

1

博弈 (games)

2

博弈中的优化决策 (Optimal Solution)

3

α - β 剪枝 (Alpha-Beta Pruning)

4

其他改进 (Improvement)

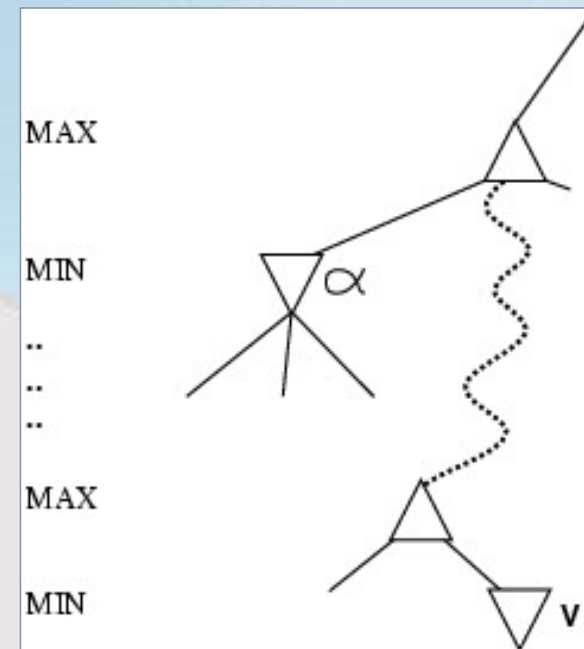
5

博弈的发展情况 (State-of-the-Art)

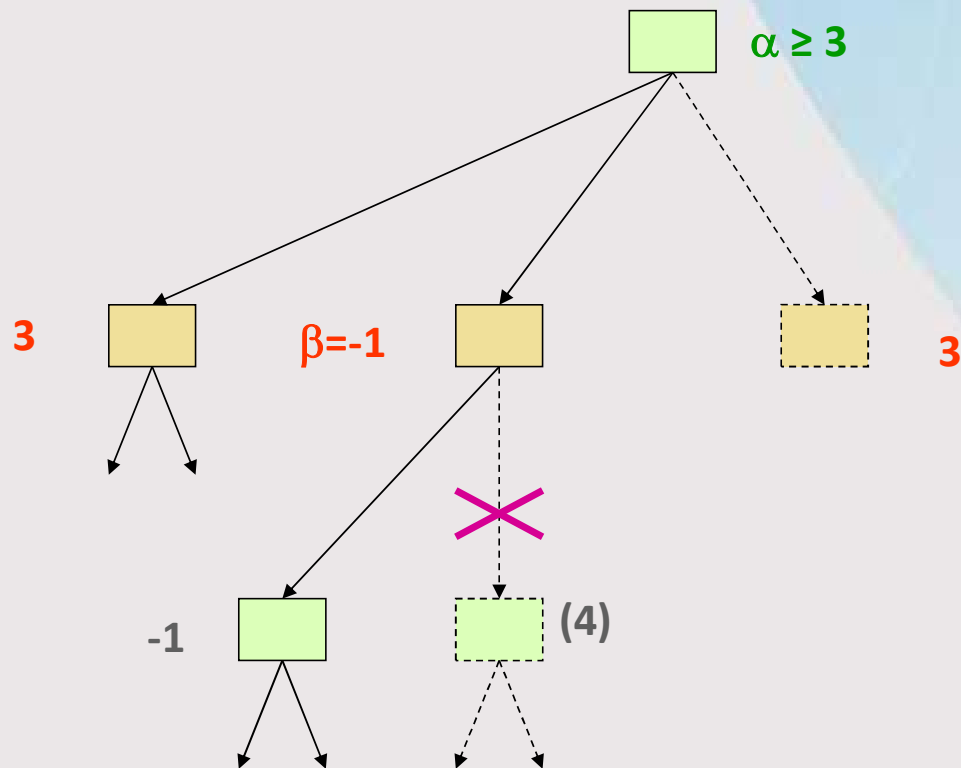


$\alpha - \beta$ 是什么?

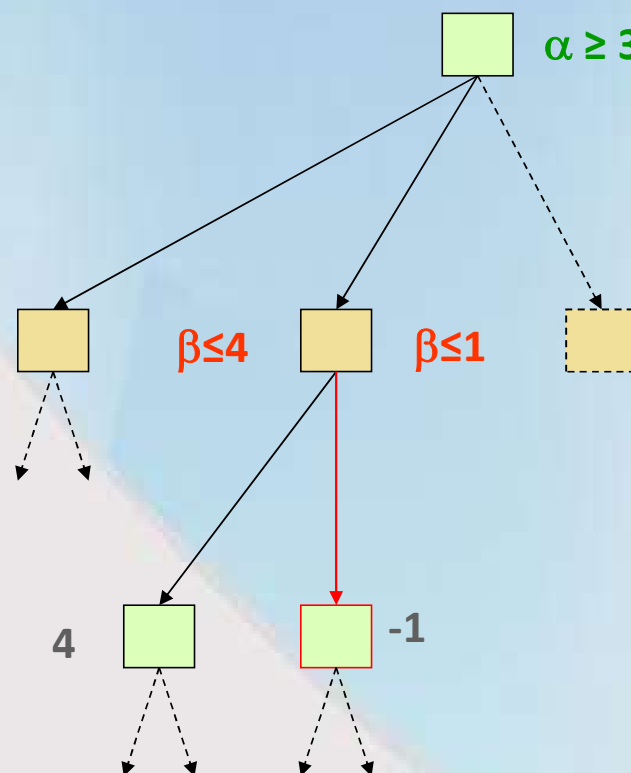
- α 是 **MAX** 至今为止的路径上所有选择点中发现的最好选择的值，即是最大值。
如果 v 比 α 差，**MAX** 会避免它，即发生剪枝。
- 类似的， β 是给 **MIN** 记录的最好结果即是最小值。
 - 如果 v 比 β 差，**MIN** 会避免它，即发生剪枝。



考虑下列二种情况：

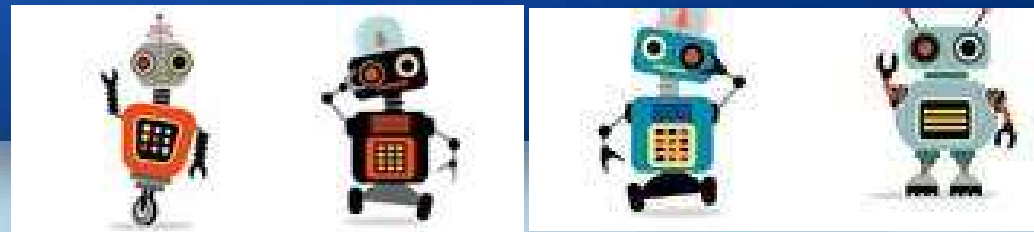


如何做的更好？



如何做的更好？

- 假设博弈树的分支因子为 b
则极大极小算法检测 $O(b^h)$ 个节点，最坏情况下 α - β 剪枝也是一样的。
 - 以下情况下 α - β 剪枝比极大极小算法优异：
 - a. 一个MAX节点的MIN孩子们是按降序排列的。
 - b. 一个MIN节点的MAX孩子们是按升序排列的。
 - 这种情况下 α - β 剪枝算法检测 $O(b^{h/2})$ 个节点。[Knuth and Moore, 1975]
 - 但这需要一个神谕（如果我们知道节点的完美排序，我们就不需要搜索博弈树了）。
- 如果节点是按随机排序的，那么 α - β 剪枝算法检测 $O(b^{3h/4})$ 个节点。
- 启发式极小极大值：节点的启发式排序
根以下的节点排序依照前一次循环所得到的倒推值进行。



其他改进方法

- 采用适应深度限制+循环加深
- 扩大搜索范围：保留 $k > 1$ 条的路径，代替仅保留一条，并且在大于设置深度的叶子节点下扩展博弈树（帮着对付水平线效应——指当前的后继都是差不多的状态）。
- 特殊情况扩展：在设置深度 h 时，如果一个节点明显地比其他的节点好，则沿着这个移动继续扩展几步。
- 用对照表法对付重复状态。
- 另外还有诸如当 $a \geq b$ 虽然不成立，但 a 不比 b 小多少时，仍然采用剪枝，特别是在开局初期。

其他

- 多玩家游戏，联盟或不联盟
- 随机游戏（如掷骰子），采用预期极大极小算法
- 不完全可观察状态空间，采用搜索信任空间的方法



1

博弈 (games)

2

博弈中的优化决策 (Optimal Solution)

3

α - β 剪枝 (Alpha-Beta Pruning)

4

其他改进 (Improvement)

5

博弈的发展情况 (State-of-the-Art)



Game Playing Algorithms Today 博弈算法的进展

Computers are better than humans 计算机优于人类

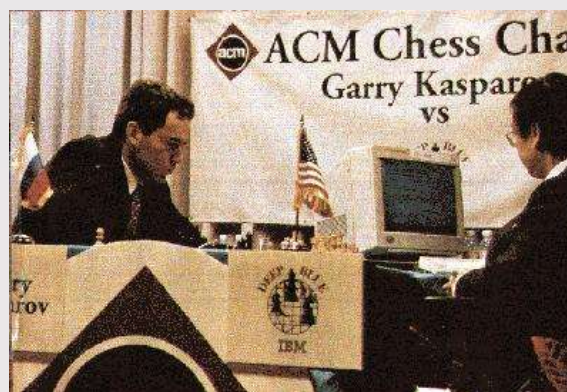
| | |
|------------------|---|
| Checkers 西洋跳棋 | Solved in 2007 2007年已解决 |
| Chess 国际象棋 | IBM Deep Blue defeated Kasparov in 1997 IBM深蓝于1997年战胜了卡斯帕罗夫 |
| Go 围棋 | Google AlphaGo beat Lee Sedol, a 9 dan professional in Mar. 2016 谷歌AlphaGo于2016年3月战胜了9段职业棋手李世乭 |

Computers are competitive with top human players 计算机与顶级人类玩家媲美

| | |
|---------------------|---|
| Backgammon 西洋双陆棋 | TD-Gammon used reinforcement learning to learn evaluation function TD-Gammon使用了强化学习方法来得到评价函数 |
| Bridge 桥牌 | Top systems use Monte-Carlo simulation and alpha-beta search 顶级的系统使用蒙特卡罗仿真和alpha-beta搜索 |

博弈程序发展现状

- 西洋跳棋: Chinook 在1994年打败了人类冠军 Marion Tinsley 。
- 黑白棋: The Logistello software 在1997年6: 0完败世界冠军。
- 国际象棋: Deep Blue 1997打败了人类冠军 Garry Kasparov.
- 围棋: 2016年7月18日, GoRatings公布最新世界排名, AlphaGo以3612分, 超越3608分的柯洁成为新的世界第一。
- 2017年10月19日国际学术期刊《自然》(Nature) 阿尔法元100: 0战胜哥哥阿尔法狗



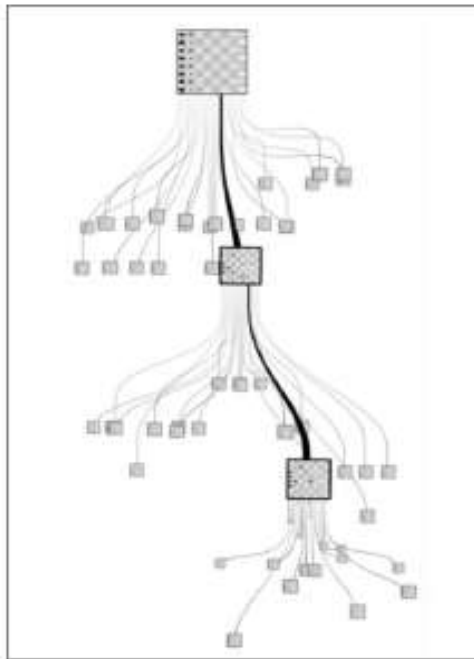
Secrets

- Many game programs are based on alpha-beta + iterative deepening + extended/singular search + transposition tables + huge databases + ...
- For instance, Chinook searched all checkers configurations with 8 pieces or less and created an endgame database of 444 billion board configurations
- The methods are general, but their implementation is dramatically improved by many specifically tuned-up enhancements (e.g., the evaluation functions) like an F1 racing car

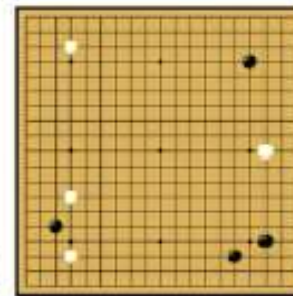
扩展阅读：Go vs. Chess 围棋与国际象棋

- Go has long been viewed as one of most complex game and most challenging of classic games for AI.

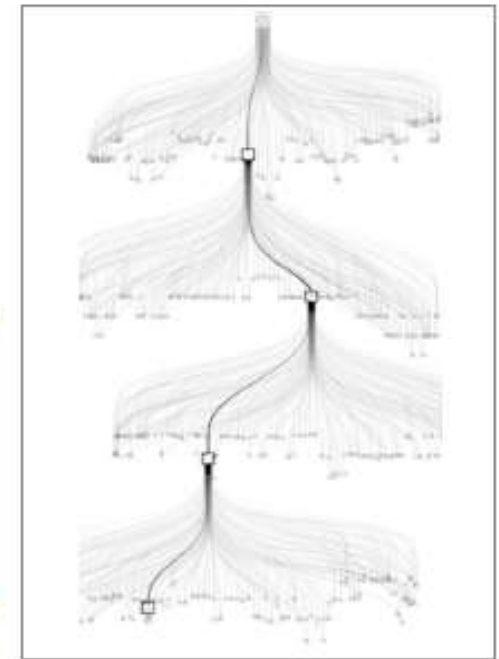
围棋一直被视为最复杂的博弈之一、而且是最具挑战性的AI经典博弈。



Chess ($b \approx 35$, $d \approx 80$)
 $8 \times 8 = 64$, possible games $\approx 10^{120}$



Go ($b \approx 250$, $d \approx 150$)
 $19 \times 19 = 361$, possible games $\approx 10^{170}$





Algorithm of AlphaGo AlphaGo的算法

□ Deep neural networks 深度神经网络

- value networks: used to evaluate board positions

价值网络：用于评估棋局

- policy networks: used to select moves.

策略网络：用于选择走子

□ Monte-Carlo tree search (MCTS) 蒙特卡罗树搜索 (MCTS)

- Combines Monte-Carlo simulation with value networks and policy networks.

将蒙特卡罗仿真与价值和策略网络相结合

□ Reinforcement learning 强化学习

- used to improve its play.

用于改进它的博弈水平。



Source: Mastering Go with deep networks and tree search
Nature, Jan. 28, 2016

About Monte-Carlo Methods 关于蒙特卡罗方法

- Monte-Carlo methods are a broad class of computational algorithms that rely on *repeated random sampling* to obtain numerical results.

蒙特卡罗方法是一大类计算算法，它凭借重复随机采样来获得数值结果。

- They tend to follow a particular pattern:

它们往往遵循如下特定模式：

- define a domain of possible inputs;
定义一个可能的输入域；
- generate inputs randomly from a probability distribution over the domain;
从该域的一个概率分布随机地生成输入；
- perform a deterministic computation on the inputs;
对该输入进行确定性计算；
- aggregate the results.
将结果聚合。

Example: Approximating π by Monte-Carlo Method 用蒙特卡罗方法估计 π

- Given that circle and square have a ratio of areas that is $\pi/4$, the value of π can be approximated using a Monte-Carlo method:

鉴于圆形与正方形面积之比为 $\pi/4$ ，则 π 的值可采用蒙特卡罗方法近似得出：

- a) Draw a square on the ground, then inscribe a circle within it.

先画出一个正方形，然后在其中画一个圆弧。

- b) Uniformly scatter some objects of uniform size over the square.

将尺寸大小一致的小颗粒散落在正方形上。

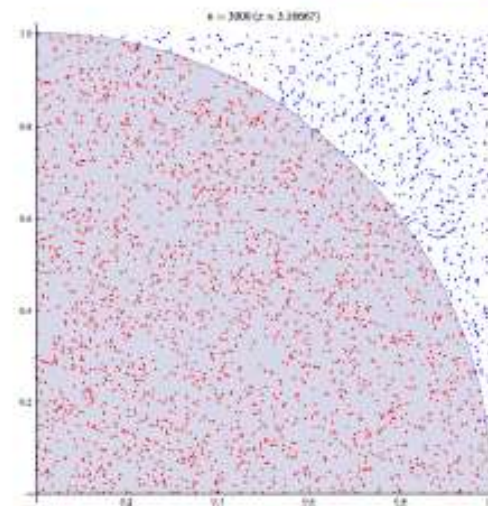
- c) Count the number of objects inside the circle and the square.

计算圆形和正方形中小颗粒的数量和总的数量。

- d) The ratio of the two counts is an estimate of the ratio of the two areas, which is $\pi/4$. Multiply the result by 4 to estimate π .

两个数量之比为两个面积的估算，即 $\pi/4$ 。结果乘以4得出 π 。

Source: Wikipedia



Family of Monte-Carlo Methods 蒙特卡罗方法的家族

是一种覆盖系统各种状态的概率分布、概率测量或者频率分布。来获得热力学性质和最小能量结构的方法

□ Classical Monte-Carlo: 经典蒙特卡罗

samples are drawn from a probability distribution, often the classical Boltzmann distribution;
样本来自于概率分布，往往是经典的玻兹曼分布；

□ Quantum Monte-Carlo: 量子蒙特卡罗

random walks are used to compute quantum-mechanical energies and wave functions;
采用随机走查方法来计算量子力学的能量和波函数；

□ Volumetric Monte-Carlo: 容积式蒙特卡罗

random number generators are used to generate volumes per atom or to perform other types of geometrical analysis;

采用随机数生成的方法产生每个原子的容量、或进行其它类型的几何分析。

□ Kinetic Monte-Carlo: 动力学蒙特卡罗

simulate processes using scaling arguments to establish timescales or by introducing stochastic effects into molecular dynamics.

仿真过程采用尺度分析来建立时间尺度、或者将随机效应引入分子动力学。

- MCTS将蒙特卡罗仿真与博弈树搜索相结合。

- 和minimax一样，每个节点对应于一个的博弈状态。

- 不同于minimax, 节点的值通过蒙特卡罗仿真来估值。

where: 其中

$N(a)$ = the number of simulations 仿真的数量

Algorithm of AlphaGo AlphaGo的算法

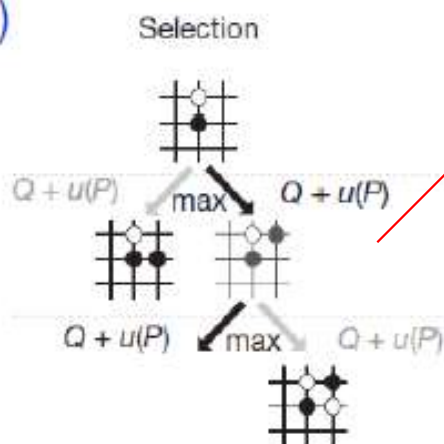
- It uses a combination of machine learning and tree search techniques, combined with extensive training, both from human and computer play.
采用机器学习和树搜索技术相结合的方式，并且用人类和计算机走棋的棋局进行大量的训练。
- Two deep neural networks 两个深度神经网络
 - **value networks** to evaluate board positions and **policy networks** to select moves.
价值网络来评价棋盘位置、策略网络来选择走棋。
- Tree search 树搜索
 - **Monte Carlo tree search (MCTS).**
蒙特卡罗树搜索 (MCTS)
- Reinforcement learning 强化学习
 - be used to improve its play.
用于改善其走棋。



Monte-Carlo Tree Search in AlphaGo AlphaGo的蒙特卡罗树搜索

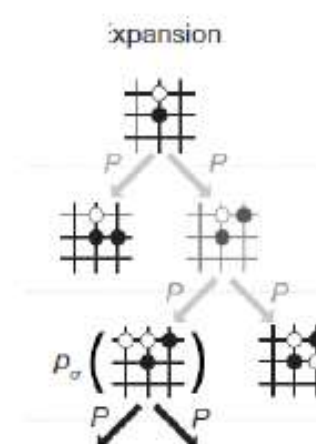
Source: Nature, Jan. 28, 2016

(a)



将深度神经网络与
树搜索技术有机结合

(b)



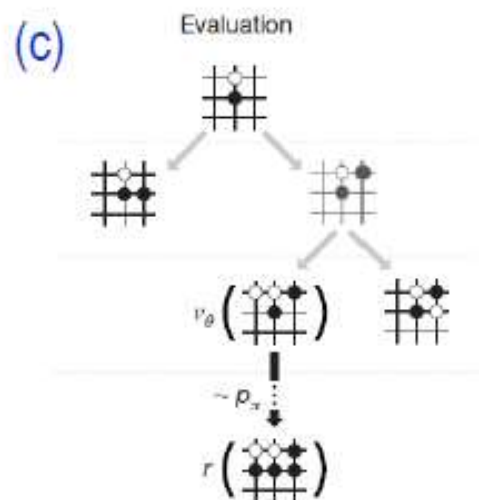
(a) **Selection:** Each simulation traverses the tree by selecting edge with maximum action value Q + bonus $u(P)$ that depends on a stored prior probability P for that edge.

选择：每次仿真通过选择边与最大动作值 Q +奖励 $u(P)$ 对搜索树进行遍历，依赖于该条边存储的先验概率 P 。

(b) **Expansion:** The leaf node may be expanded; the new node is processed once by the policy network p_σ and the output probabilities are stored as prior probabilities P for each action.

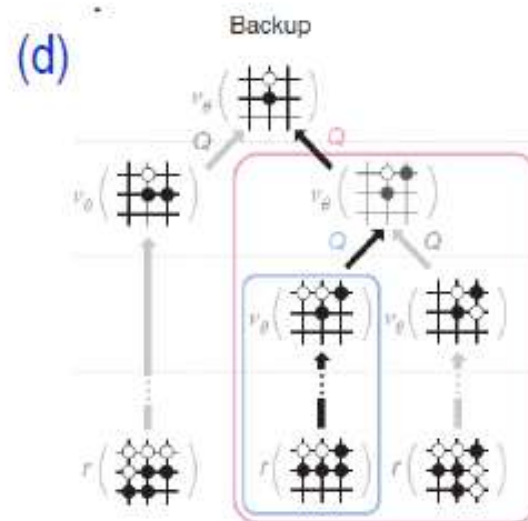
扩展：叶节点可以扩展；新的节点先由策略网络 p_σ 处理，然后其输出概率存储为每个动作的先验概率 P 。

Monte-Carlo Tree Search in AlphaGo



(c) **Evaluation** (Simulation): The leaf node is evaluated in two ways: 1) using the **value network** v_θ ; 2) by running a rollout to the end of the game with the fast **rollout policy** p_π , then computing the winner with function r .

评价（仿真）：叶节点用两种方法评价：1) 使用价值网络 v_θ ；2) 使用快速走子策略 p_π 运行到博弈结束，然后用函数 r 计算出胜者。



(d) **Backup** (Back propagation): Action values Q are updated to track the mean value of all evaluations $r(\cdot)$ and $v_\theta(\cdot)$ in the subtree below that action.

后援（反向传播）：更新动作值 Q 来跟踪在该动作下面子树的所有评价函数 $r(\cdot)$ 和 $v_\theta(\cdot)$ 的平均值。

Neural Network Training Pipeline in AlphaGo AlphaGo的神经网络训练管线

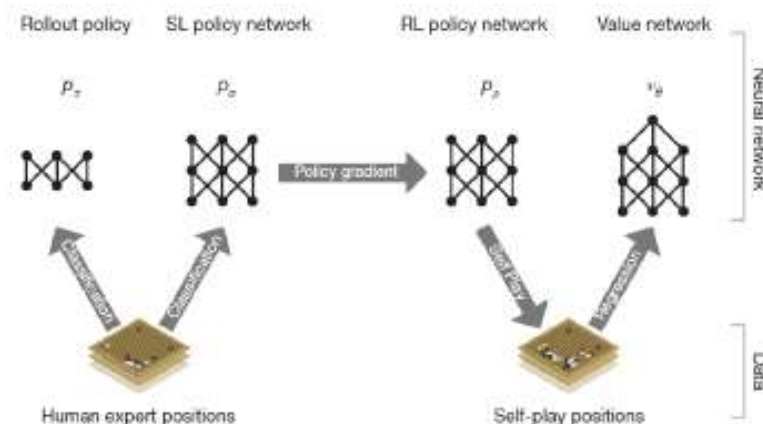
- A fast rollout policy p_π and supervised learning (SL) policy network p_σ are trained to predict human expert moves in a data set of positions.

快速走子策略 p_π 与有监督学习 (SL) 策略网络 p_σ 用棋局数据集进行训练，来预测人类的走棋。

- A reinforcement learning (RL) policy network p_ρ is initialized to the SL policy network, and then improved by policy gradient learning to maximize the outcome.
- 强化学习 (RL) 策略网络 p_ρ 被初始化为 SL 策略网络，然后通过策略梯度学习使输出最大化。

- A new data set is generated by playing games of self-play with the RL policy network.
- 经过自我对弈和 RL 策略网络生成一个新的数据集。

- A value network v_θ is trained by regression to predict the expected outcome.
- 通过回归训练价值网络 v_θ 来预测所期望的输出。



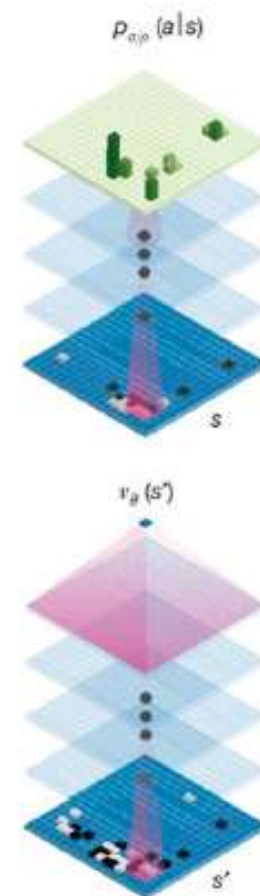
Neural Network Architecture in AlphaGo AlphaGo的神经网络架构

□ The **policy network** 策略网络

- input: the board position s
输入：棋盘位置 s
- passes s through convolutional layers with parameters σ or ρ
将 s 穿过具有参数 σ 或 ρ 的卷积层
- outputs: a probability distribution over legal moves a .
输出：一个合法走子 a 的概率分布。

□ The **value network** 价值网络

- input: the board position s'
输入：棋盘位置 s'
- similarly uses many convolutional layers with parameters θ
同样采用具有参数 θ 的卷积层
- output: a scalar value $v_\theta(s')$ that predicts the expected outcome.
输出：一个预测期望输出的标量值 $v_\theta(s')$



Final Project 参考选题

- (1) 采用 α - β 算法实现井字棋游戏。
- (2) 采用本章技术实现自选游戏。

要求:

- (1) 图形化界面。
- (2) 随机选取先手后手。
- (3) 可以人-计算机或计算机-计算机

