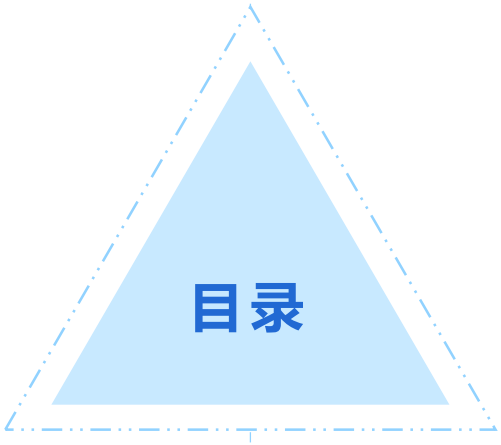
# AlphaGo的胜利

(理工类)

东南大学 计算机科学与工程学院

# 目录



# 围棋简介

# “琴棋书画”之棋

- 围棋起源于中国，称之为**“弈”**
- 经日本传入欧洲被译成**“Go”**

尧造围棋，丹朱善之

——先秦典籍《世本》

# 围棋基本介绍

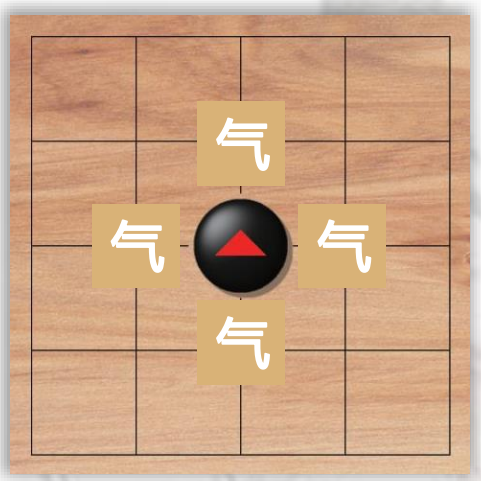
棋盘与棋子

行棋规则

胜负判定

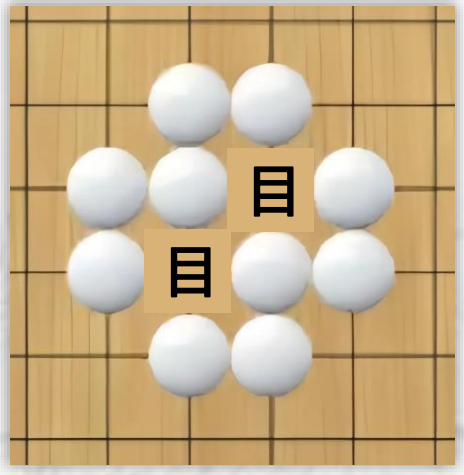
特殊规则

# 围棋基本介绍



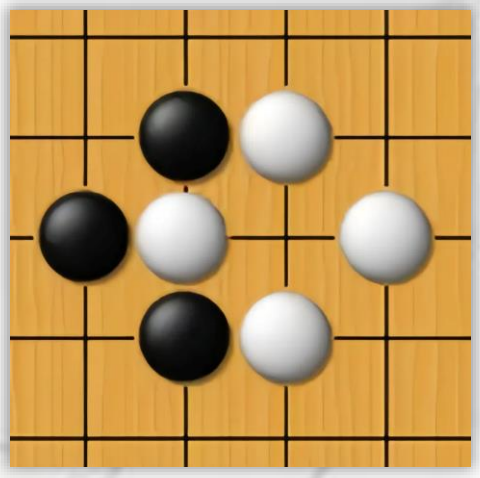

## 气

与棋子直线相邻空交叉点




## 目

被一方棋子所围地域的空交叉点



## 打劫

双方可以轮流提取对方棋子的情况

# 围棋棋手的养成

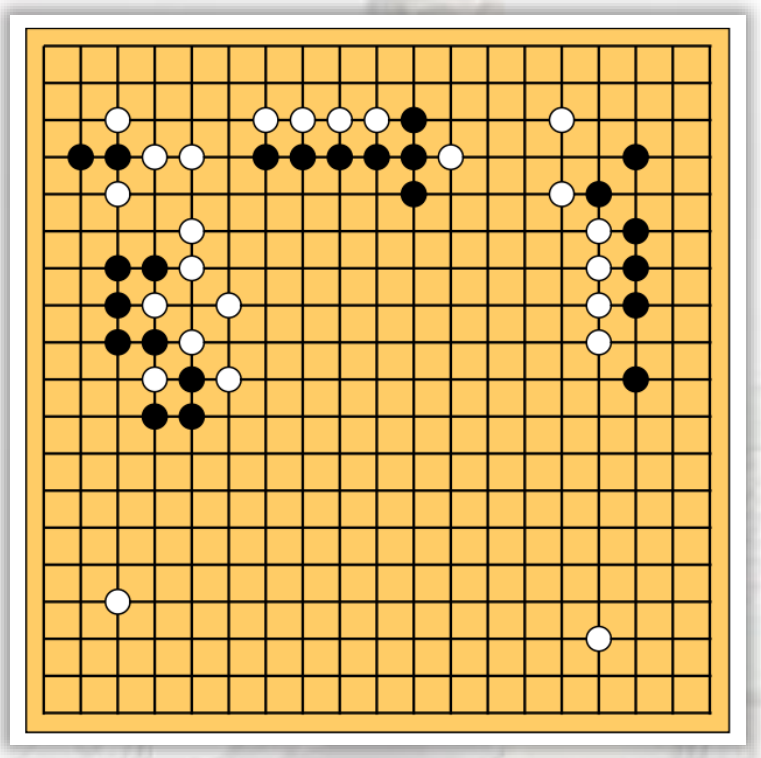
## 一名优秀的棋手的成长路径

- 基本的布局理论
- 死活、对杀常识
- 基本的布局理论
- 数万死活题
- 对局中磨练

→ 短期/局部战术

→ 长期/全局战略

# 围棋复杂性



- **棋盘规模**

  19x19 的棋盘拥有 361 个交　叉点

- **棋局数量**

  大约为 $10^{170}$，超越宇宙原子数

- **决策深度**

  通常每盘棋需要250步左右

# 围棋的特点

## 典型的完全信息博弈问题

**信息对称**

**零和**

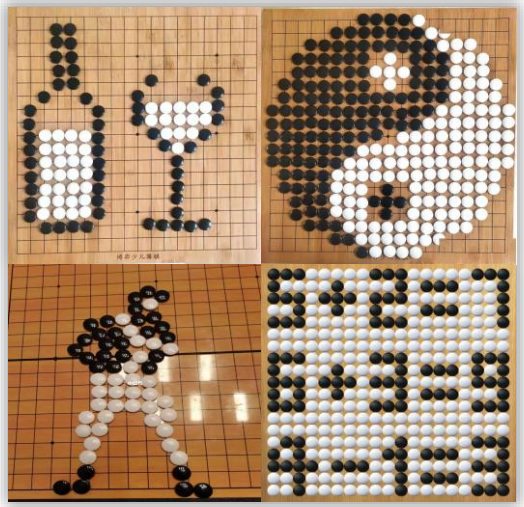
**纳什平衡**

任何一种完全信息博弈的棋类游戏，在当前状态下都存在一个最优策略以及与之对应的最优状态价值函数

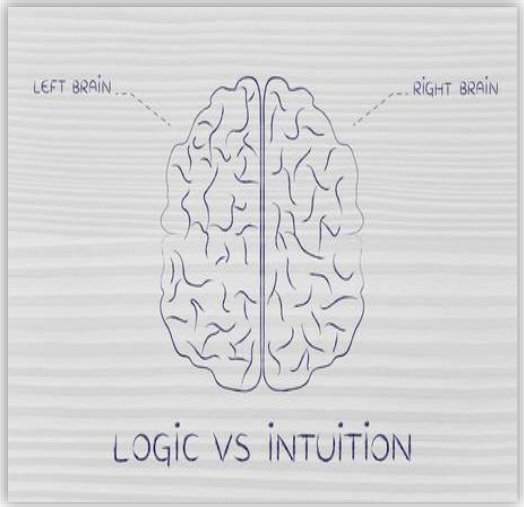
恩斯特·策梅洛

# 围棋与人工智能

## 围棋长期以来被认为是人工智能领域的“圣杯”



庞大的搜索空间



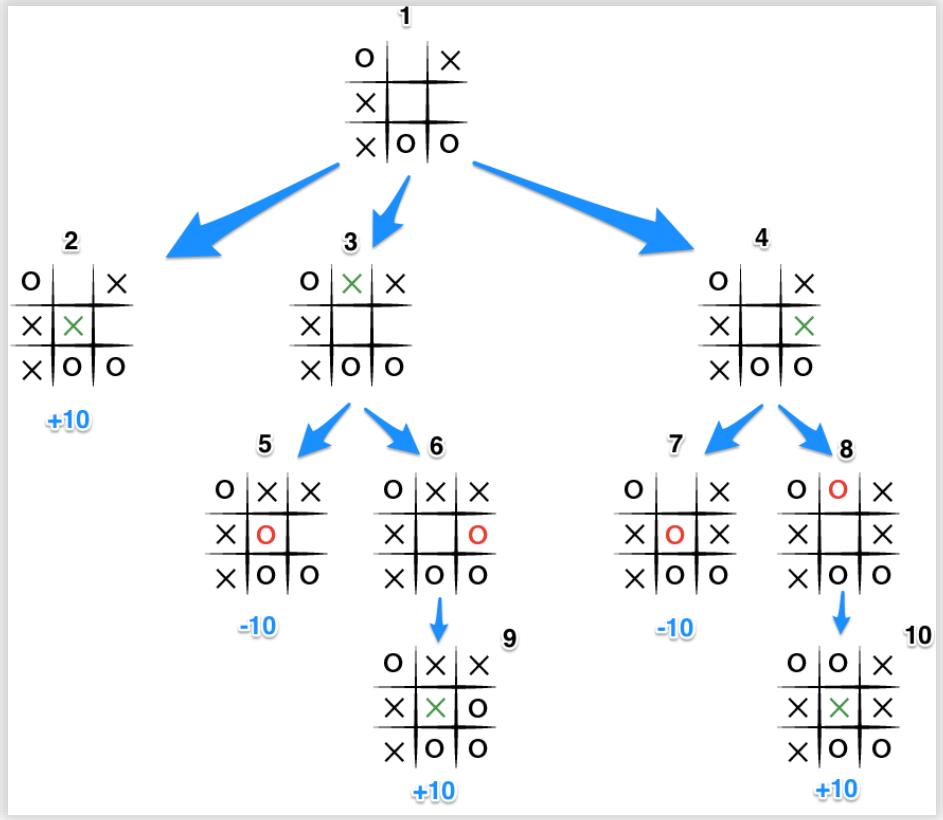

长远战略与直觉

$$\mathcal{L}_{\text{StandardModel}} =$$
$$-\frac{1}{2}\partial_\nu g^a_\mu \partial_\nu g^a_\mu - g_s f^{abc}\partial_\mu g^a_\nu g^b_\mu g^c_\nu - \frac{1}{4}g_s^2 f^{abc} f^{ade} g^b_\mu g^c_\nu g^d_\mu g^e_\nu +$$
$$\frac{1}{2}ig^2_s(\bar{q}^\sigma_i\gamma^\mu q^\sigma_j)g^a_\mu + \bar{G}^a\partial^2 G^a + g_s f^{abc}\partial_\mu \bar{G}^a G^b g^c_\mu - \partial_\nu W^+_\mu \partial_\nu W^-_\mu -$$
$$M^2 W^+_\mu W^-_\mu - \frac{1}{2}\partial_\nu Z^0_\mu \partial_\nu Z^0_\mu - \frac{1}{2c^2_w}M^2 Z^0_\mu Z^0_\mu - \frac{1}{2}\partial_\mu A_\nu \partial_\mu A_\nu - \frac{1}{2}\partial_\mu H \partial_\mu H -$$
$$\frac{1}{2}m^2_h H^2 - \partial_\mu \phi^+ \partial_\mu \phi^- - M^2\phi^+\phi^- - \frac{1}{2}\partial_\mu \phi^0 \partial_\mu \phi^0 - \frac{1}{2c^2_w}M\phi^0\phi^0 - \beta_h[\frac{2M^2}{g^2} +$$
$$\frac{2M}{g}H + \frac{1}{2}(H^2 + \phi^0\phi^0 + 2\phi^+\phi^-)] + \frac{2M^4}{g^2}\alpha_h - igc_w[\partial_\nu Z^0_\mu(W^+_\mu W^-_\nu -$$
$$W^+_\nu W^-_\mu) - Z^0_\nu(W^+_\mu \partial_\nu W^-_\mu - W^-_\mu \partial_\nu W^+_\mu) + Z^0_\mu(W^+_\nu \partial_\nu W^-_\mu -$$
$$W^-_\nu \partial_\nu W^+_\mu)] - igs_w[\partial_\nu A_\mu(W^+_\mu W^-_\nu - W^+_\nu W^-_\mu) - A_\nu(W^+_\mu \partial_\nu W^-_\mu -$$
$$W^-_\mu \partial_\nu W^+_\mu) + A_\mu(W^+_\nu \partial_\nu W^-_\mu - W^-_\mu \partial_\nu W^+_\mu)] - \frac{1}{2}g^2 W^+_\mu W^-_\mu W^+_\nu W^-_\nu +$$
$$\frac{1}{2}g^2 W^+_\mu W^-_\nu W^+_\mu W^-_\nu + g^2 c^2_w(Z^0_\mu W^+_\mu Z^0_\nu W^-_\nu - Z^0_\mu Z^0_\mu W^+_\nu W^-_\nu) +$$
$$g^2 s^2_w(A_\mu W^+_\mu A_\nu W^-_\nu - A_\mu A_\mu W^+_\nu W^-_\nu) + g^2 s_w c_w[A_\mu Z^0_\nu(W^+_\mu W^-_\nu -$$
$$W^+_\nu W^-_\mu) - 2A_\mu Z^0_\mu W^+_\nu W^-_\nu] - g\alpha[H^3 + H\phi^0\phi^0 + 2H\phi^+\phi^-] -$$
$$\frac{1}{8}g^2\alpha_h[H^4 + (\phi^0)^4 + 4(\phi^+\phi^-)^2 + 4(\phi^0)^2\phi^+\phi^- + 4H^2\phi^+\phi^- + 2(\phi^0)^2H^2] -$$
$$gMW^+_\mu W^-_\mu H - \frac{1}{2}g\frac{M}{c^2_w}Z^0_\mu Z^0_\mu H - \frac{1}{2}ig[W^+_\mu(\phi^0\partial_\mu\phi^- - \phi^-\partial_\mu\phi^0) -$$
$$W^-_\mu(\phi^0\partial_\mu\phi^+ - \phi^+\partial_\mu\phi^0)] + \frac{1}{2}g[W^+_\mu(H\partial_\mu\phi^- - \phi^-\partial_\mu H) - W^-_\mu(H\partial_\mu\phi^+ -$$
$$\phi^+\partial_\mu H)] + \frac{1}{2}g\frac{1}{c_w}(Z^0_\mu(H\partial_\mu\phi^0 - \phi^0\partial_\mu H) - ig\frac{s^2_w}{c_w}MZ^0_\mu(W^+_\mu\phi^- - W^-_\mu\phi^+) +$$
$$igs_w MA_\mu(W^+_\mu\phi^- - W^-_\mu\phi^+) - ig\frac{1-2c^2_w}{2c_w}Z^0_\mu(\phi^+\partial_\mu\phi^- - \phi^-\partial_\mu\phi^+) +$$
$$igs_w A_\mu(\phi^+\partial_\mu\phi^- - \phi^-\partial_\mu\phi^+) - \frac{1}{4}g^2W^+_\mu W^-_\mu[H^2 + (\phi^0)^2 + 2\phi^+\phi^-] -$$
$$\frac{1}{4}g^2\frac{1}{c^2_w}Z^0_\mu Z^0_\mu[H^2 + (\phi^0)^2 + 2(2s^2_w - 1)^2\phi^+\phi^-] - \frac{1}{2}g^2\frac{s^2_w}{c_w}Z^0_\mu\phi^0(W^+_\mu\phi^- +$$
$$W^-_\mu\phi^+) - \frac{1}{2}ig^2\frac{s^2_w}{c_w}Z^0_\mu H(W^+_\mu\phi^- - W^-_\mu\phi^+) + \frac{1}{2}g^2 s_w A_\mu\phi^0(W^+_\mu\phi^- +$$
$$W^-_\mu\phi^+) + \frac{1}{2}ig^2 s_w A_\mu H(W^+_\mu\phi^- - W^-_\mu\phi^+) - g^2\frac{s_w}{c_w}(2c^2_w - 1)Z^0_\mu A_\mu\phi^+\phi^- -$$
$$g^1 s^2_w A_\mu A_\mu\phi^+\phi^- - \bar{e}^\lambda(\gamma\partial + m^\lambda_e)e^\lambda - \bar{\nu}^\lambda\gamma\partial\nu^\lambda - \bar{u}^\lambda_j(\gamma\partial + m^\lambda_u)u^\lambda_j -$$
$$\bar{d}^\lambda_j(\gamma\partial + m^\lambda_d)d^\lambda_j + igs_w A_\mu[-(\bar{e}^\lambda\gamma^\mu e^\lambda) + \frac{2}{3}(\bar{u}^\lambda_j\gamma^\mu u^\lambda_j) - \frac{1}{3}(\bar{d}^\lambda_j\gamma^\mu d^\lambda_j)] +$$
$$\frac{ig}{4c_w}Z^0_\mu[(\bar{\nu}^\lambda\gamma^\mu(1+\gamma^5)\nu^\lambda) + (\bar{e}^\lambda\gamma^\mu(1 - 4s^2_w - 1\gamma^5)e^\lambda) + (\bar{u}^\lambda_j\gamma^\mu(\frac{8}{3}s^2_w -$$
$$1 - \gamma^5)u^\lambda_j) + (\bar{d}^\lambda_j\gamma^\mu(1 - \frac{8}{3}s^2_w - \gamma^5)d^\lambda_j)] + \frac{ig}{2\sqrt{2}}W^+_\mu[(\bar{\nu}^\lambda\gamma^\mu(1+\gamma^5)e^\lambda) +$$
$$(\bar{u}^\lambda_j\gamma^\mu(1+\gamma^5)C_{\lambda\kappa}d^\kappa_j)] + \frac{ig}{2\sqrt{2}}W^-_\mu[(\bar{e}^\lambda\gamma^\mu(1+\gamma^5)\nu^\lambda) + (\bar{d}^\kappa_j C^\dagger_{\lambda\kappa}\gamma^\mu(1+$$
$$\gamma^5)u^\lambda_j)] + \frac{ig}{2\sqrt{2}}\frac{m^\lambda_e}{M}[-\phi^+(\bar{\nu}^\lambda(1-\gamma^5)e^\lambda) + \phi^-(\bar{e}^\lambda(1+\gamma^5)\nu^\lambda)] -$$

评估函数的复杂性

# 传统方法

# 博弈树搜索



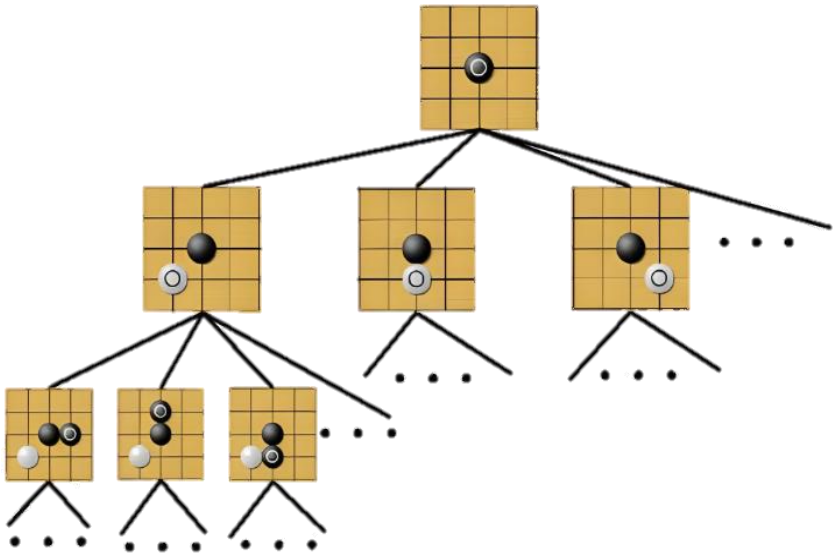

**模拟**所有可能未来局面

**评估**每个未来局面好坏

帮助玩家找到**最优策略**

# Minimax（或MinMax）算法

假设双方都是**理性玩家**，并且都试图做出**最优决策**



我方棋步，选择对我最有利（max）

↓

对方棋步，选择对我最有害（min）

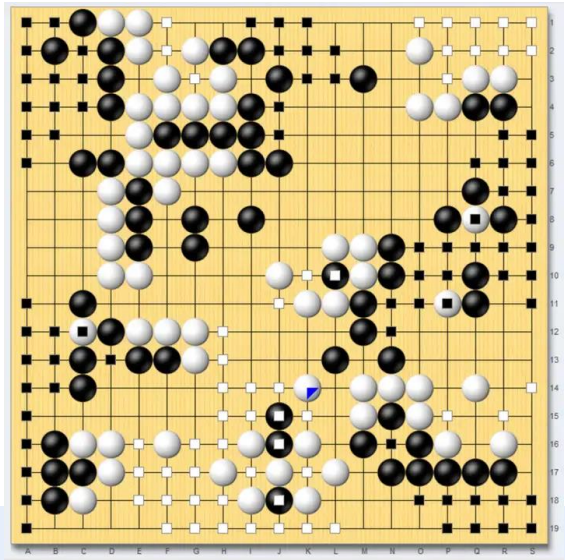
↓

我方棋步，选择对我最有利（max）

↓

. . .

# 评估局面



地盘控制

气数

目数

边角控制

其他评估

人工设计评估函数 → 局限性

- 局部与全局难平衡
- 死活等问题的复杂性
- 状态空间庞大

# 博弈树搜索的局限

## 1.计算资源的限制

不能构建整个决策树，较浅的搜索深度可能导致不能发现最佳策略

## 2.规则和局面特征多样

难以构建精确的评估函数，整个搜索过程的价值将大打折扣


第一代围棋AI“手谈”
达到约13层的搜索深度

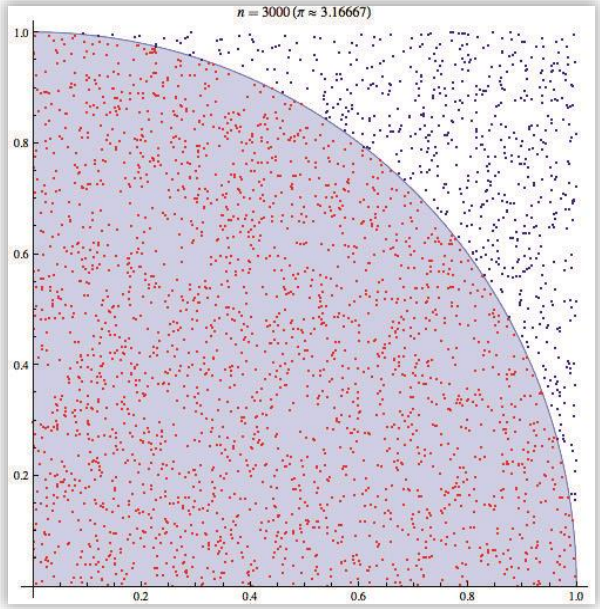
# 蒙特卡洛树搜索（Monte Carlo Tree Search）

## 蒙特卡洛树搜索算法开启第二代围棋人工智能

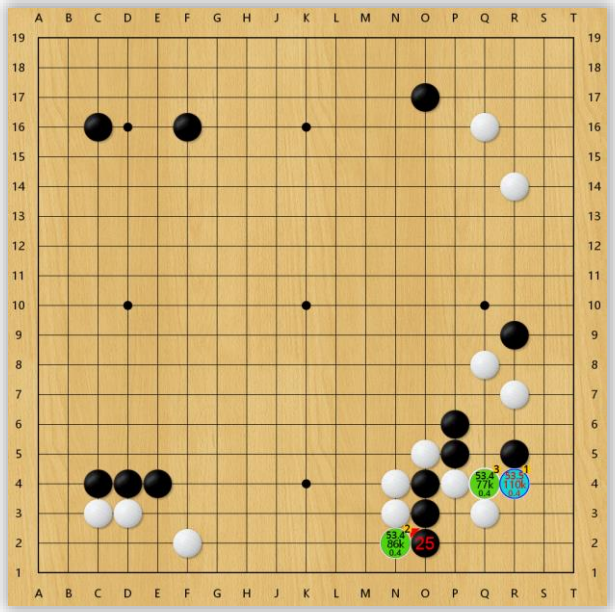
2015年，职业棋手连笑七段在让4子和让5子两场人机对战中，战胜计算机围棋冠军韩国软件 “石子旋风” （DolBaram），但在让6子比赛中败下阵来。

# 蒙特卡洛方法

## 一种基于“随机数”的计算方法。




空间内多次落点来近似计算圆周率

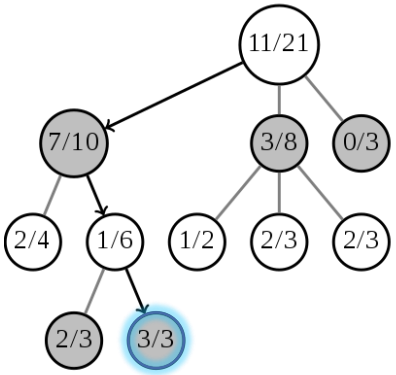


多次模拟棋局后看哪个选点的
“获胜概率”最高

# 蒙特卡洛树搜索（MCTS）

## 蒙特卡洛方法和搜索树结构的结合

### 选择（Selection）

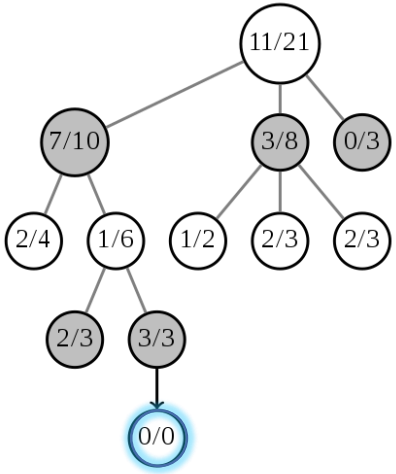



选择一个“<u>最值得看的子节点</u>”

1.已被证明胜率高

2.尚未被充分探索

### 扩展（Expansion）

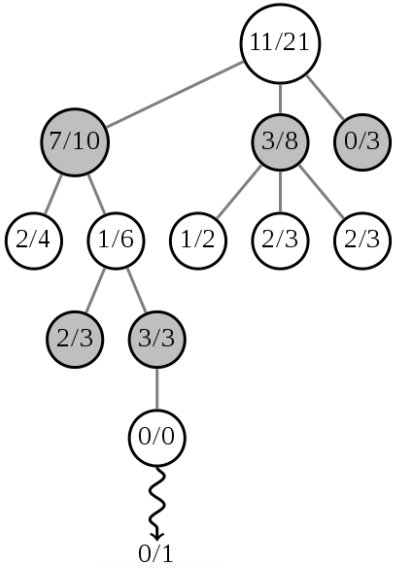



在被选择节点添加一个子节点来扩展搜索树

# 蒙特卡洛树搜索（MCTS）

## 蒙特卡洛方法和搜索树结构的结合

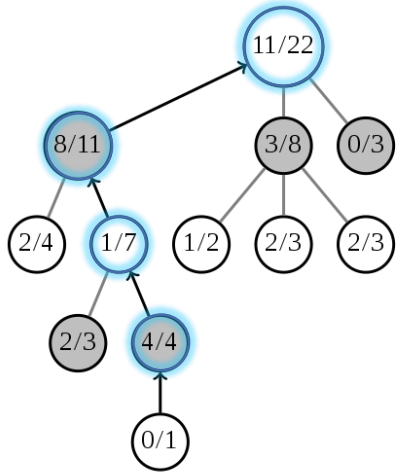
### 模拟（Simulation）




输/赢

随机落子模拟到游戏结束，评估最终结果

### 回溯（Backpropagation）




模拟结果沿着路径更新节点的统计信息

1.访问次数
2.节点胜率

# 蒙特卡洛树搜索的优势和局限

## 优势

适用于庞大状态空间

逐步收敛到最优解

避免复杂的评估函数

## 局限

模拟阶段的随机走法过于粗糙

效率取决于模拟的次数


第二代AI围棋程序

棋手战鹰
职业二段

# 传统方法和人类的差别

## 棋感、大局观，非“机器”所能掌握

**蒙特卡洛树搜索法**

模拟统计结果驱动

局部累积形成全局

没有积累缺乏创新

局面感知能力较弱

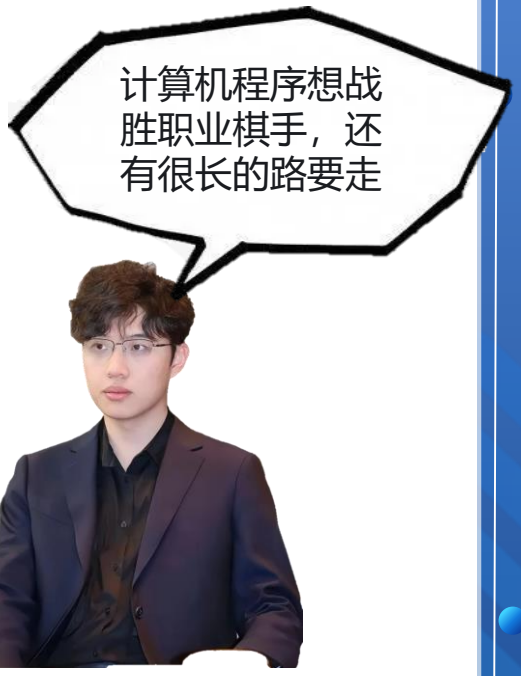
**职业棋手**

直觉与逻辑驱动

局部与全局平衡

过往与当前棋局

了解定式和棋形

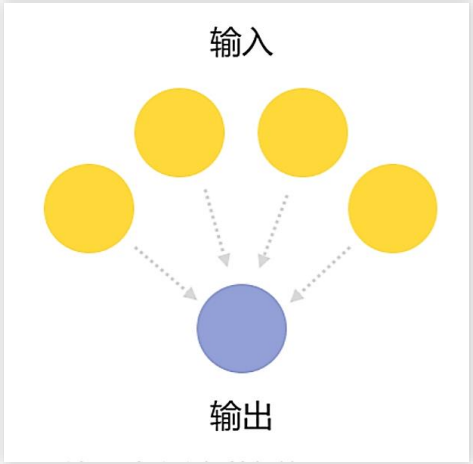



**棋手连笑**
**职业七段**

# AlphaGo

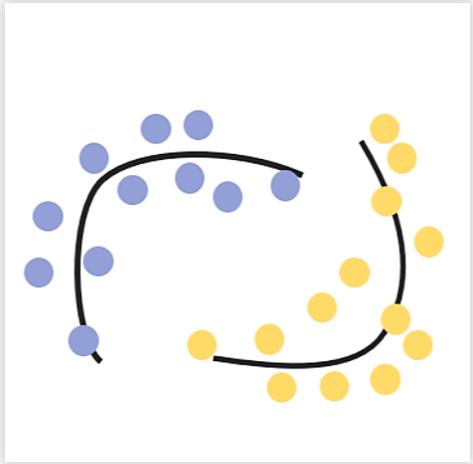
# 机器学习

## 监督学习
## Supervised Learning



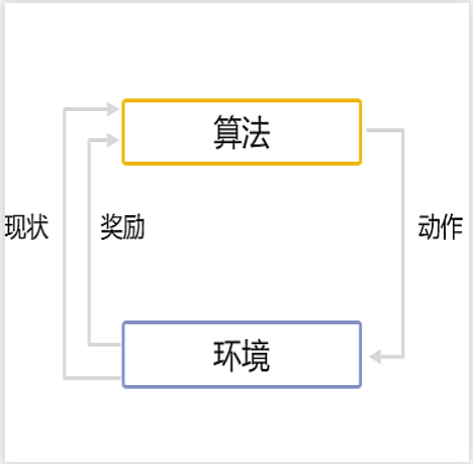

有预测目标Y，通过输入X预测

## 无监督学习
## Unsupervised Learning



只通过输入X进行分析
并识别模式

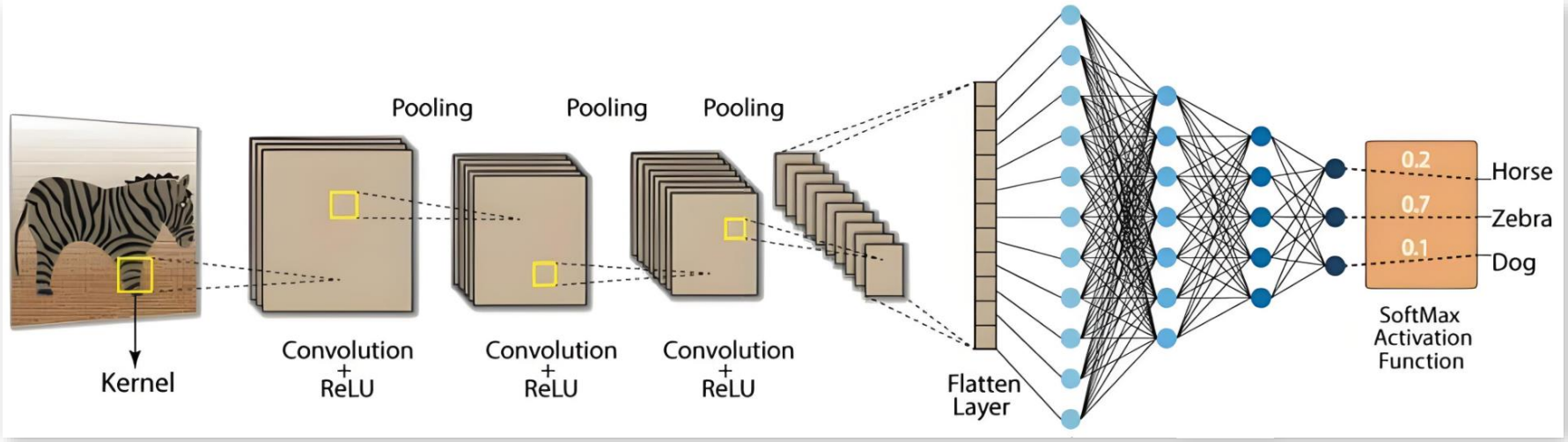
## 强化学习
## Reinforcement Learning




通过环境与奖励循环迭代优化出
最合适的动作

# 深度卷积神经网络

| 输入层 | 隐含层 | 输出层 |
| --- | --- | --- |
| 接收多维数组 | 卷积层、汇合层和全连接层 | 输出分类标签 |



# AlphaGo

## 结合蒙特卡洛树搜索（MCTS）和深度学习的围棋人工智能

### 策略网络

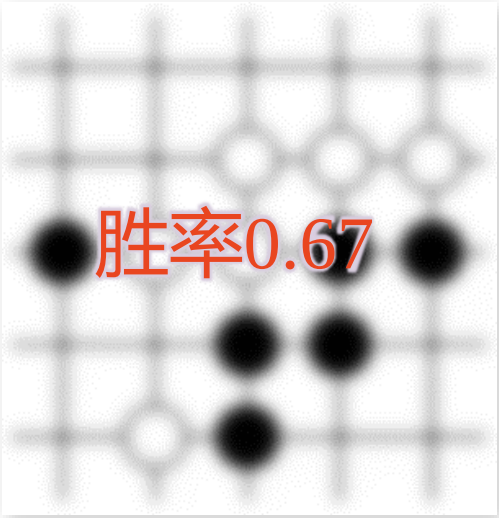
预测下一步最优的棋步

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0.1 | 0 | 0 |
| 0 | 0.1 | - | - | - |
| - | - | - | - | - |
| 0 | 0.5 | - | - | 0 |
| 0 | 0.1 | - | 0.2 | 0 |

每一个可能的落子点赋予一个概率值，快速筛选出一组有潜力的候选动作

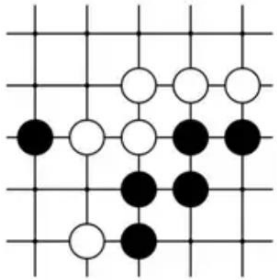
### 价值网络

评估当前局面的胜负概率




评估整个局面对于当前执棋方的好坏，替代部分模拟对局

# 神经网络的输入



白棋

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |

黑棋

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |

可落子位置

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |

其他已知先验知识，围棋中的气，目，打劫等的特征。

气

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 5 | 5 | 5 |
| 2 | 5 | 5 | 3 | 3 |
| 0 | 0 | 3 | 3 | 0 |
| 0 | 2 | 3 | 0 | 0 |

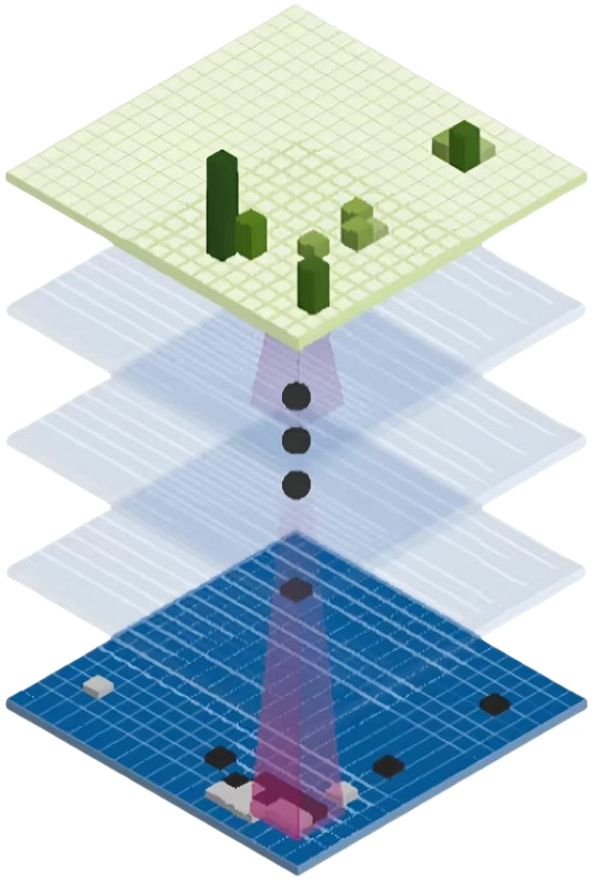
…

输入维度

$19 \times 19 \times 48$

# 策略网络P(Policy Networks)

## 预测下一步棋的可能落子位置



| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0.1 | 0 | 0 |
| 0 | 0.1 | - | - | - |
| - | - | - | - | - |
| 0 | 0.5 | - | - | 0 |
| 0 | 0.1 | - | 0.2 | 0 |

每一个合法动作的概率分布$P(a|s)$

Softmax输出层

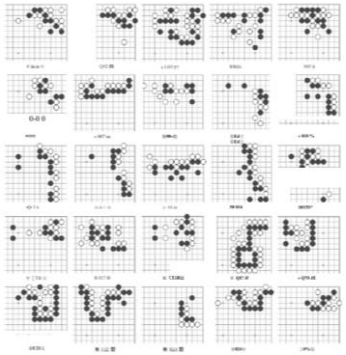
非线性函数

13层卷积神经网络

输入

# 监督学习训练策略网络 $P_\sigma$

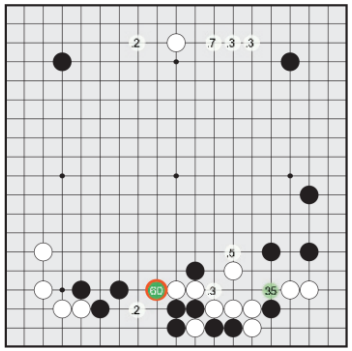
## 模仿人类专家下棋



160,000局6至9段人类选手的围棋对弈
**30,000,000**个构建的 $< s, a >$

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | - | - | - |
| - | - | - | - | - |
| 0 | 1 | - | - | 0 |
| 0 | 0 | - | 0 | 0 |

优化目标



策略网络输出概率分布 $P_\sigma(a/s)$

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0.1 | 0 | 0 |
| 0 | 0.1 | - | - | - |
| - | - | - | - | - |
| 0 | 0.5 | - | - | 0 |
| 0 | 0.1 | - | 0.2 | 0 |

# 快速走子网络$P_\pi$(Rollout Policy Networks)

**用于快速模拟棋局发展，代替原本蒙特卡洛树中模拟阶段的随机落子**

**轻量化网络设计：**

输入仅包含当前黑白棋子的分布，

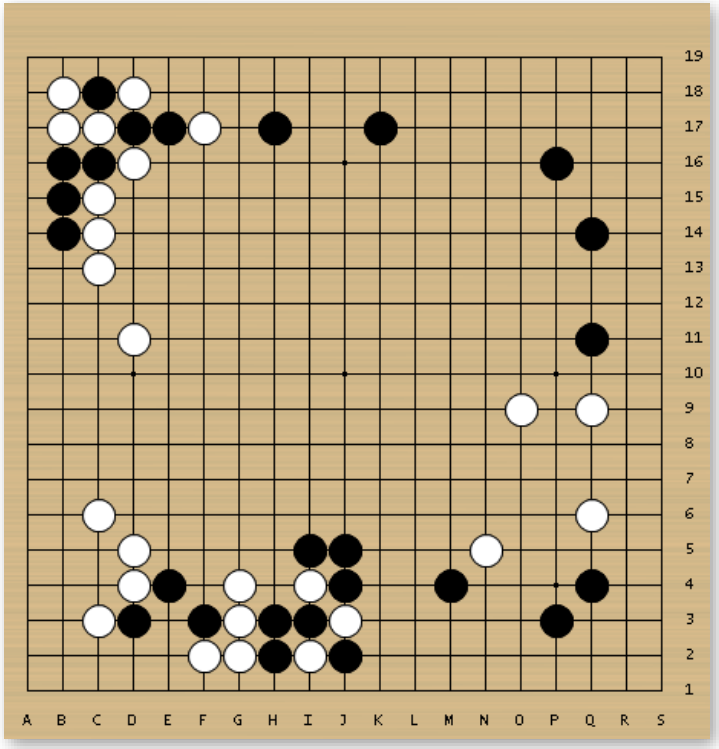
网络由较少层数的神经网络构成

**运算时间：**

$P_\pi$ 2μs> $P_\sigma$ 3ms

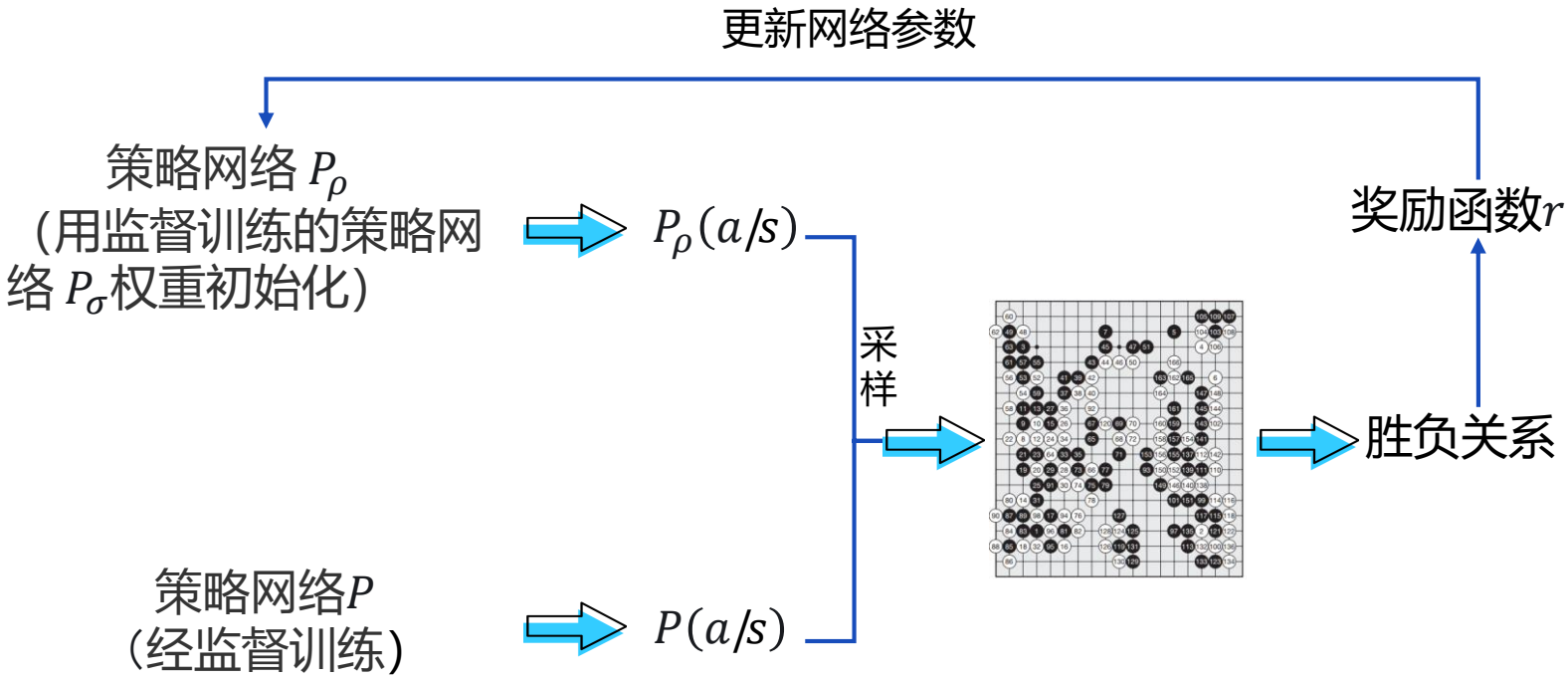
落子准确率：

$P_\sigma$ 55.7%> $P_\pi$ 24.2% > 随机选点



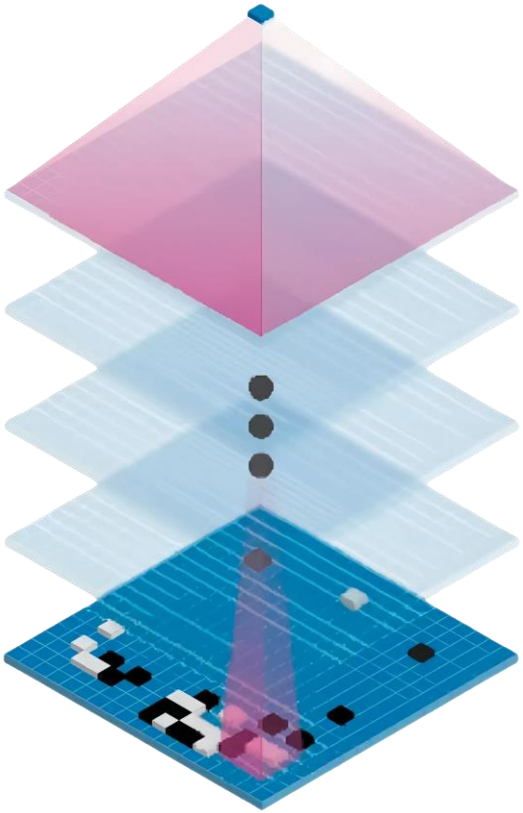
# 强化学习训练策略网络 $P_\rho$

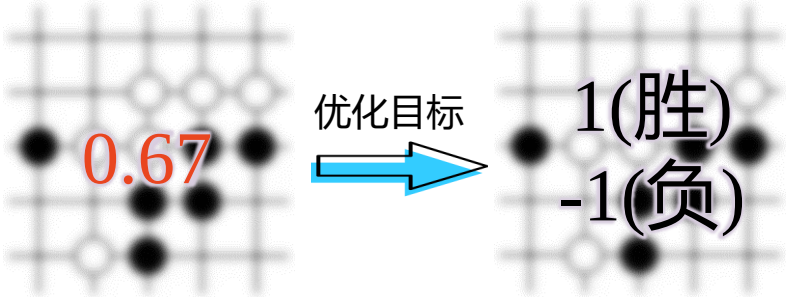
**策略网络的“左右互搏”，根据自我对弈的结果来优化决策**



# 价值网络$V_\theta$

神经网络学习一个函数 $v(s)$，直接输出一个该局面赢得比赛的概率



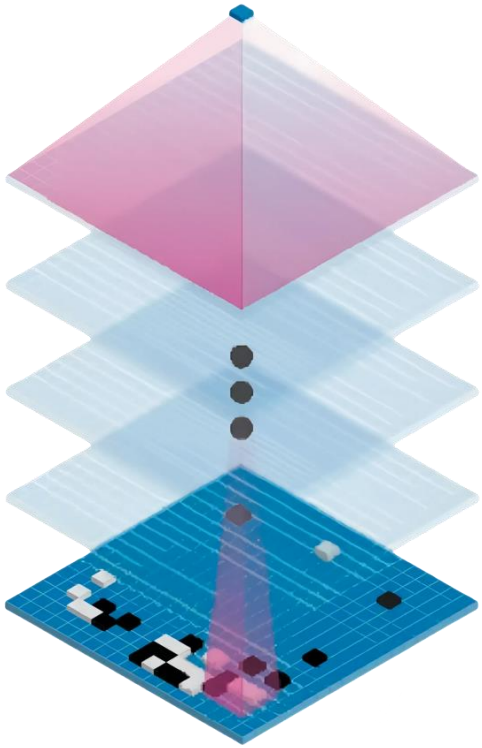



价值网络输出$v_\theta(s)$　　真实的胜负关系$v^*(s)$

传统方法
- 手工设计复杂的评估函数
- 模拟直到游戏结束再评估

# 强化学习训练价值网络$V_\theta$

**直接从人类完整棋局中学习价值网络会导致过拟合**



棋面胜负预测$v_\theta(s) \approx$ 本局真实胜负关系$v^*(g)$

优化目标

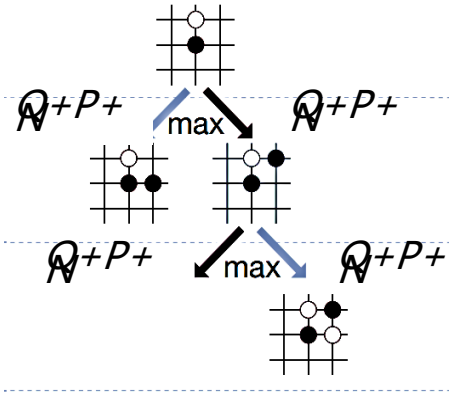
某个棋面下最终赢棋预测值$v_\theta(s)$

抽样出3000万个棋面$s$

策略网络 $P_\rho$进行自我对弈生成3000万个棋局$g$

# AlphaGo中的蒙特卡罗树搜索

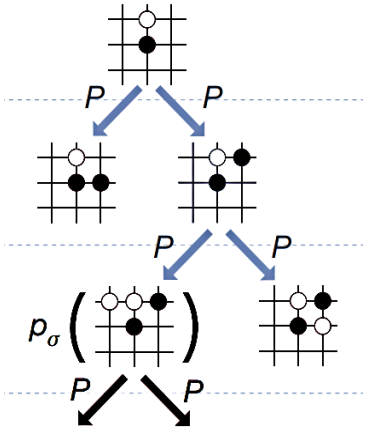
**a 选择分支**




节点胜率$Q(s,a)$
策略网络$P(s,a)$
访问次数$N(s,a)$

选择棋面$s$下的最佳动作$a$

**b 扩展分支**

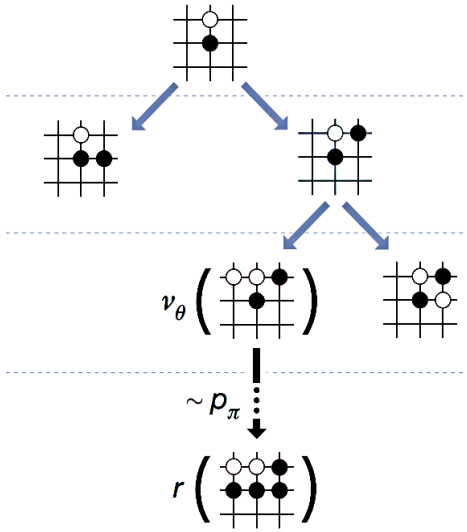



$$P(s,a) = P_\sigma(s,a)$$

保存子节点棋面策略网络结果作为先验概率

# AlphaGo中的蒙特卡罗树搜索

**c 结果判断**

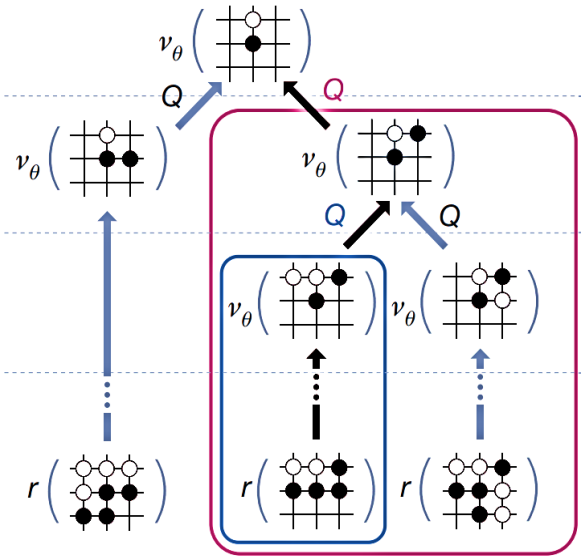



价值网络输出结果$v_\theta$

快速走子网络$P_\pi$模拟

胜负评估$r$

**d 反向传播**




反向更新所经过的边的
节点胜率$Q$和访问次数$N$

**最终决策：「选择访问次数最多的落子节点」**

# AlphaGo计算需求

| 传统围棋程序 | AlphaGo最终版本 | AlphaGo分布式版本 |
| --- | --- | --- |
|  |  |  |
| 1 个 线程 | 40 个 线程 | 40 个 线程 |
| 1 个 CPU | 48 个 CPU | 1202 个 CPU |
| 1 个 GPU | 8 个 GPU | 176 个 GPU |

# 终局之战

**AlphaGo**

分布式架构
1202个CPU与176个GPU训练了大约1周


2015 年， AlphaGo Fan 5 : 0 樊辉二段

**AlphaGo Lee**

分布式架构
采用48个TPU训练数月


2016，AlphaGo Lee 4 : 1 世界冠军李世石九段

# 终局之战

2017 年

AlphaGo Master

单机版

4 TPU

AlphaGo自我对弈提高棋力

拥有更强大的策略/价值网络

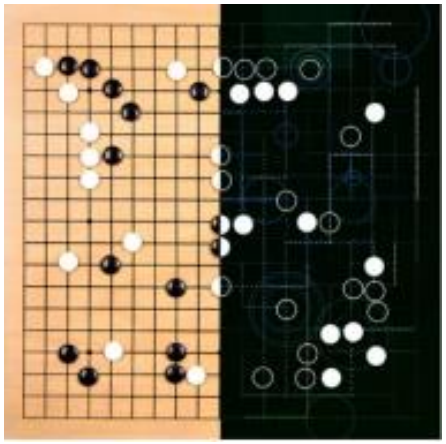
运算量只有AlphaGo Lee的十分之一

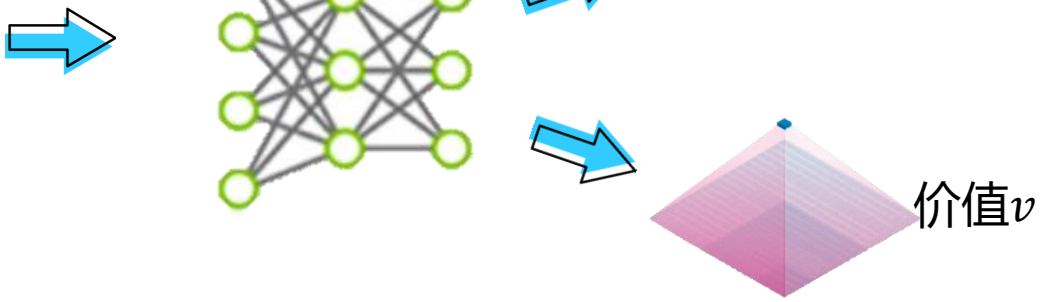
3 : 0


柯洁九段
世界排名第一
“人类最强棋手”

# AlphaGo Zero

## 从“人类专家知识”转向“纯粹的机器学习”



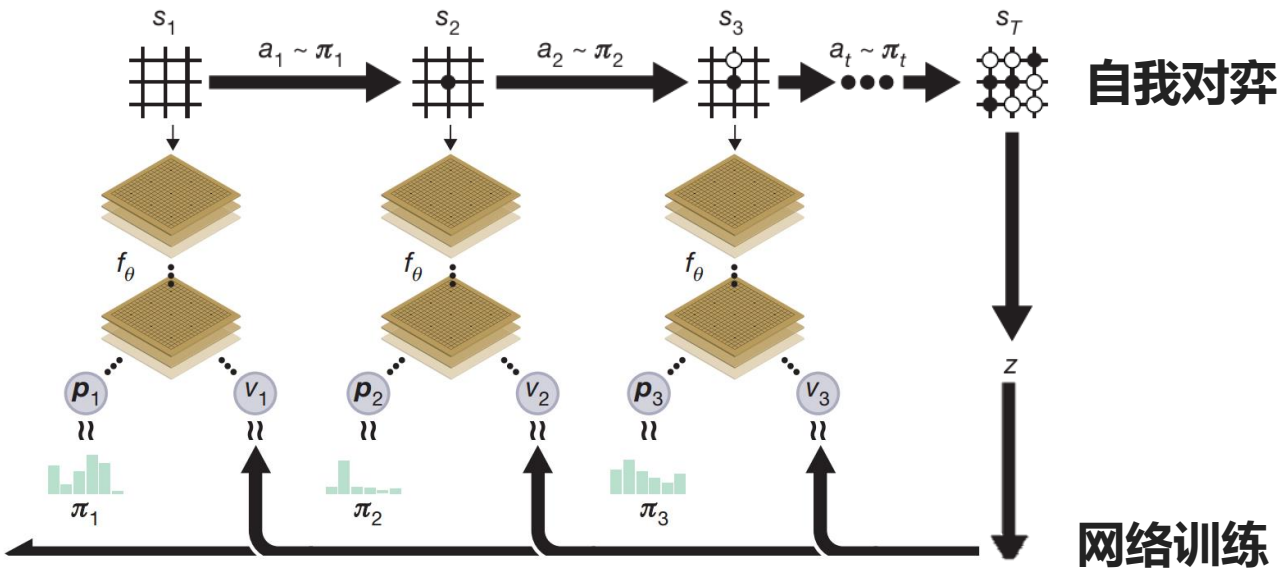



**无领域知识输入$s$**

**单一神经网络$f_\theta$**

# AlphaGo Zero——自我对弈

## 从“人类专家知识”转向“纯粹的机器学习”




**优化目标：** 1. 预测落子 ≈ 实际落子

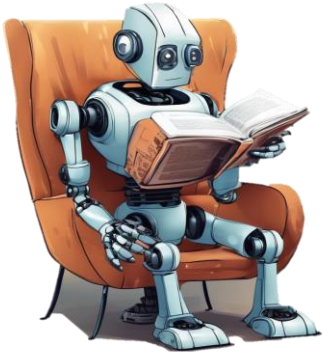
2. 预测获胜概率 ≈ 实际对局结果

# AlphaGo Zero的里程碑意义

AlphaGo Zero vs. AlphaGo Lee 100 : 0

AlphaGo Zero vs. AlphaGo Master 89 : 11


证明AI的潜力

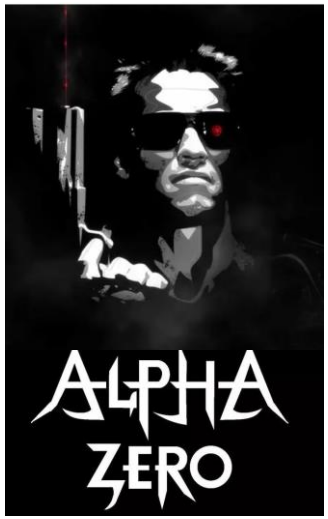


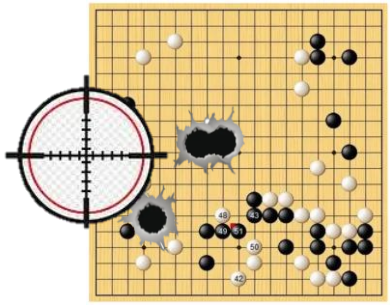
AI可以无需人类知识

# 后AlphaGo时代

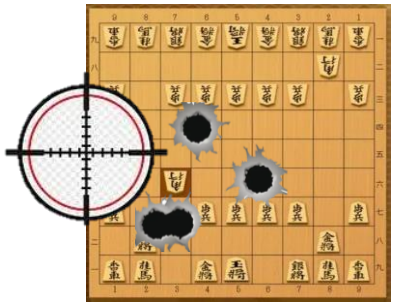
# AlphaZero

## 棋类游戏的“终结者”,迈向棋类通用人工智能系统



围棋



将棋

国际
象棋

# AlphaFold系列

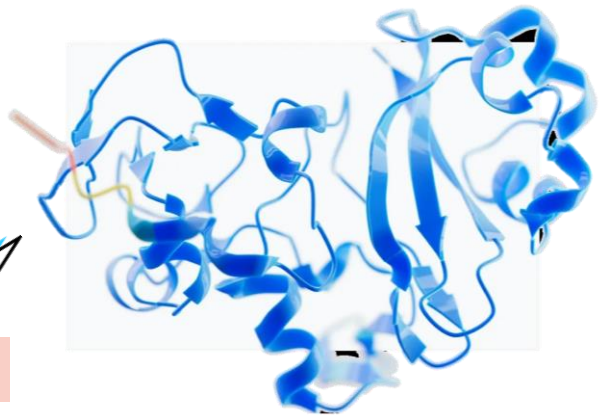
## Alpha系列首次应用于科学领域的重大突破

多个氨基酸单元组成
的氨基酸序列

结构预测模块

空间结构
- 氨基酸之间的**夹角分布**
- 氨基酸之间的**距离分布**

结构生成模块



**蛋白质三维结构预测**

# AlphaStar

## 应用于即时战略游戏《星际争霸II》

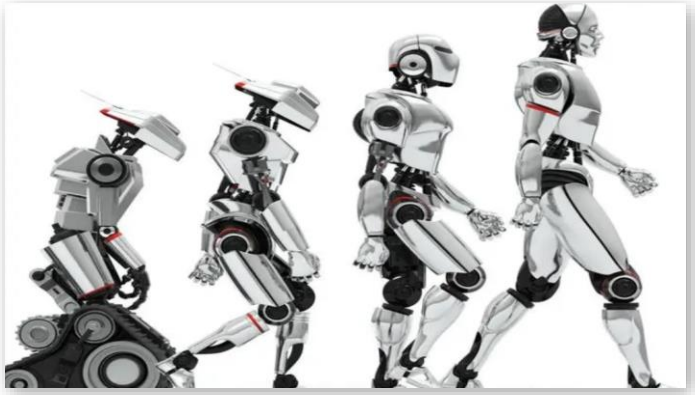

- 实时决策与不完全信息
- 深度强化学习与模仿学习结合
- 多智能体学习

# AI新时代的百花齐放


**ChatGPT**

**Stable Diffusion**

**Sora**



**具身智能**