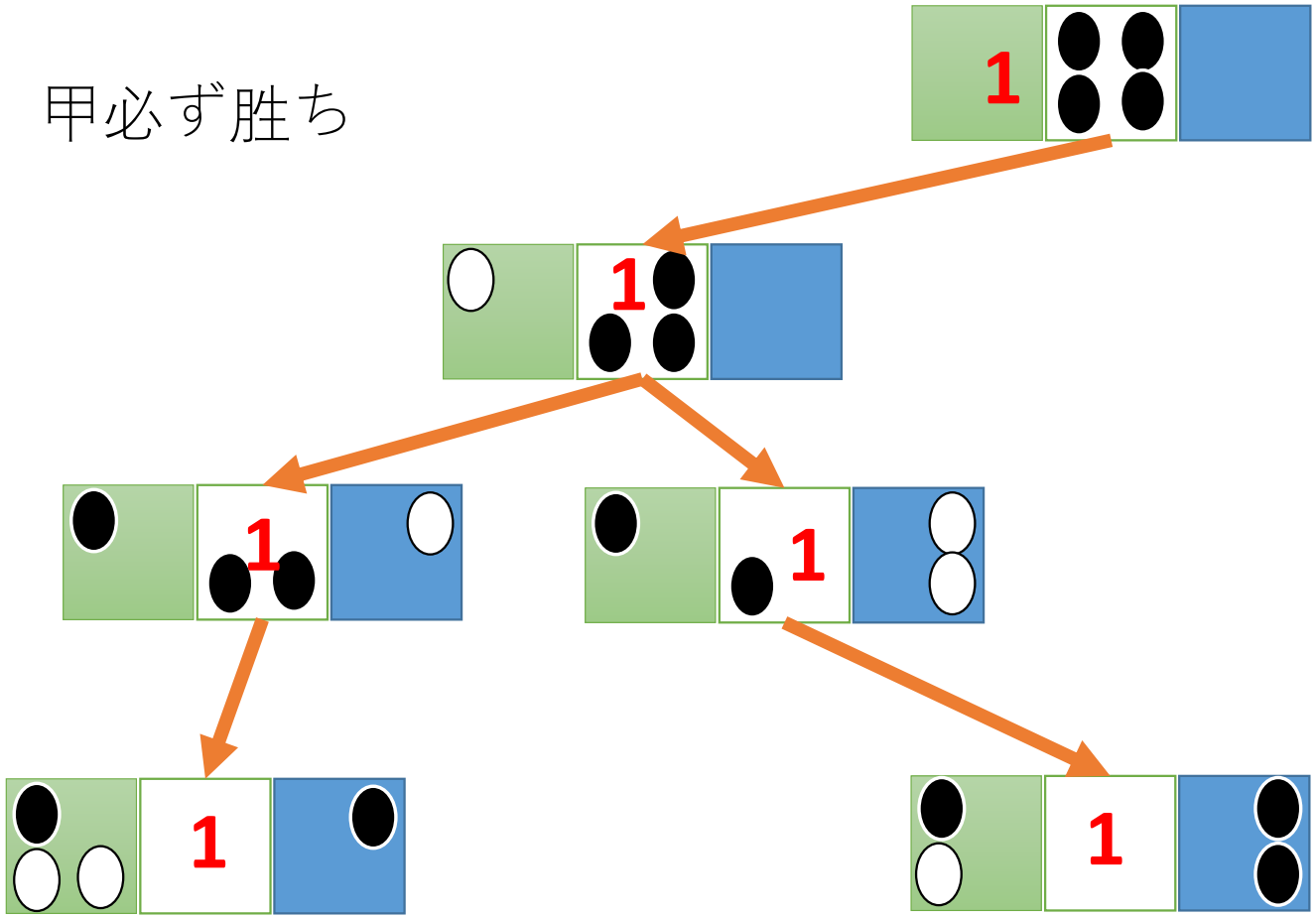
从游戏结束理性反推，甲：争取走向1 乙争取走向-1
MinMax



MinMax结果



甲必ず胜ち

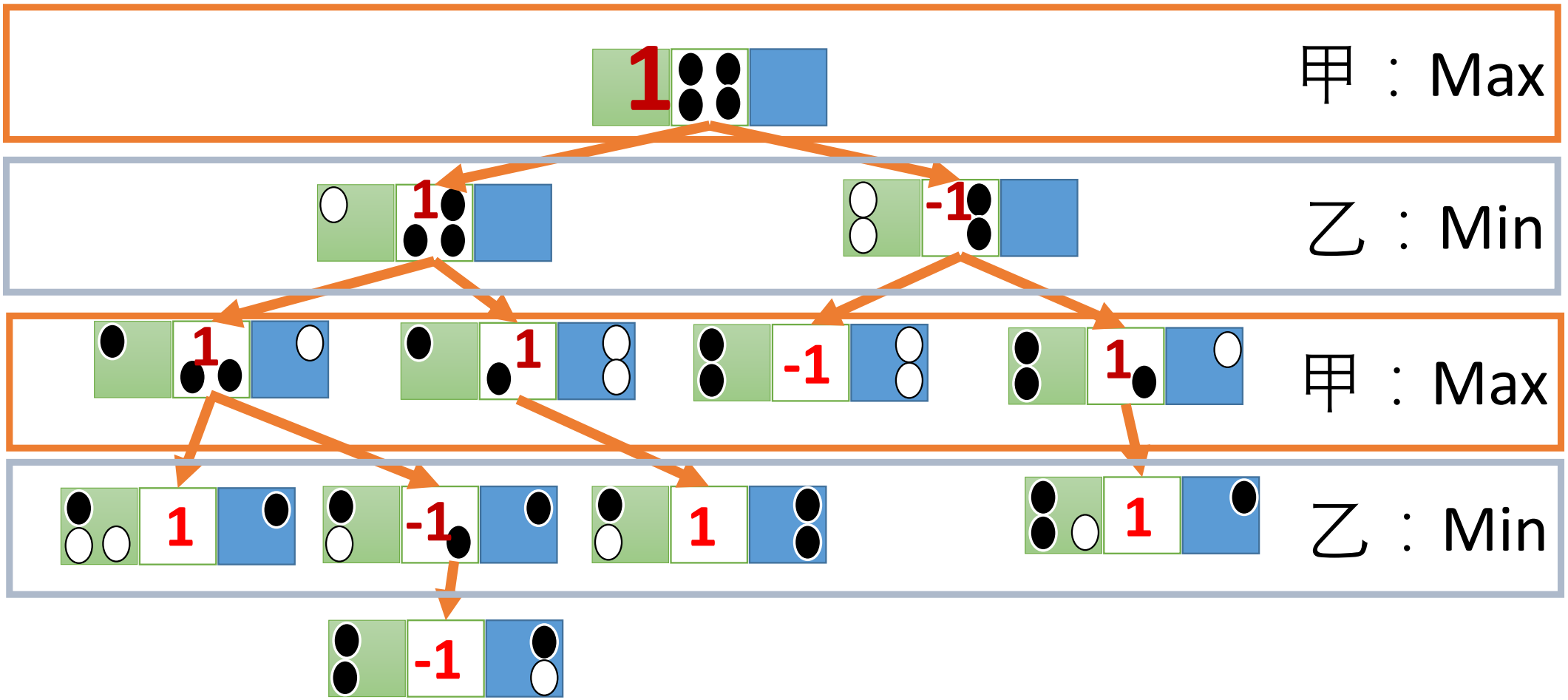


四蛋博弈

→ 有限状态、完美信息、零和、无平局博弈

- 每个状态都是输赢已定
- 每个状态下，当前玩家都有确定的最优策略

学会了什么？



学会了什么

- 到我玩的时候
  - 桌上一个蛋：我都拿了，我赢
  - 桌上两个蛋：我都拿了，我赢
  - 桌上三个蛋：只要对方不犯错，我怎么拿都输
  - 桌上四个蛋（状态）：我只拿一个（行动），我必赢
    - 先手赢

- 艰辛网红路: 郁晓东* → 杨鑫奇、成臣 → 各种协会 → 放弃
  - 万能师姐：看到导数公式放弃
  - 小学生、幼儿园小孩：**能听懂**，认为讲的很无聊，输了26个冰激凌

直播500人→20人 （重新写讲义）
→给数学不扎实的程序员讲明白梯度提升树

$$obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i + t) + \dots$$

$$h_i = \frac{\delta^2 l(y_i, \hat{y}_i)}{\delta^2 \hat{y}_i} \qquad ob$$

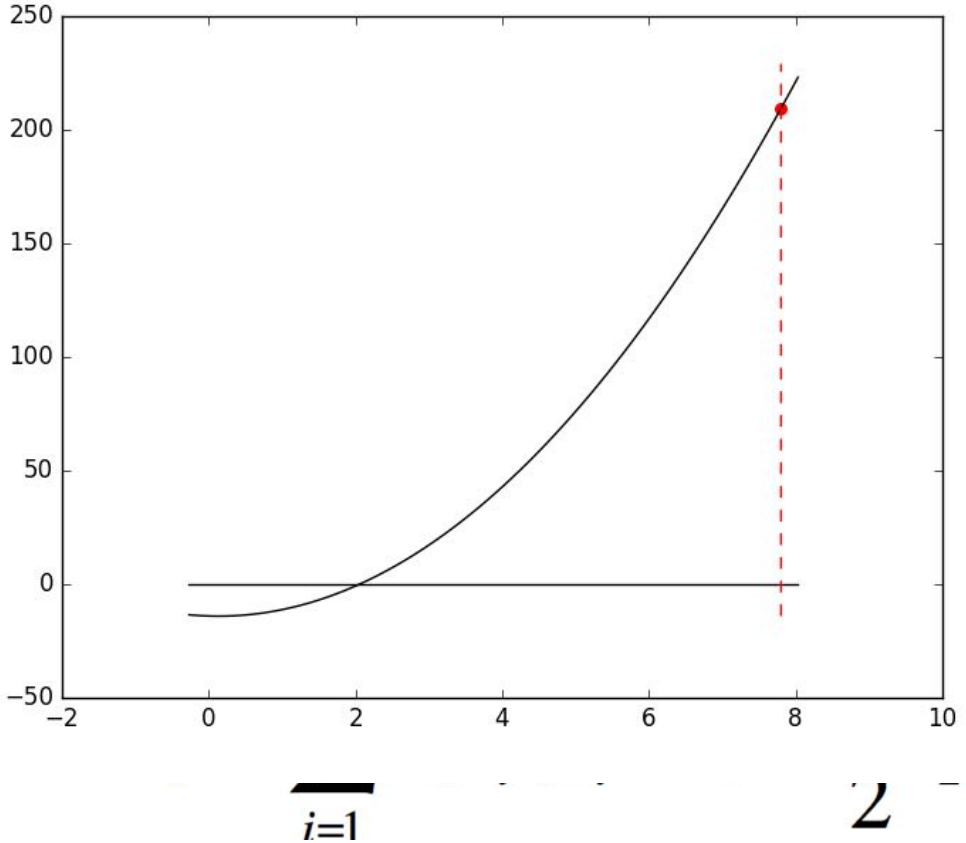
$$H = \sum_{i=1}^{n} h_i$$

$$G = \sum_{i=1}^{n} g_i$$

$$obj \simeq \sum_{i=1}^{n} [l(y_i$$

$$obj = \frac{1}{2}(\sum_{i=1}^{n} h$$

$$obj = \frac{1}{2}(H + \dots$$



$$+ Const$$

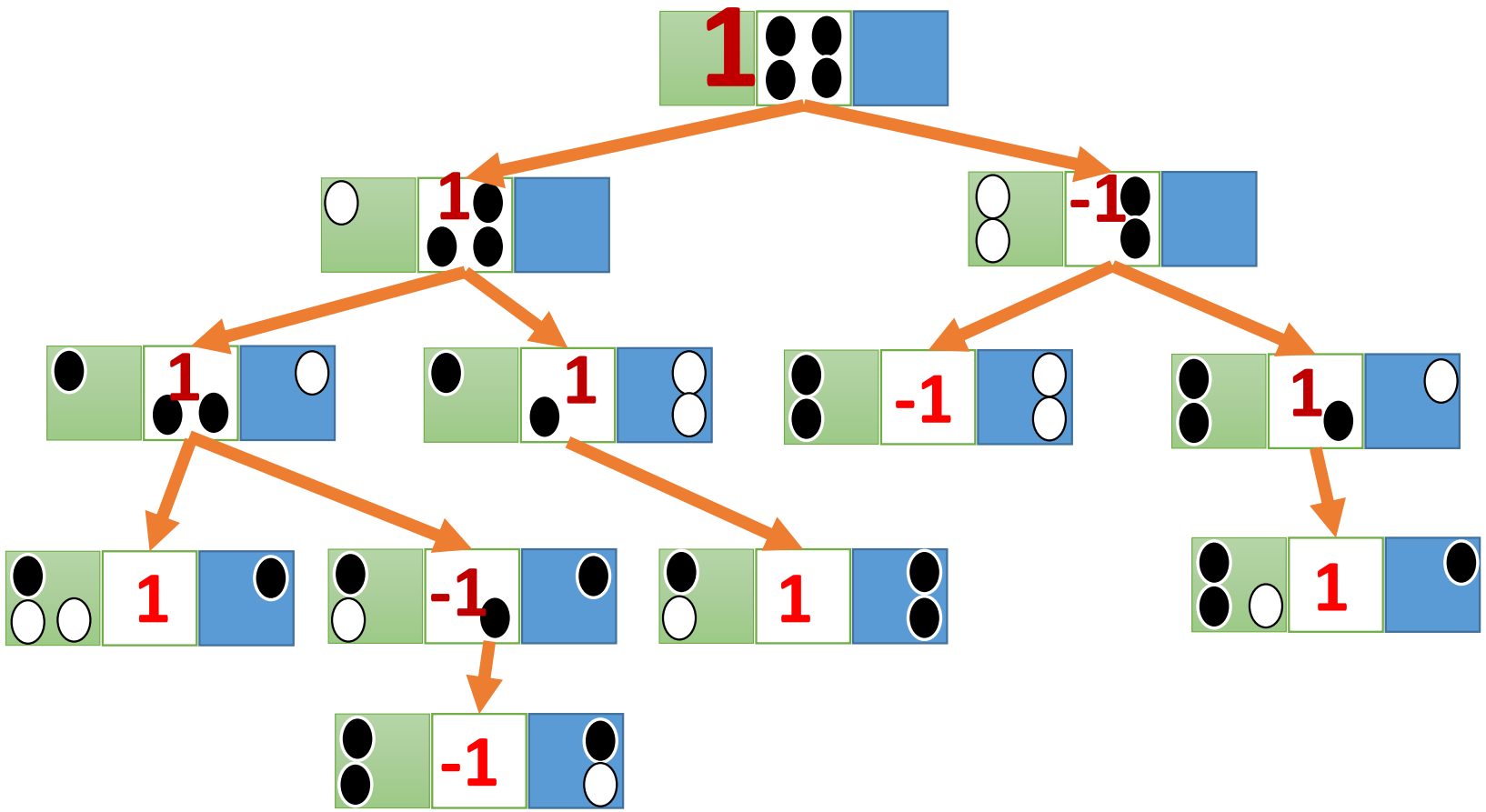
$$\frac{G}{+ l_2}$$

$$\frac{1}{2}\frac{G^2}{H + l_2} + l_1 +$$

$$\frac{1}{2} h_i tt] + \frac{1}{2} l$$

$$+ l_1 + C$$

Why MinMax? 为什么黑蛋白蛋? → → → 引出AlphaGo



四蛋博弈

→ 有限状态完美信息零和无平局博弈

- 每个状态都是输赢已定
- 每个状态下，当前玩家都有确定的最优策略
- **枚举状态，从结局开始，可以通过MinMax手法反推找到当前状态下玩家输赢与当前状态下玩家的最优策略**

围棋、国际象棋

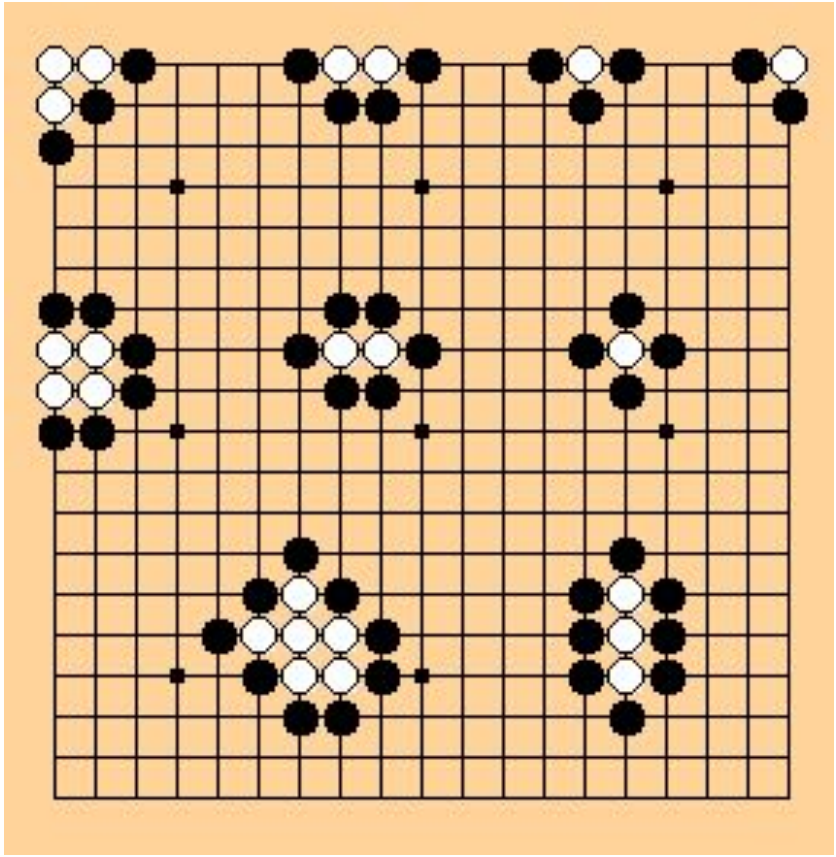
→ 有限状态完美信息零和有平局博弈

围棋

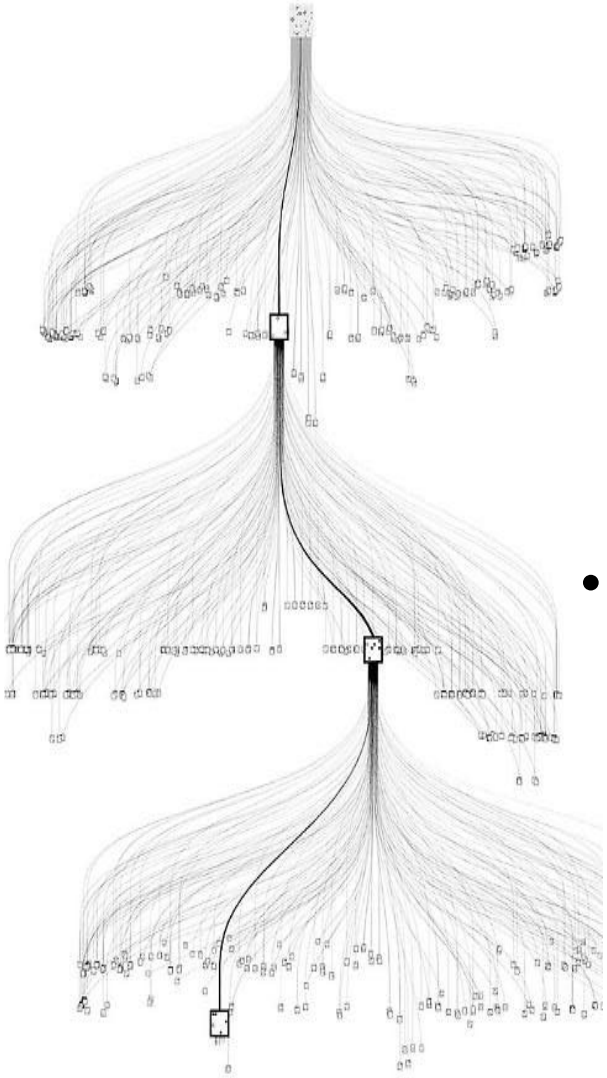
- 小棋盘 （4×4）
  - MinMax

- 19×19棋盘
  - 状态太多。。。

- 19×19=381
  - 2步：300 × 300 = 9万
  - 4步：9万 × 9万 = 81亿

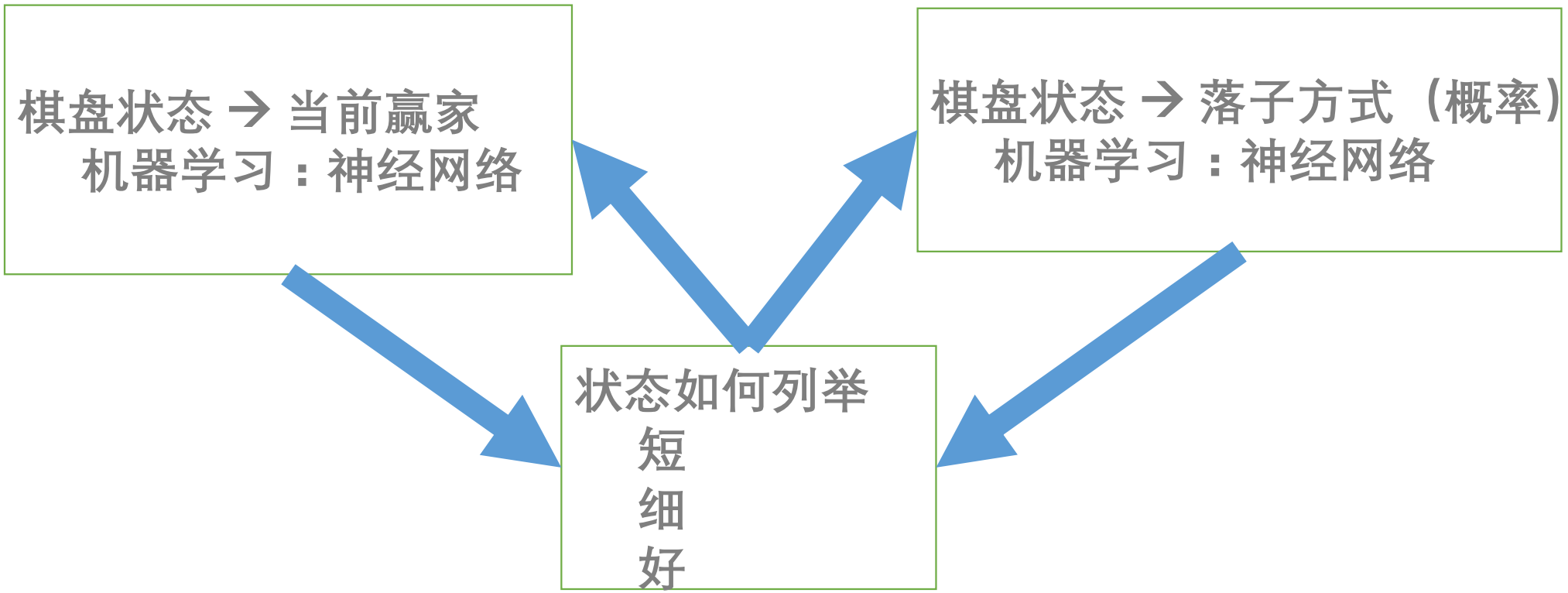


- 解决四蛋博弈问题
  - 从结局推
    - 当前赢的是谁
    - 应该怎么玩
- AlphaGo
    - 当前赢的是谁
    - 应该怎么玩
  - 从结局推不可能！！
    - 状态太多（缓存）
    - 计算量太大
  - 另外想办法



- 变短变细
  - 列举状态的时候有所选择
  - 不要玩到底
- 鸡→蛋 蛋→鸡
  - 如果大概知道怎么玩
    - 选赢概率大的地方去列举
  - 如果大概知道状态下赢的概率
    - 到底前，可以类似MinMax反推

AlphaGo系列方法组成部分



下面是什么？

- 机器学习：神经网络
  - 棋盘状态到输赢判断、棋盘状态到落子策略（概率）
- 状态如何列举
- 怎么有机 **组合** 到一块儿
- 问答

机器学习定义

- 对于某类任务T和（对任务的）性能指标P，一个计算机程序能够从经验E里学习，也就是说，基于经验E，（计算机程序）在任务T上的性能指标P有所提升。 -- Tom Mitchell
    - 学习：从经历（历史数据）里面找到道理，来做的更好
- 机器学习就是不直接编程而让计算机有学习（解决问题）的能力 -- Arthur Samuel
    - 自动从**数据**中发现**规律**，并使用规律**解决问题**

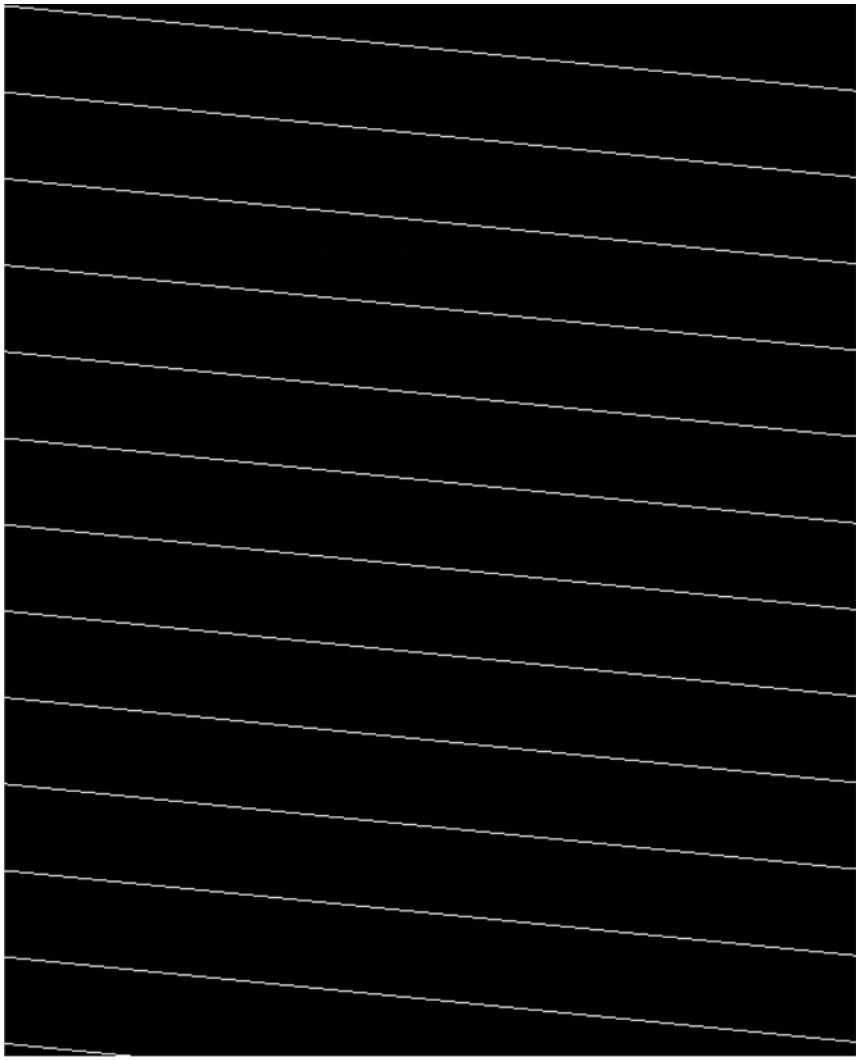
- 机器学习资源推荐
  - Andrew Ng
    - Machine learning https://www.coursera.org/learn/machine-learning
  - 林轩田
    - 机器学习基石 https://www.youtube.com/playlist?list=PLXVfgk9fNX2I7tB6oIINGBmW50rrmFTqf
    - 机器学习技法 https://www.youtube.com/playlist?list=PLXVfgk9fNX2IQOYPmqjqWsNUFl2kpk1U2
  - Google **王博士 机器学习**
  - 数学基础
    - 《深度学习》 第二章到第五章
  - 参考书
    - 周志华
      - 《机器学习》
    - Chapman & Hall
      - *Machine Learning: An Algorithmic Perspective, Second Edition*

•近期进展

- 图片识别（image net）　演示
  - 百万量级图片
  - 1000类
  - 5次错误率: 小于5% （随机猜的错误率 99.5%）

- 语音识别、自动翻译
  - 微信的语音识别
  - Google、facebook等公司的端到端翻译

- 近期进展
  - 游戏
    - Atari



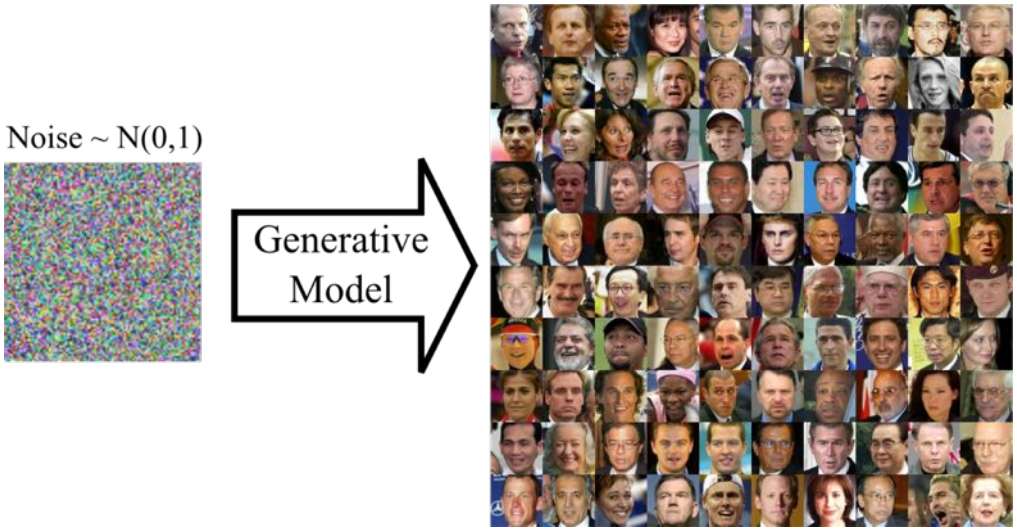
•近期进展

- 写程序

```
/*
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */
static int indicate_policy(void)
{
  int error;
  if (fd == MARN_EPT) {
    /*
     * The kernel blank will coeld it to userspace.
     */
    if (ss->segment < mem_total)
      unblock_graph_and_set_blocked();
    else
      ret = 1;
    goto bail;
  }
  segaddr = in_SB(in.addr);
  selector = seg / 16;
  setup_works = true;
  for (i = 0; i < blocks; i++) {
    seq = buf[i++];
    bpf = bd->bd.next + i * search;
    if (fd) {
      current = blocked;
    }
  }
  rw->name = "Getjbbregs";
  bprm_self_clearl(&iv->version);
  regs->new = blocks[(BPF_STATS << info->historidac)] | PFMR_CLOBATHINC_SECONDS << 12;
  return segtable;
}
```

•近期进展

- 点石成金、变水为油
  - Generative adversarial networks



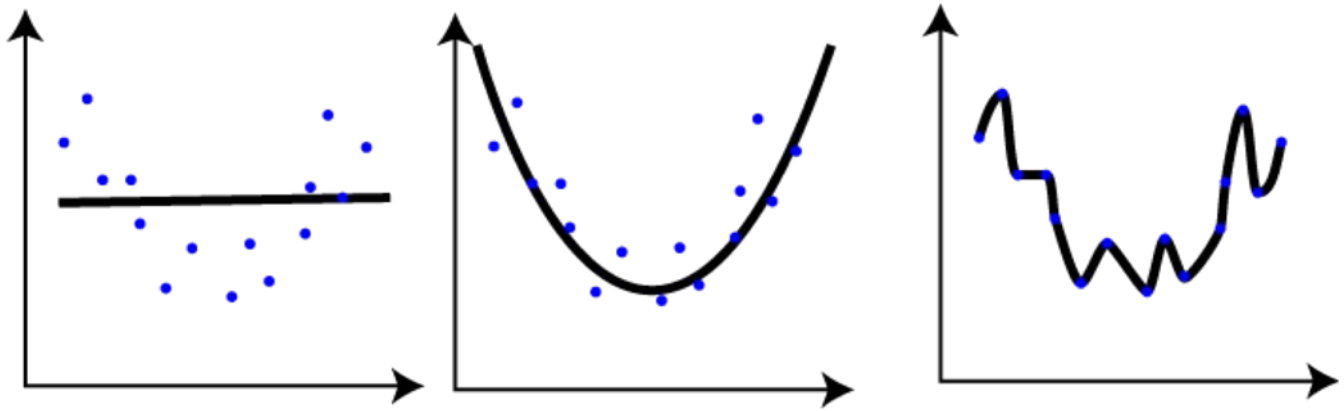
机器学习可完全代替的工作的特性

- 输入明确且输出明确（或反馈清晰）
- 有大量标注数据
- 任务随时间变化不大
- 对错误有一定容忍
- 不需要对决策过程进行解释
- 不需要长逻辑链（长时间记忆）、不需要广泛知识背景

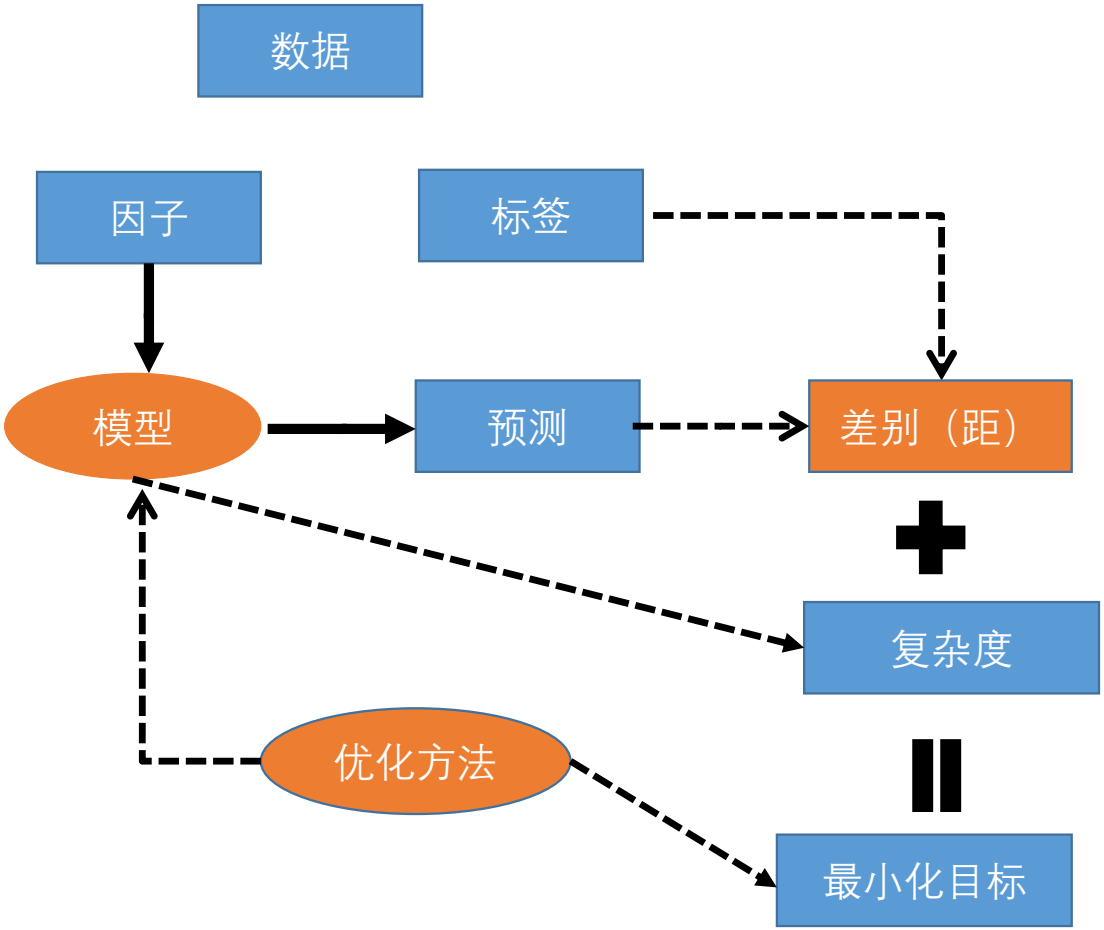
---- 摘编译自 Eric Brynjolfsson

机器学习之以简为美

- 描画 (expression) 能力
- 泛化 (generalization) 能力



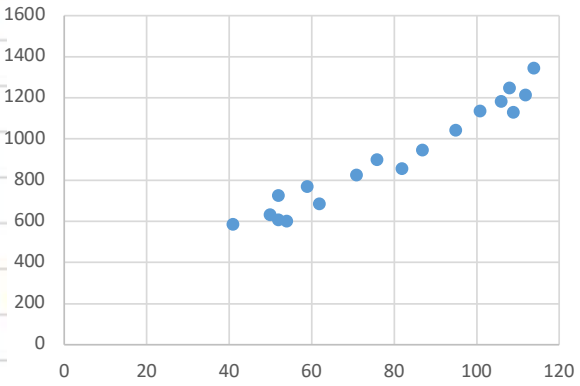
监督学习

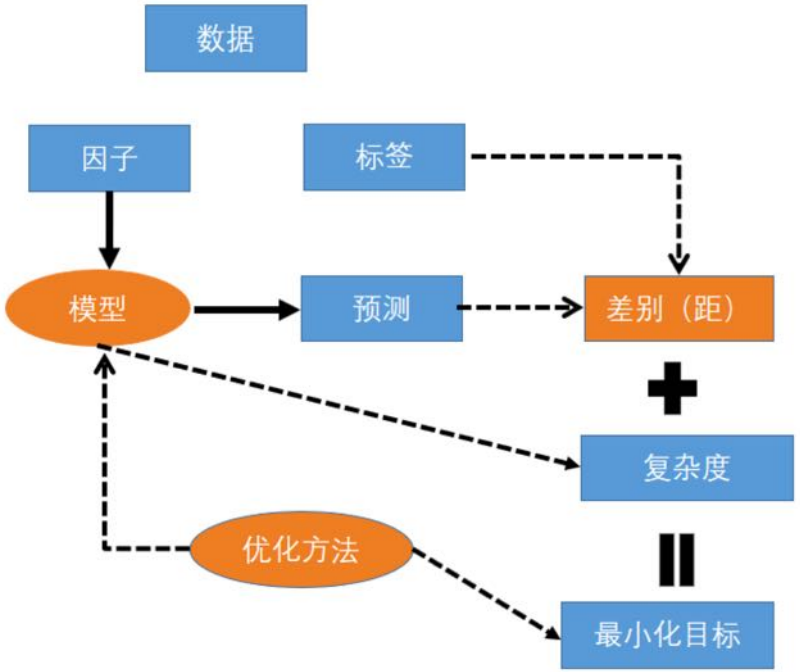


线性回归

数据：因子+标签

房屋面积（特征）	价格（标签）
87	943.2
52	603.5
114	1340.9
71	821.3
76	898.5
54	598
112	1210.3
101	1134.3
41	583.9
82	852.3
50	629.8
62	683.2
59	766.7
109	1127.2
106	1180.9
108	1245.2
52	722.1
95	1041.2





模型

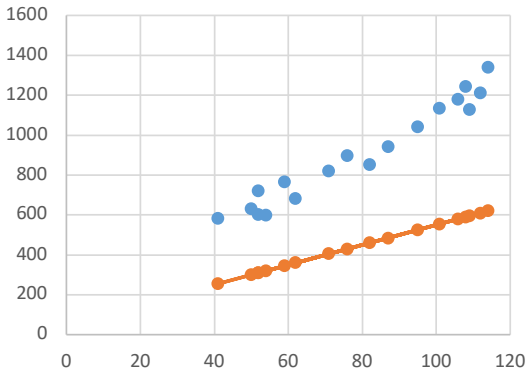
a	b
5	50

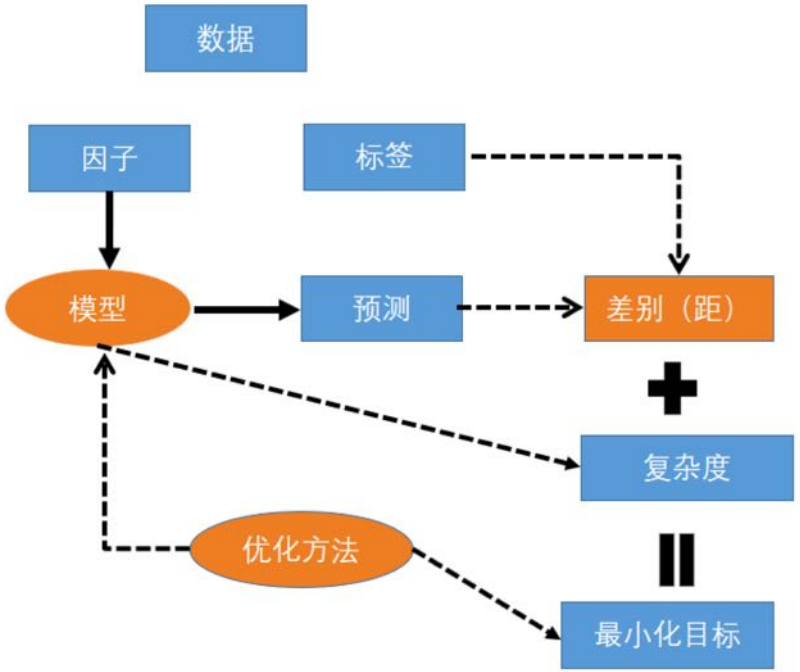
$$y = a * x + b$$

excel演示

模型：价格 = a * 面积 + b		
面积（特征）	价格（标签）	预测
87	943.2	485
52	603.5	310
114	1340.9	620
71	821.3	405
76	898.5	430
54	598.2	320
112	1210.3	610
101	1134.3	555
41	583.9	255
82	852.3	460
50	629.8	300
62	683.2	360
59	766.7	345
109	1127.2	595
106	1180.9	580
108	1245.2	590
52	722.1	310
95	1041.2	525

仅有的几个公式之一

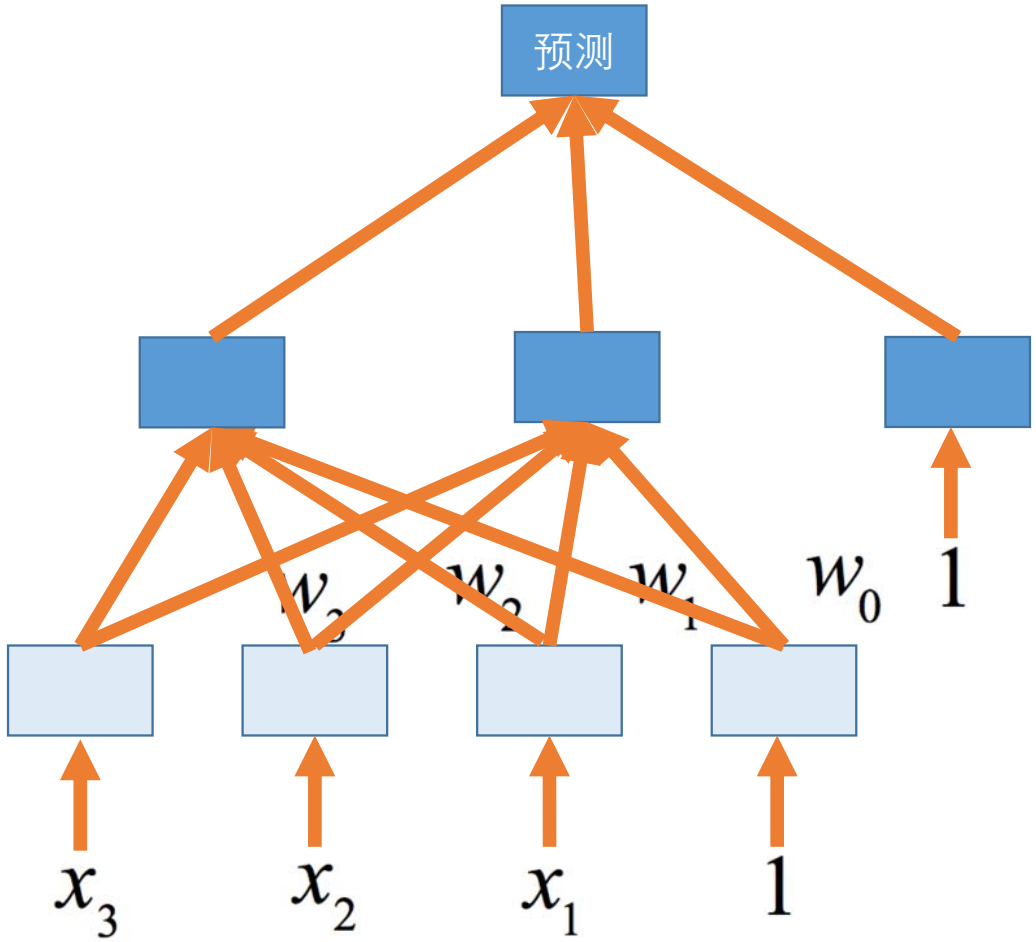




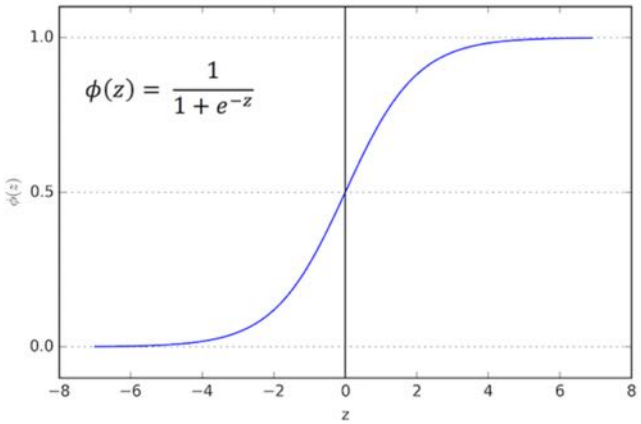
优化器

- 根据模型的参数对差别（差距）的影响进行调整
  - 梯度（导数）

神经网络之多层感知机

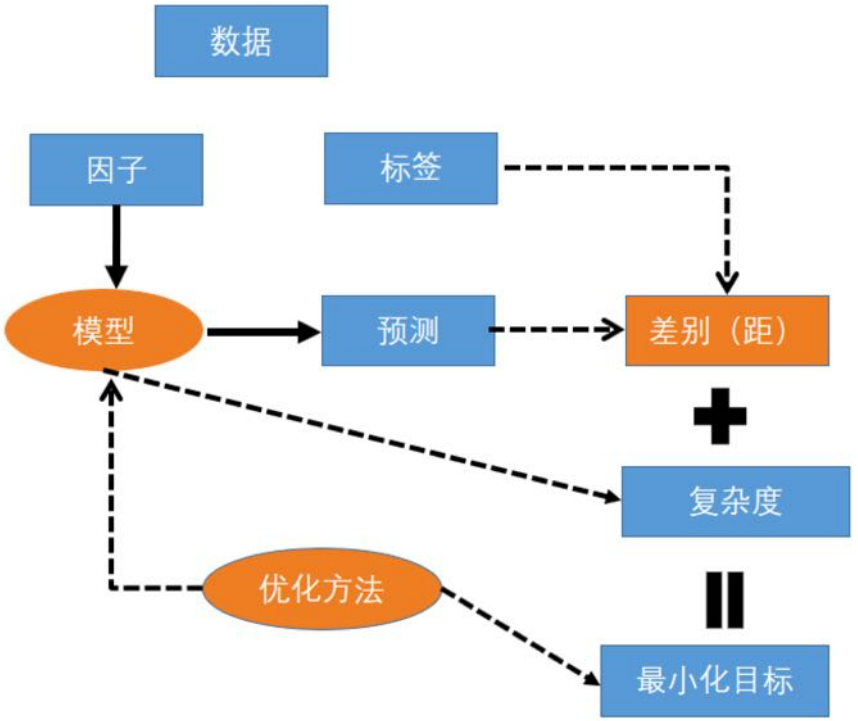

预测
w_3
w_2
w_1
w_0
1
x_3
x_2
x_1
1

$$w_1 \cdot x_1 + w_0$$


$\phi(z) = \frac{1}{1+e^{-z}}$
1.0
0.5
0.0
$\phi(z)$
−8
−6
−4
−2
0
2
4
6
8
z

神经网络基本组成部分

- 乘法
- 加法
- 非线性激活函数



多层感知机

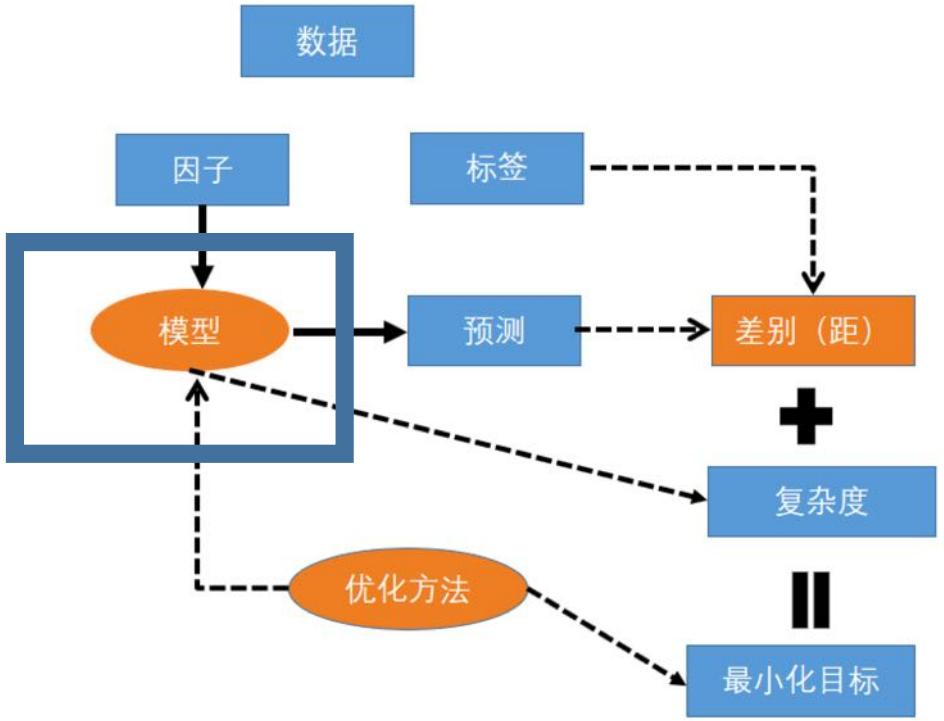
- 如同线性模型一样求解
  - 从预测与标签的差别（距）开始，算出模型每个参数对其影响，对参数进行优化

神经网络之卷积神经网络

卷积神经网络

- 卷积（convolution）

- 最大池化（max-pooling）



神经网络基础：乘法、加法、激活函数、构造（卷积、池化）

- 卷积的初衷

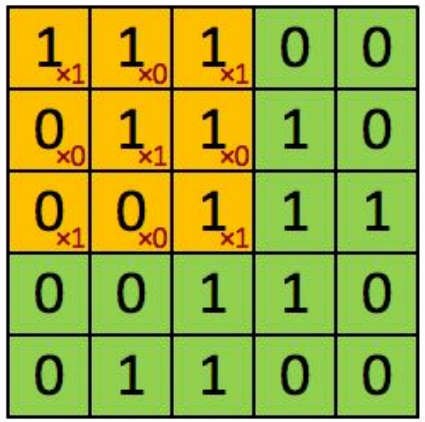
  - 对图像来说，全连接参数太多
    - 比如200 × 200的图片，一个神经元需要40,000个参数

    - 图片上距离比较远的像素的直接关系并不强烈

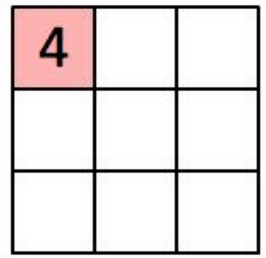
卷积：乘法和加法（矩形区域、移动进行）

需要学习的参数

1	0	1
0	1	0
1	0	1



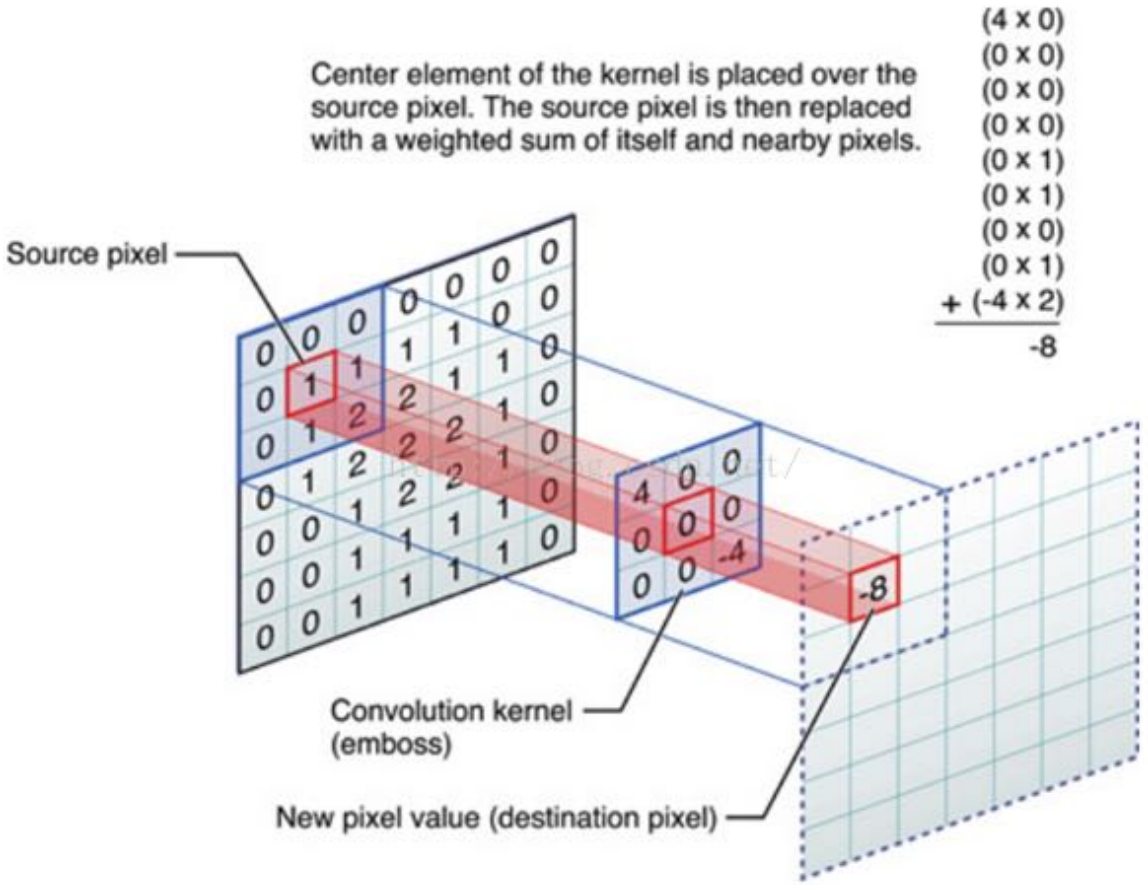
Image



Convolved Feature

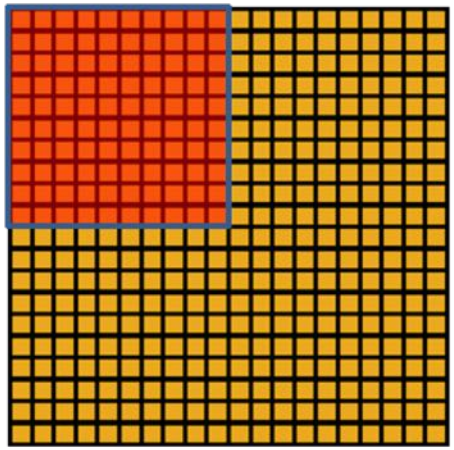
- 普通神经元：多对一
- 卷积神经元：多对多（5×5）--（3×3）卷积 --> （3×3）
  - 参数为9个

卷积：乘法和加法（矩形区域、移动进行）

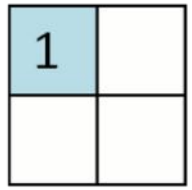


Max-pooling

- 取区域最大值



Convolved feature

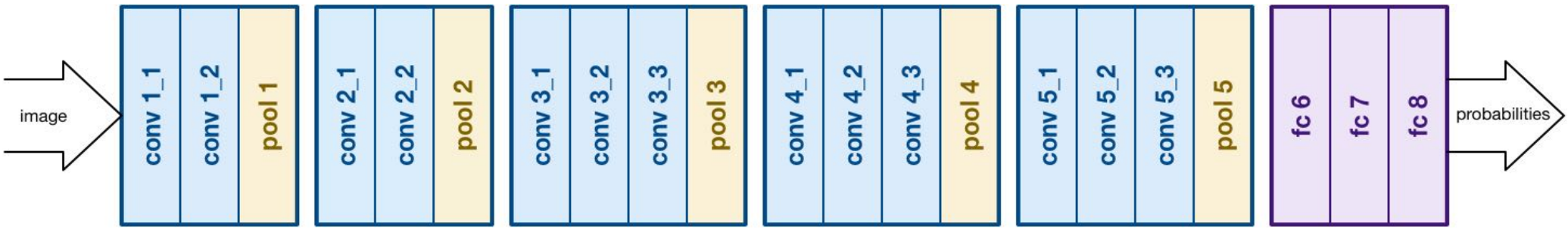


Pooled feature

卷积神经网络

- 卷积：
  - 四处寻找有用模式
- Max-pooling：
  - 找到有用模式的高点

VGG NET 16

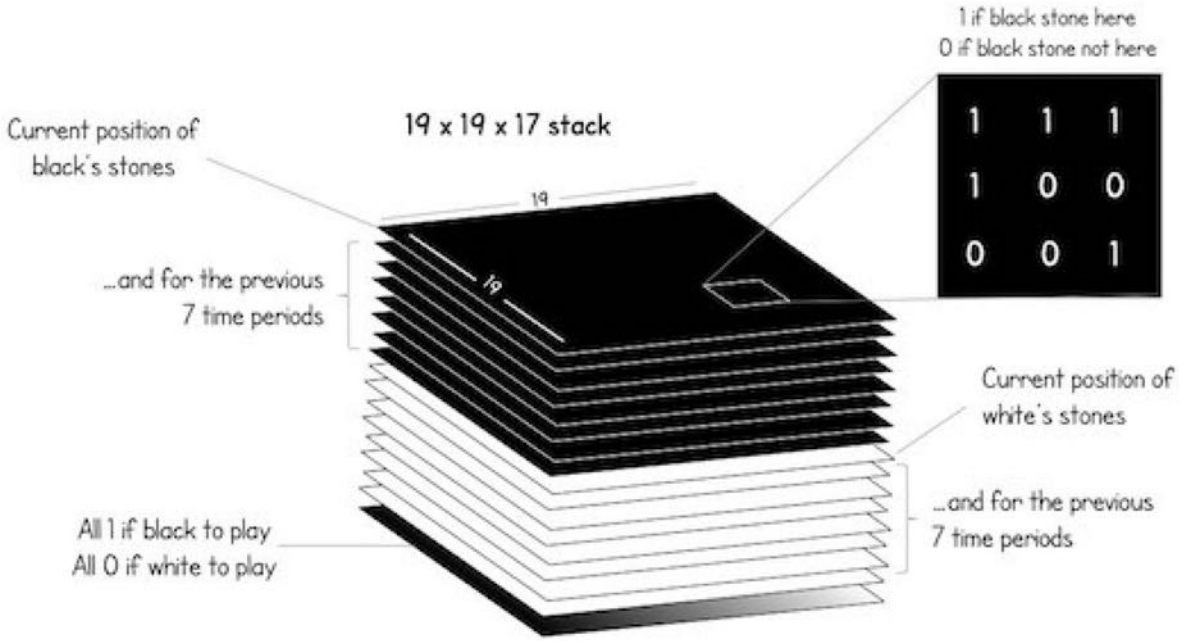


卷积神经网络求解

- 与线性模型以及多层感知机一样

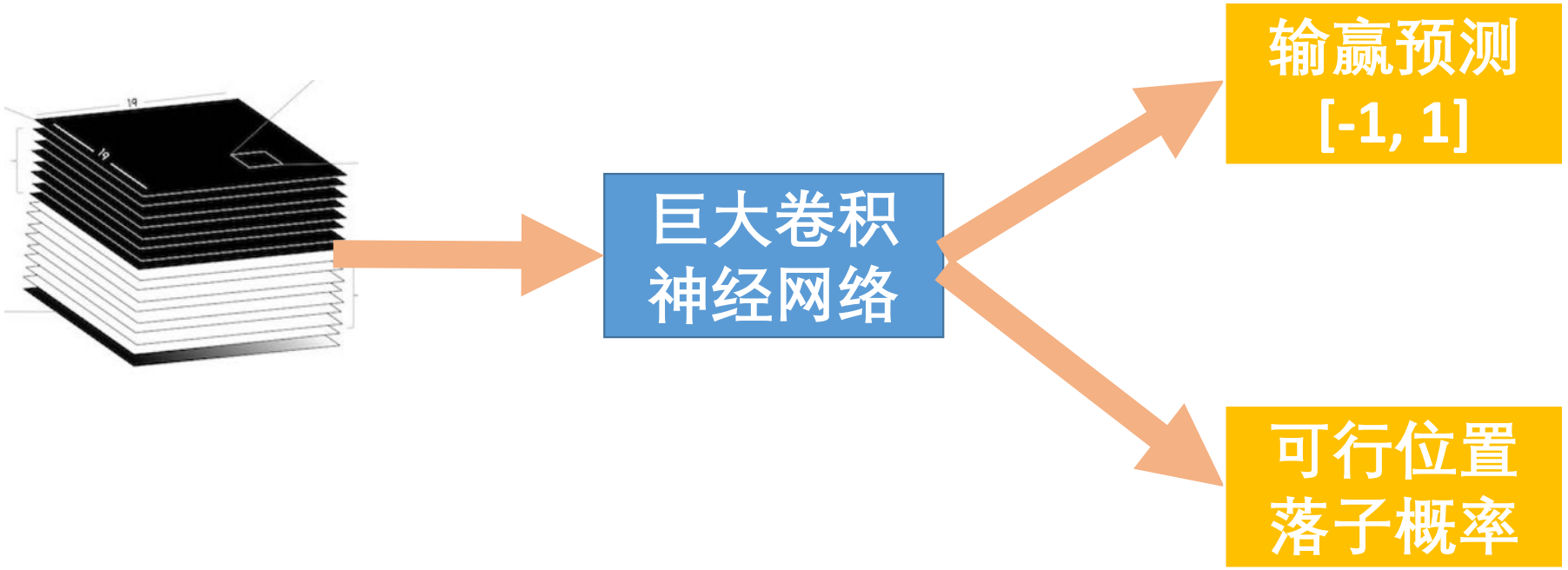
AlphaGo

- 以卷积神经网络来从棋盘映射到输赢预测以及落子策略（概率）



AlphaGo

- 以卷积神经网络来从棋盘映射到输赢预测以及落子策略（概率）

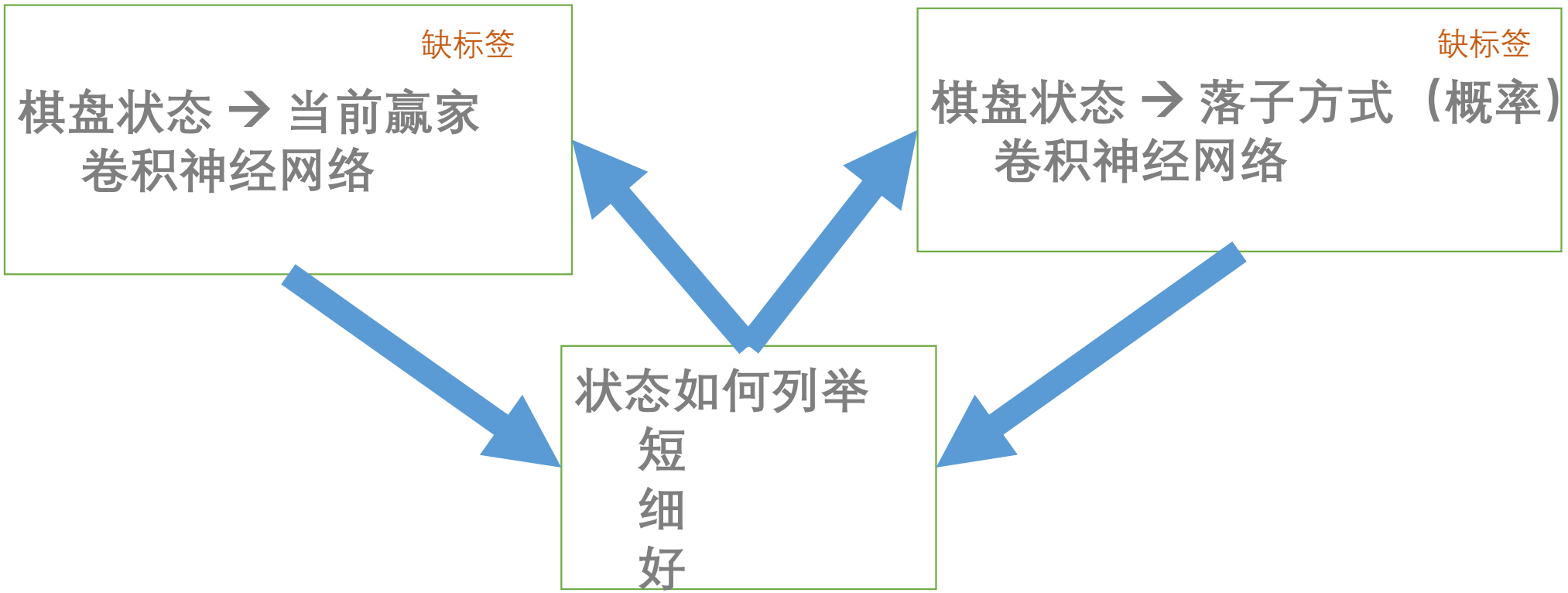


不论输赢预测还是落子策略
缺标签啊

- 人工标？
  - 太耗费时间
  - 标的水准堪忧

- 回想四蛋博弈
  - 通过结局反推
    - 得到状态下输赢情况
    - 状态下最优策略

AlphaGo系列方法组成部分



- 娶太太的例子

- 与环境进行积极交互
  - 活用知识（exploitation）
  - 探索（exploration）

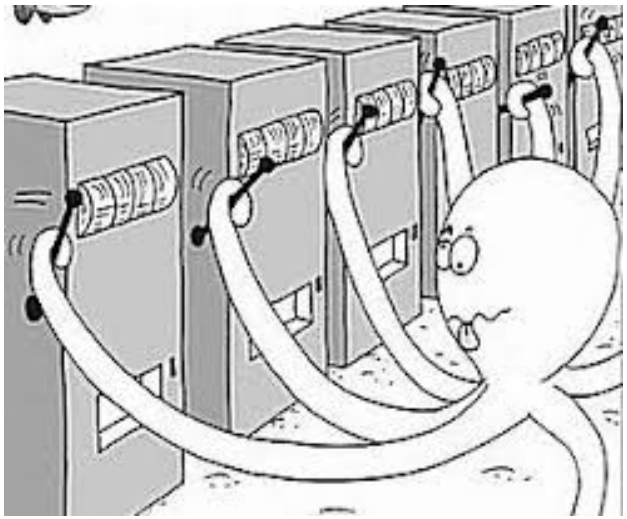
上置信区间

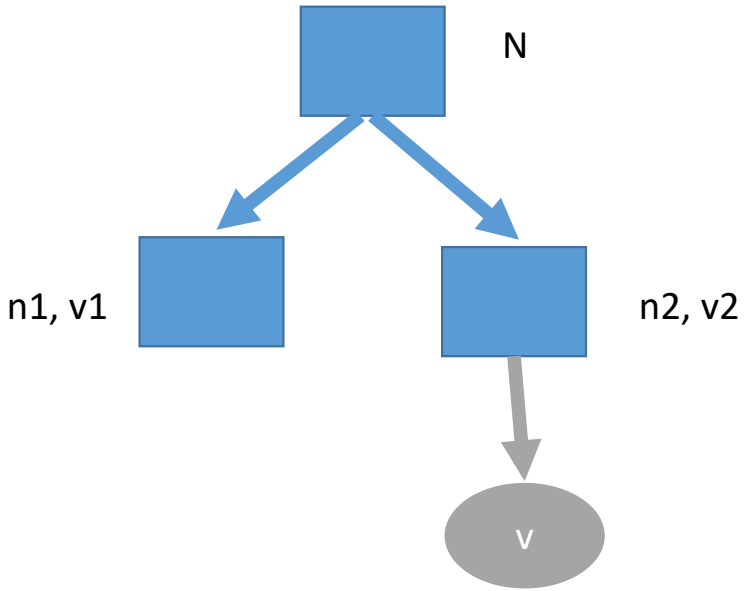
$$v_i + C \times \sqrt{\frac{\ln N}{n_i}}$$

- 回报高
- 试的少

活用知识、勇于尝试

- 吐钱机器
- 非完美信息

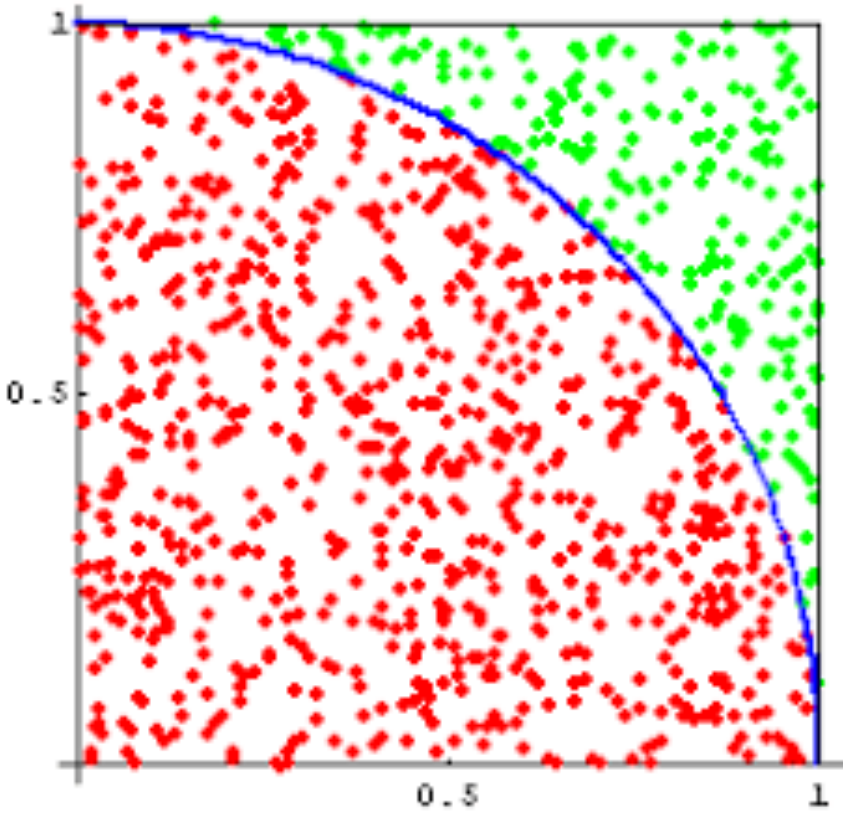




https://homes.di.unimi.it/~cesabian/Pubblicazioni/ml-02.pdf

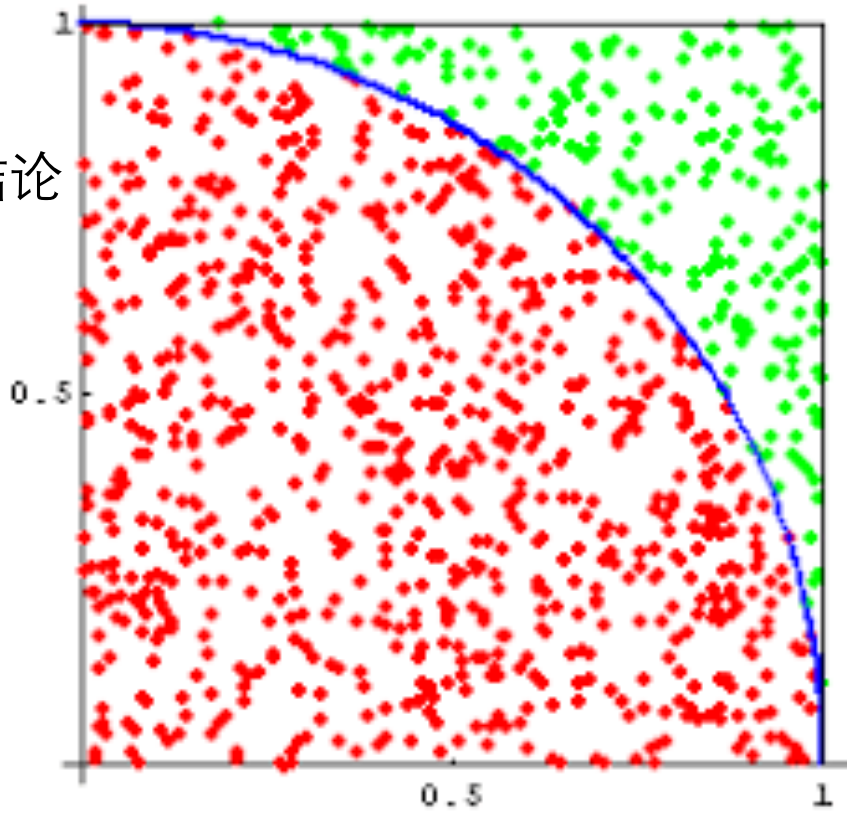
AlphaGo式 蒙特卡罗树搜索 （MCTS, Monte Carlo Tree Search）

- 蒙特卡罗算 π
- 在矩形内随机产生点
- 红点在¼ 圆内
- ¼圆的面积是 ¼ π， 矩形面积是1
- 红点数 / 总点数 就是 ¼ π

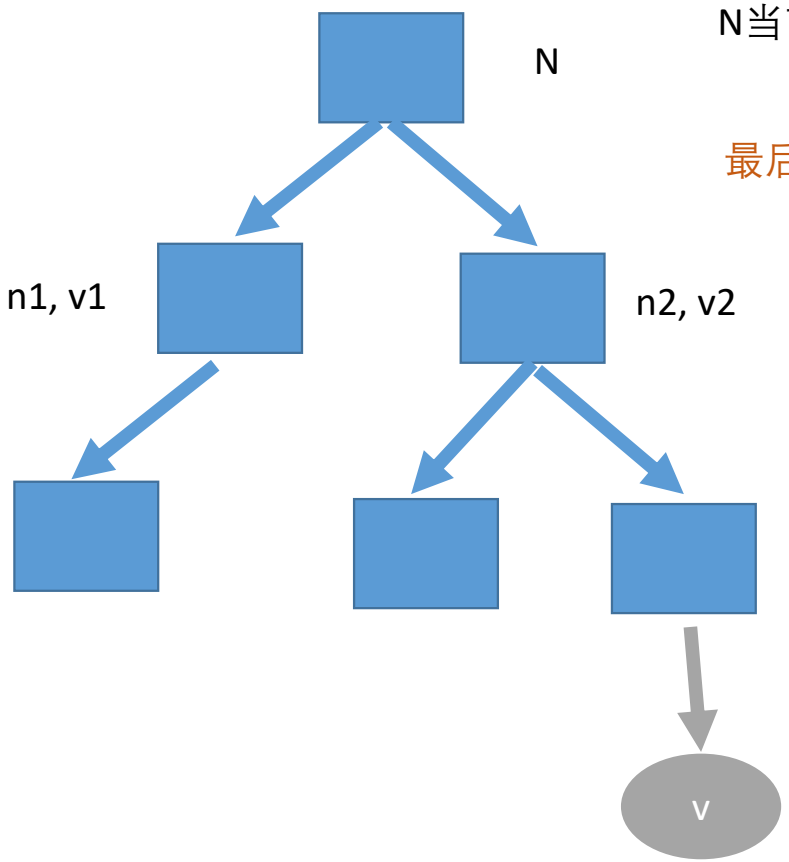


AlphaGo式 蒙特卡罗树搜索（MCTS）

- 蒙特卡罗要义
  - 通过大量简单模拟的“平均”，逼近复杂结论



AlphaGo式 蒙特卡罗树搜索（MCTS）



V平均胜利分数，p落子概率（神经网络给出），n各子状态访问次数，
N当前状态总访问次数

最后一个公式

$$v_i + C * P_i \frac{\sqrt{N}}{1 + n_i}$$

当前（根）状态，选落子位置使上式最大

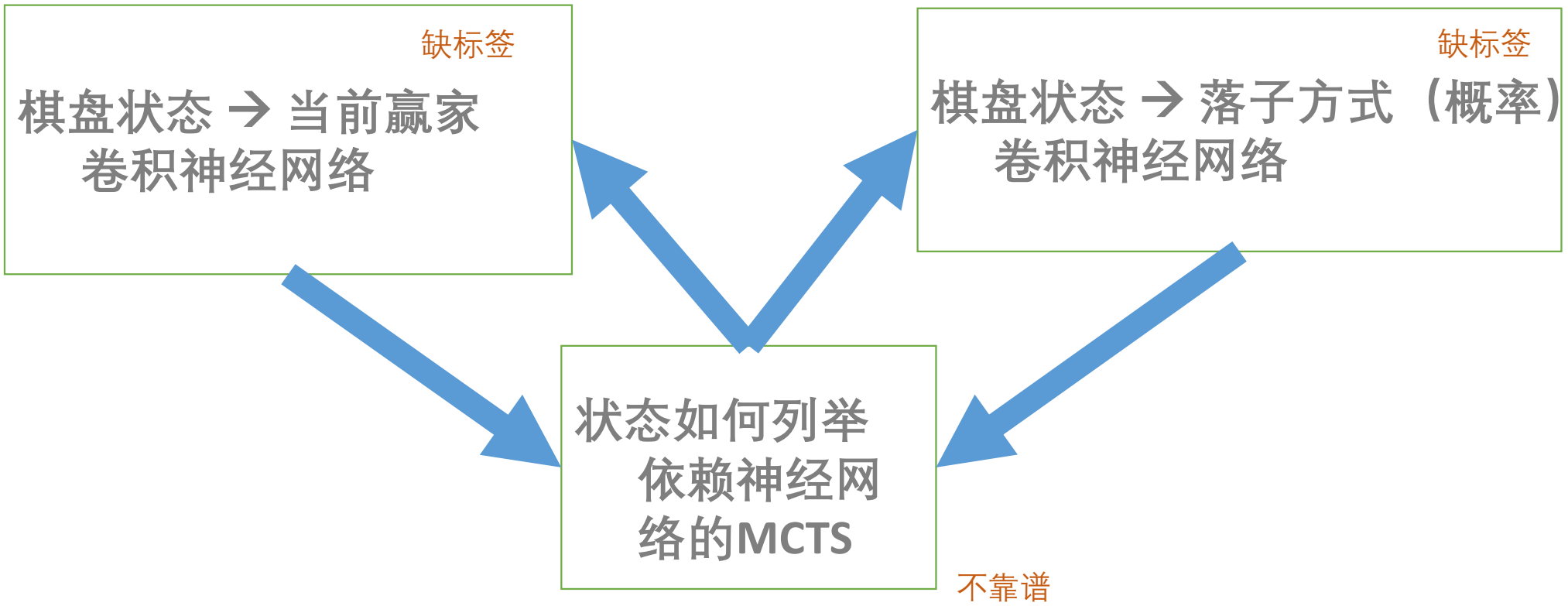
到达下个状态，继续同样落子

- 从根状态仿真1600次，每次仿真不超过40步
  或棋局结束

得到结束状态的值，更新经过状态的v、n、N

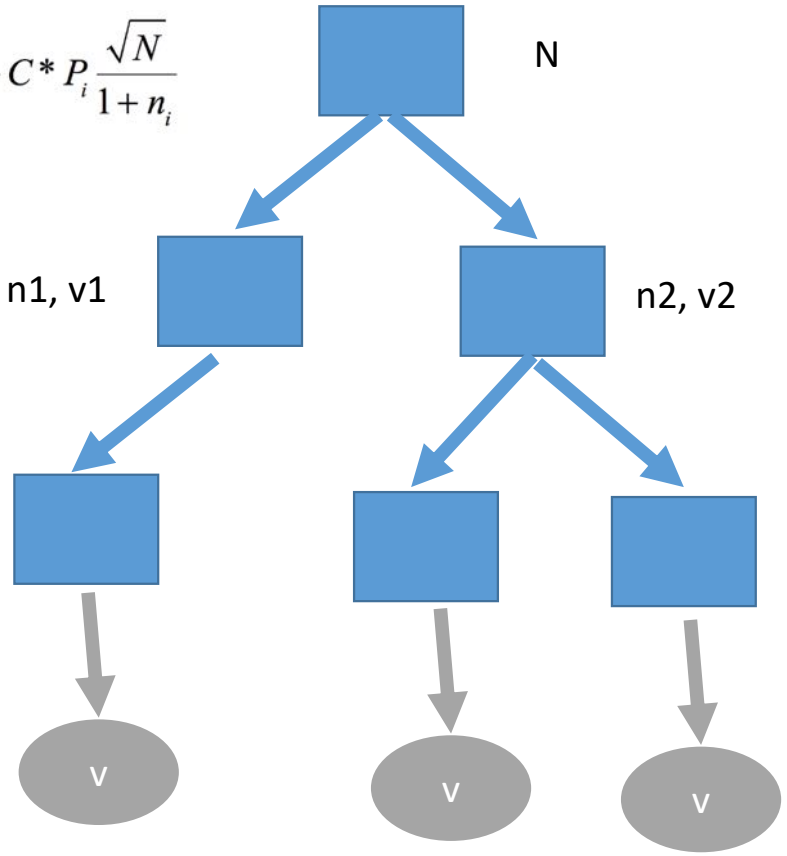
- 棋局没结束，神经网络给出

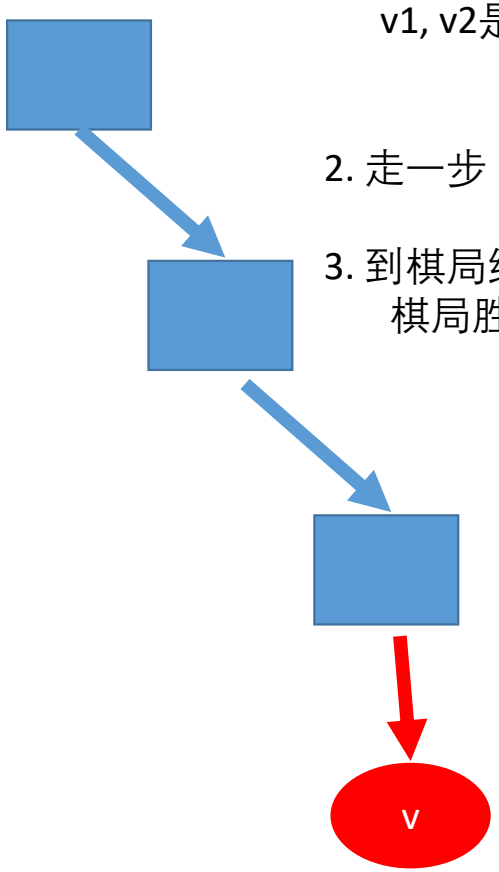
AlphaGo系列方法组成部分



攒一起

$$v_i + C * P_i \frac{\sqrt{N}}{1+n_i}$$





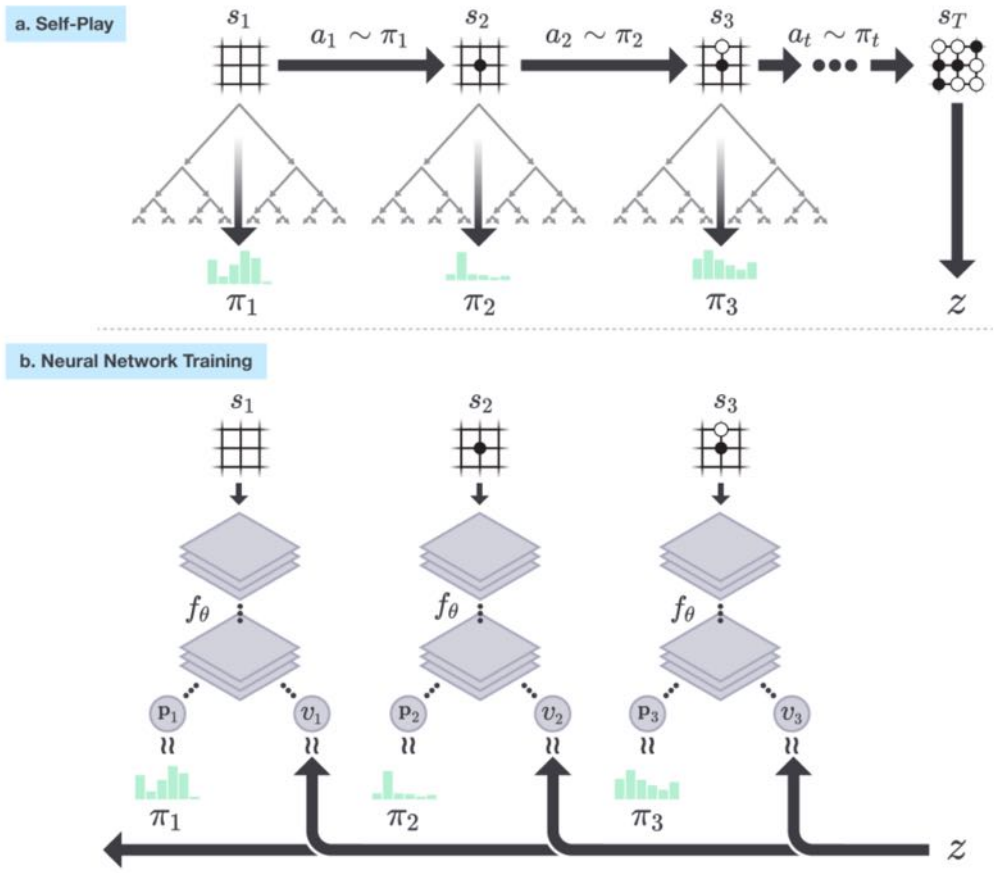
1. 每步1600次仿真
   (n1/N, n2/N)作为状态下落子策略的标签
   v1, v2是蒙特卡罗仿真的结果，丢掉

2. 走一步：n1/N概率选1 n2/N概率选2

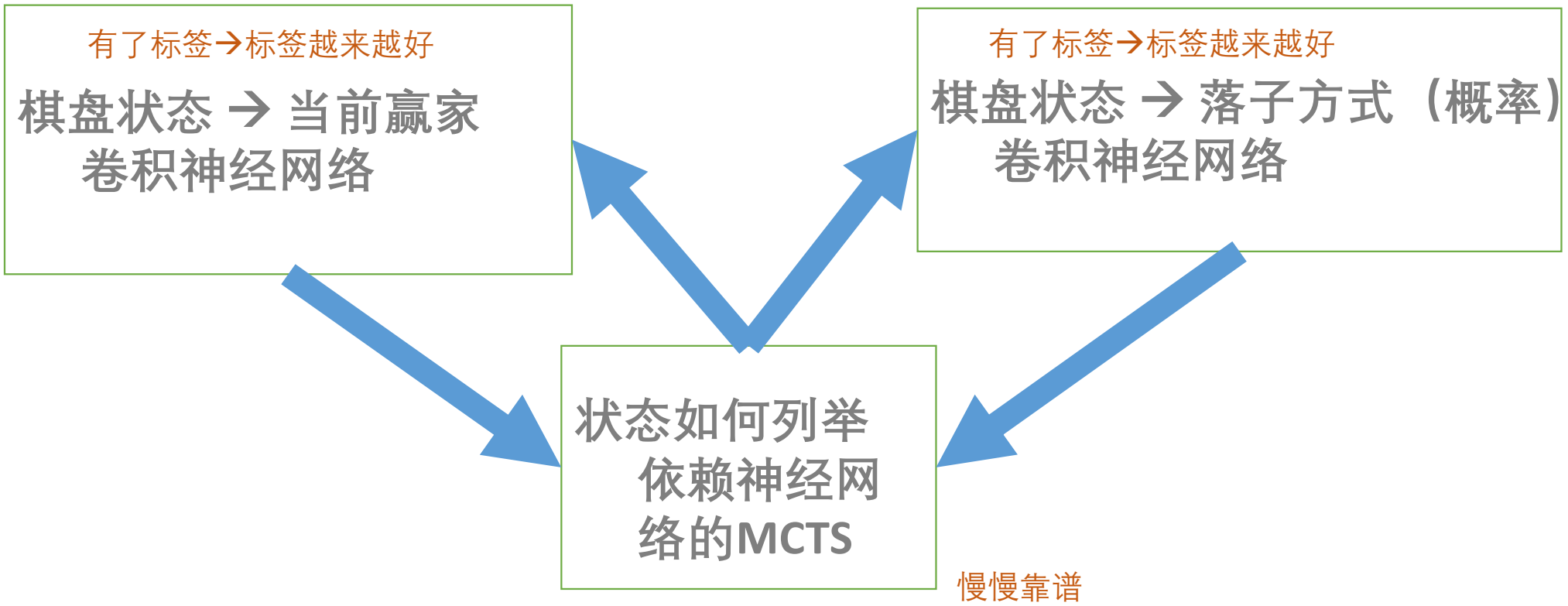
3. 到棋局结束：
   棋局胜负作为经历的落子状态的胜负

攒一起

- 当前状态MCTS搜索的比例作为状态下落子方式的标签
  - 2，3，3, → 0.25, 0.375, 0.375
- 棋局结束，棋局胜负作为所经过状态输赢判断的标签



AlphaGo各部分：等跑起来

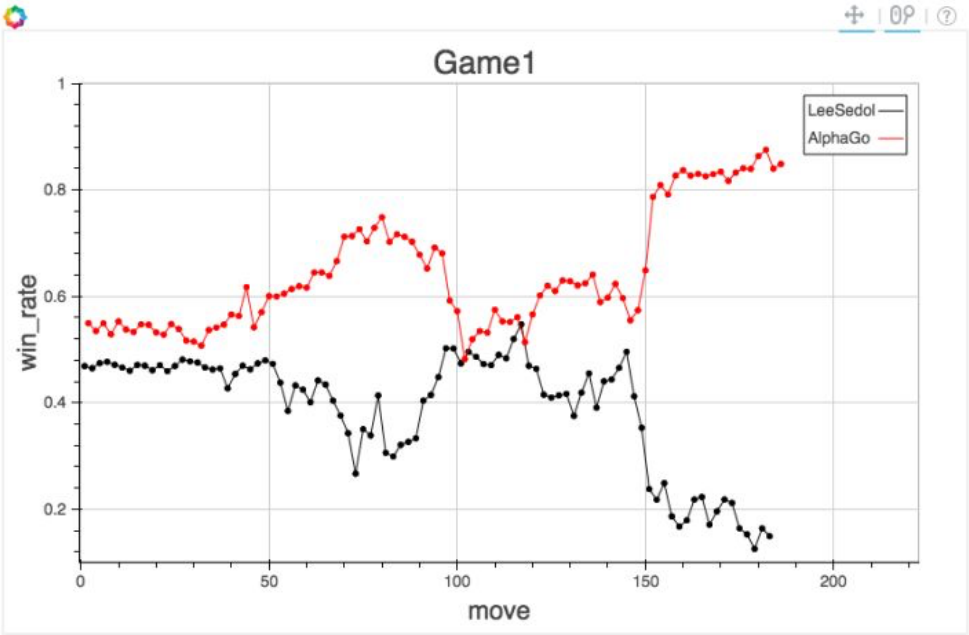


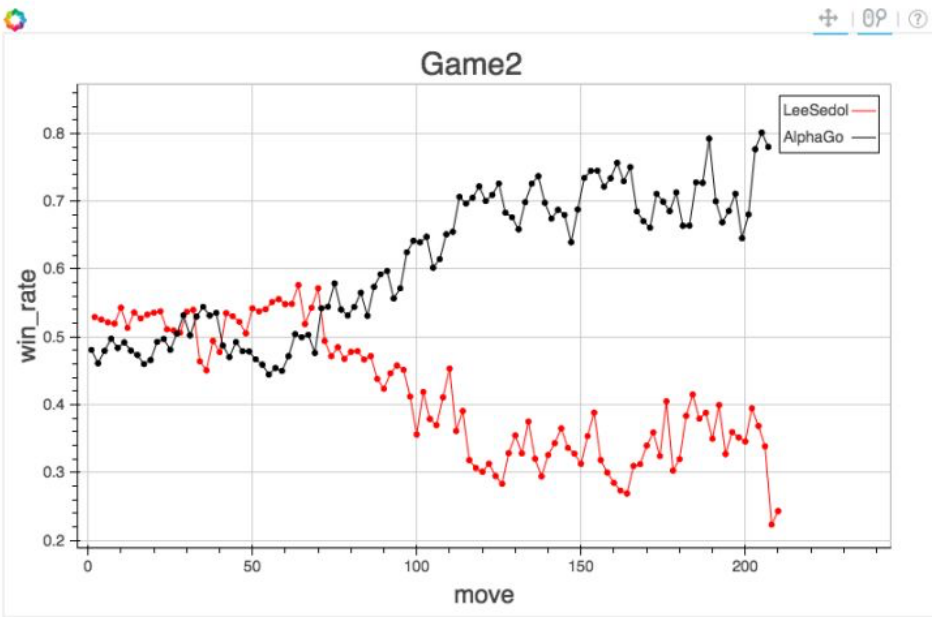
思考

- AlphaGo是先学会了官子，还是先学会了开局

思考

- AlphaGo是先学会了官子，还是先学会了开局





20蛋博弈
每次最多4
凑5而胜

pitybea AlphaEggZero
https://github.com/pitybea/alphaeggzero

神经网络
胜负判断

蛋少是开局吗？

参数：

仿真20次
仿真深度2
三层双头神经网络：

剩余蛋数→（拿蛋策略softmax，胜负判断tanh）

20蛋博弈
每次最多4
凑5而胜

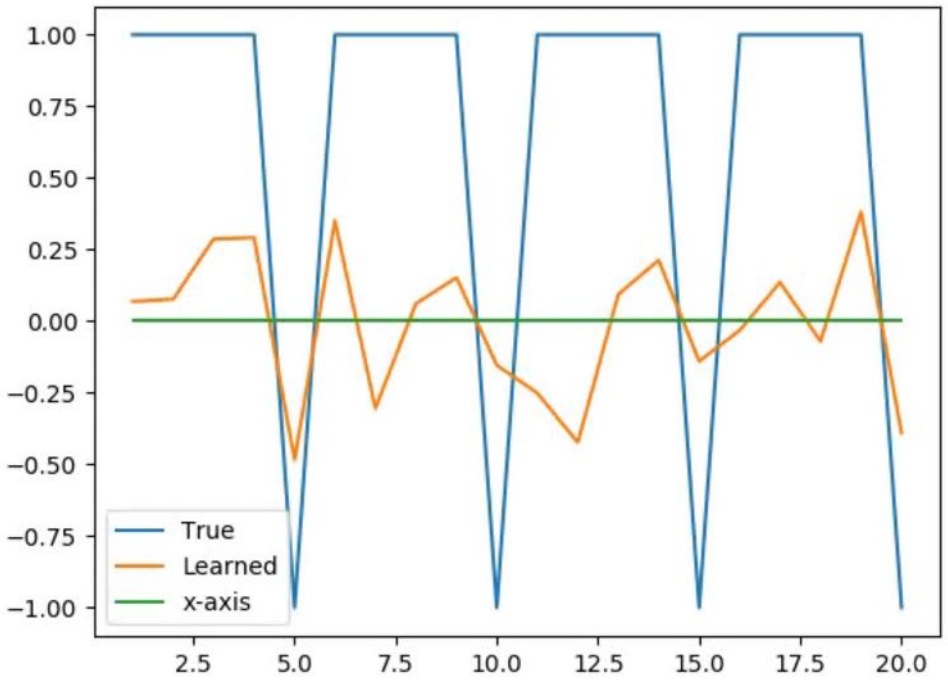
蛋少是开局吗？

参数：

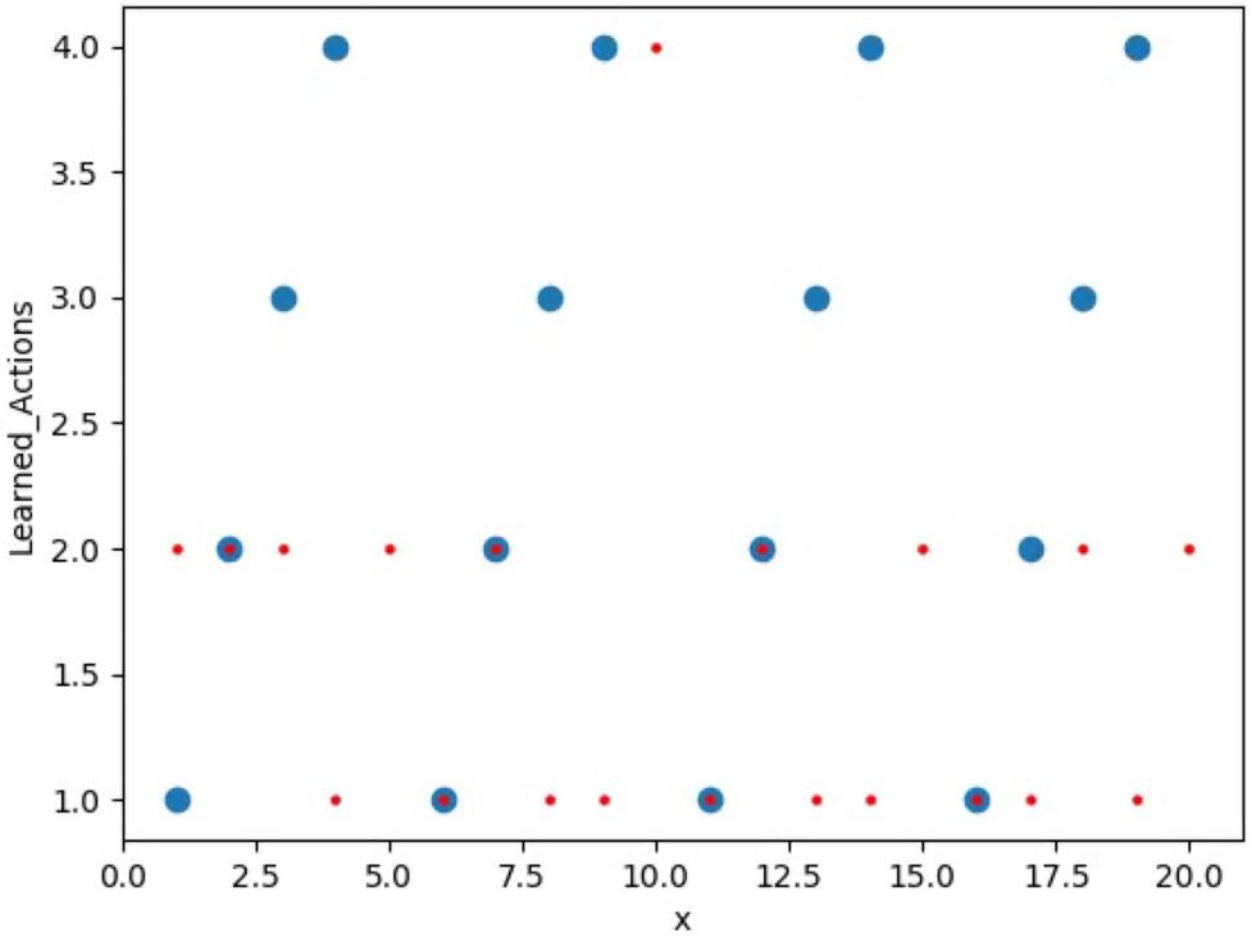
仿真20次
仿真深度2
三层双头神经网络：

剩余蛋数→（拿蛋策略softmax，胜负判断tanh）

pitybea AlphaEggZero
https://github.com/pitybea/alphaeggzero



神经网络
胜负判断


Learned_Actions
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
x

- 开始不会下棋，接近随机走

探索

- 后来不太靠谱的简单模拟，产生神经网络的不太好的标签（输赢判断、落子策略）

探索+知识活用

- 不断慢慢变好
  - 神经网络预测变好
  - 简单模拟变靠谱，产生标签更好

- 最后成为世界冠军

知识活用

AlphaGo的启示

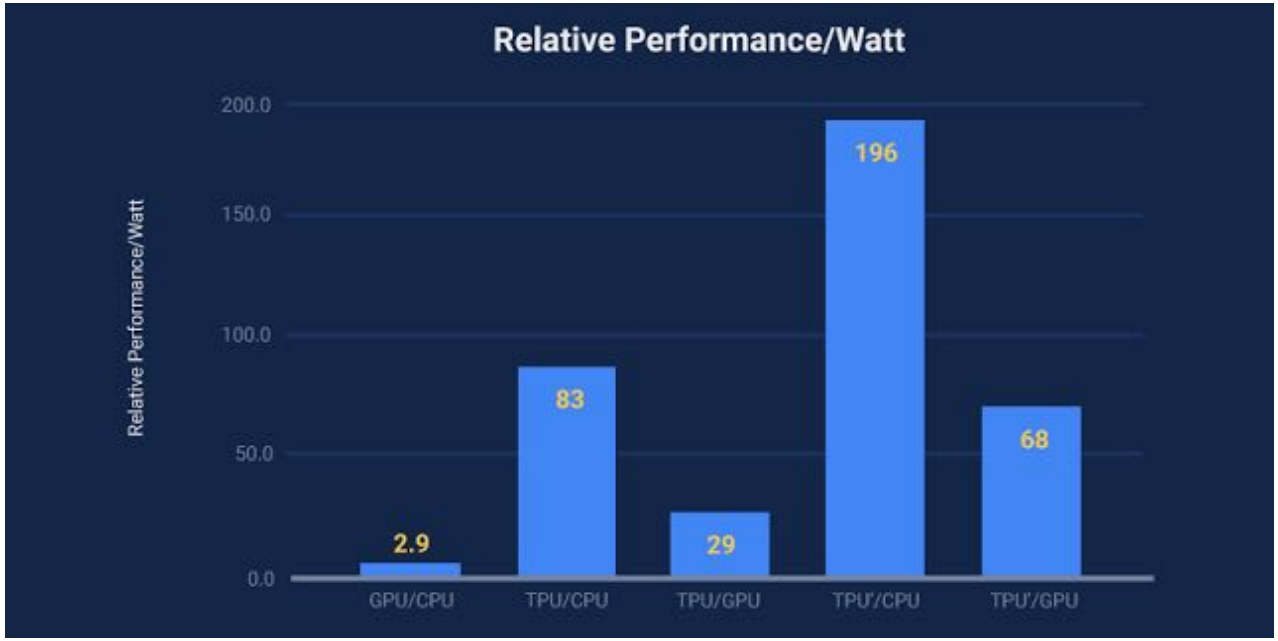
- 经常反思，通过结果与预期的不同，更新行动指引模型
- 活用知识，勇于探索
- 向前看，向前算，算到结局最好，算不到也有收获
- 经过迭代，不靠谱的也会变靠谱

AlphaGo各个版本

- AlphaGo Fan：2015年10月击败欧洲冠军樊麾
  - 1202个CPU和176GPU
- AlphaGo Lee：2016年3月，4：1击败世界冠军李世乭
  - 初始下棋策略从3000万局人类对弈中学习
- AlphaGo Master：2016年年底，网络战胜60名顶级高手
  - 放弃人类棋谱
- AlphaGo Zero：2017年5月战胜柯洁，三天从tabula rasa到世界冠军
  - 退役
- Alpha Zero：Go之后，在国际象棋、日本将棋等棋类中棋力达到世界顶级水准

历史

AlphaGo Zero训练过程一些技术指标



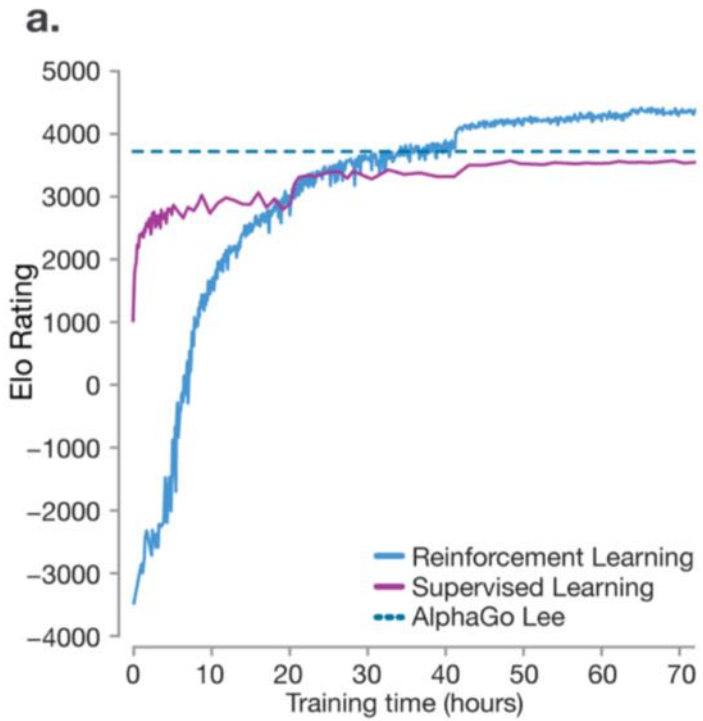
AlphaGo Zero训练过程一些技术指标

- 学习时使用5000个一代TPU
  - 约等于一百万台普通电脑
- 下棋时使用4个TPU
  - 1000台左右普通电脑

AlphaGo Zero训练过程一些技术指标

- 72小时自我对弈，产生500万棋局
- 使用14亿（状态，输赢判断，下棋策略）组合训练神经网络
  - 也就是每局面棋大约280手结束
- 每次MCTS，进行1600次 “仿真”

AlphaGo Zero训练过程一些技术指标
超级72小时



非完美信息博弈

- 德州扑克、两个人玩石头剪刀布
  - Counterfactual regret minimization

- 今天就不讲了，与MCTS部分相通

大状态空间完美信息博弈与强人工智能

状态搜索 + 函数拟合不能催生强人工智能吧。。。

谢谢大家

Reference:

AlphaGo系列论文

AlphaGo Zero Cheat Sheet

浅述：从MinMax到AlphaZero，完全（美）信息博弈之路 @ 知乎