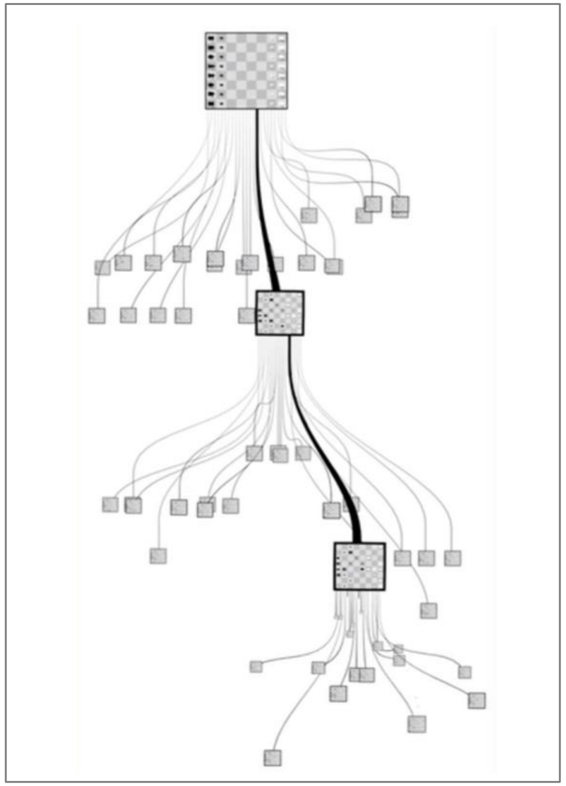
Contents:

- 5.1. Games
- 5.2. Optimal Decisions in Games
- 5.3. Alpha-Beta Pruning
- 5.4. Imperfect Real-time Decisions
- 5.5. Stochastic Games
- 5.6. Monte-Carlo Methods

Go vs. Chess 围棋与国际象棋

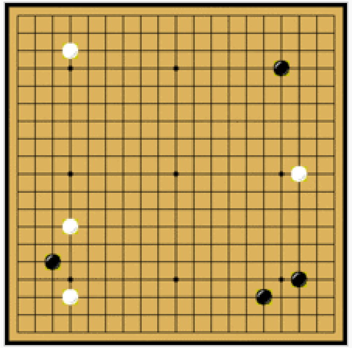
- Go has long been viewed as one of most complex game and most challenging of classic games for AI.

  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。



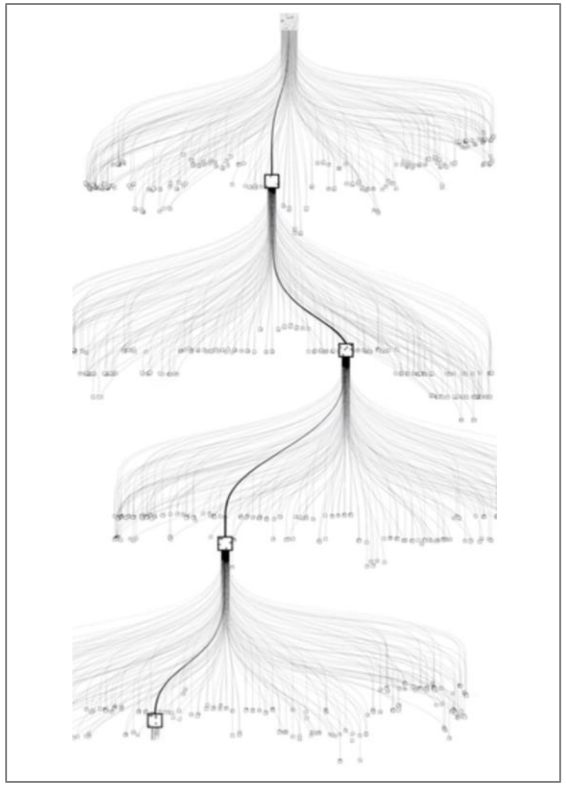
Chess ($b \approx 35$, $d \approx 80$)

8 x 8 = 64, possible games $\approx 10^{120}$



Go ($b \approx 250$, $d \approx 150$)

19 x 19 = 361, possible games $\approx 10^{170}$



Algorithm of AlphaGo AlphaGo的算法

- Deep neural networks 深度神经元网络
  - 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

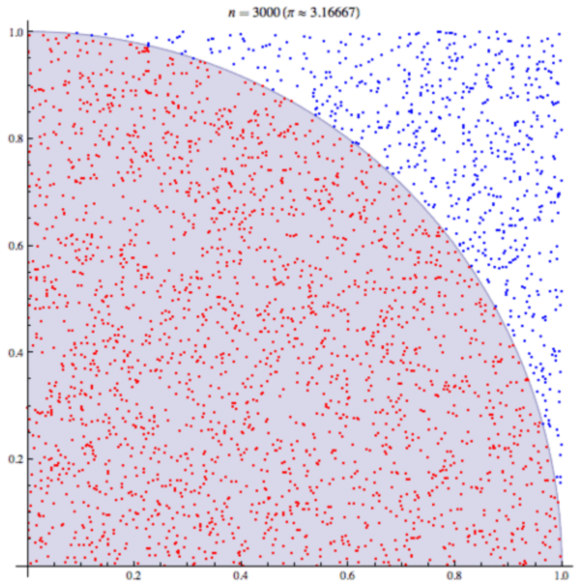
About Monte-Carlo Methods 关于蒙特卡罗方法

- 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

- **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。

Monte-Carlo Tree Search (MCTS) 蒙特卡罗树搜索 (MCTS)

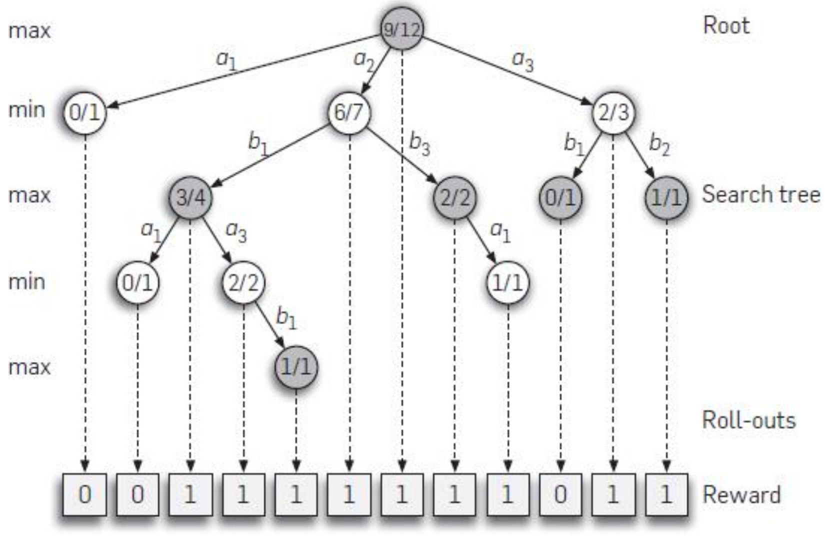
- MCTS combines Monte-Carlo simulation with game tree search.
  MCTS将蒙特卡罗仿真与博弈树搜索相结合。
- Like minimax, each node corresponds to a single state of game.
  和minimax一样，每个节点对应于一个的博弈状态。
- Unlike minimax, the values of nodes are estimated by Monte-Carlo simulation.
  不同于minimax，节点的值通过蒙特卡罗仿真来估值。

$W(a)/N(a)$ = the value of action a 动作a的值

where: 其中

$W(a)$ = the total reward 总的奖励

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



Source: Communications of the ACM, Mar. 2012, 55(3), pp. 106-113

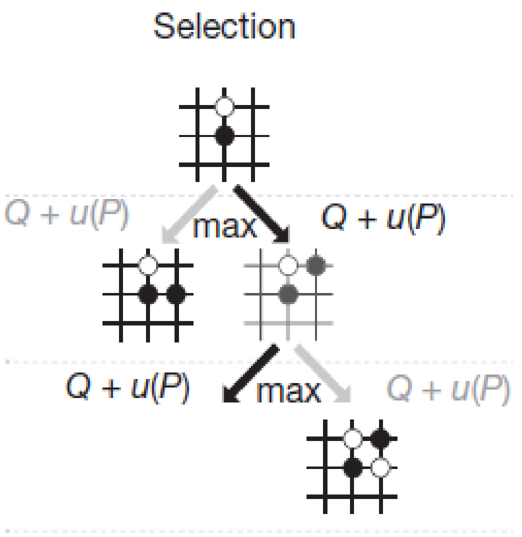
Algorithm of AlphaGo AlphaGo的算法

- 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。

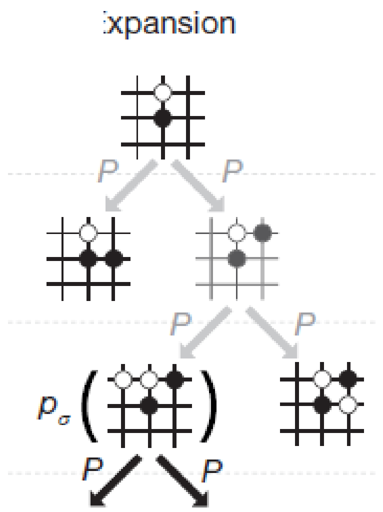
Monte-Carlo Tree Search in AlphaGo AlphaGo的蒙特卡罗树搜索

Source: Nature, Jan. 28, 2016

(a)



(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.

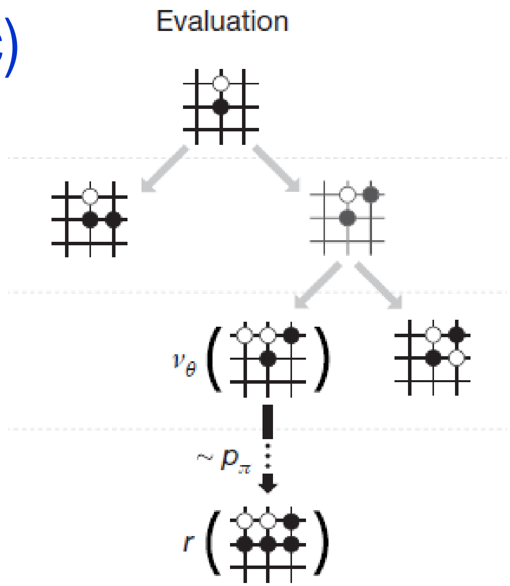
选择：每次仿真通过选择边与最大动作值$Q + $奖励$u(P)$对搜索树进行遍历，依赖于该条边存储的先验概率$P$。

(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.

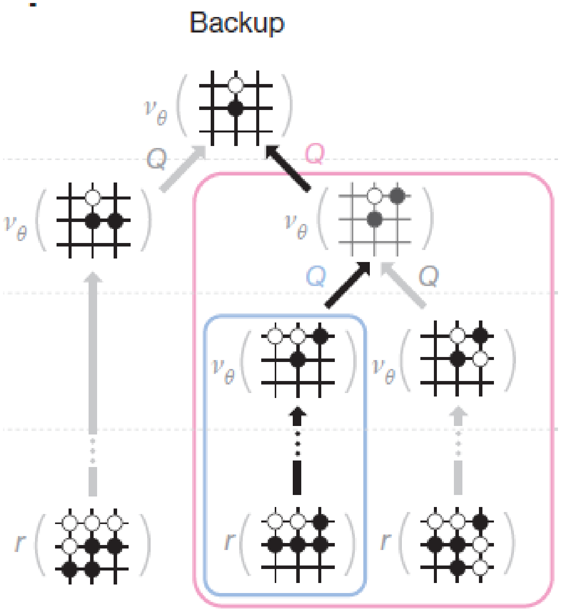
扩展：叶节点可以扩展；新的节点先由策略网络p_σ处理，然后其输出概率存储为每个动作的先验概率P。

Monte-Carlo Tree Search in AlphaGo

(c)



(d)



(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。

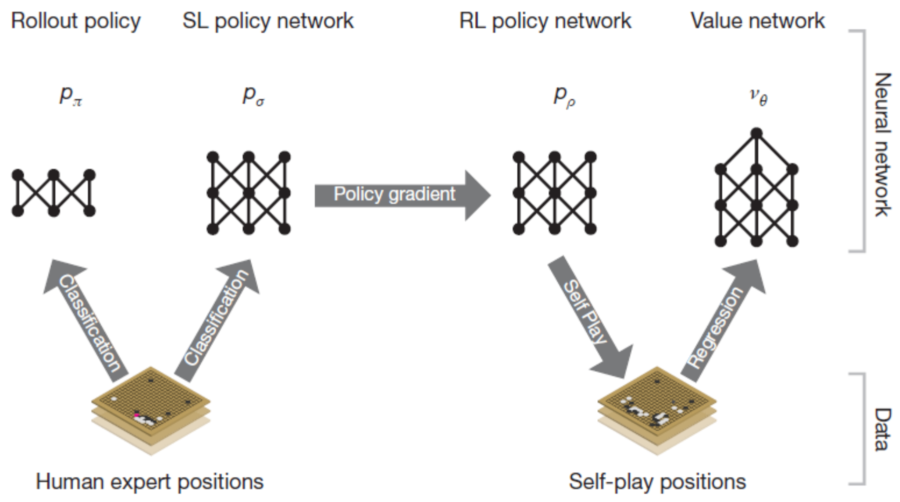
Neural Network Training Pipeline in AlphaGo AlphaGo的神经网络训练管线

- 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.

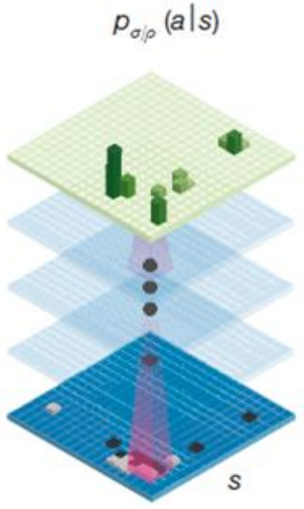
  经过自我对弈和PL策略网络生成一个新的数据集。

- 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$v_\theta(s')$