Mastering the game of Go with deep neural networks and tree search

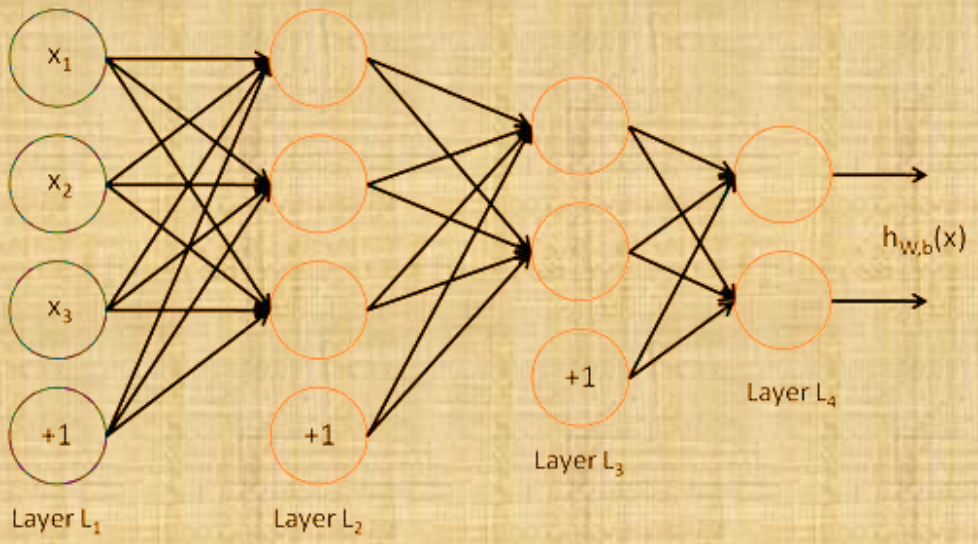
使用深度神经网络和搜索树战胜围棋

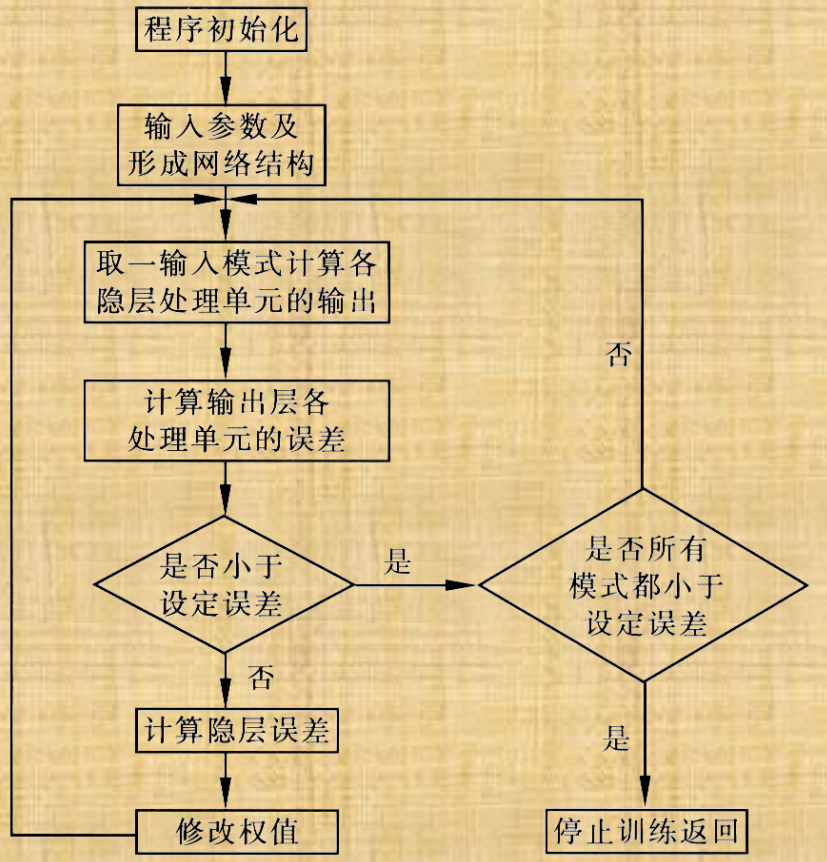
江浩

背景

- 暴力搜索：“深蓝”
- 基于卷积神经网络（CNN）做落子的预测
  输入（当前盘面s），输出（下一步每个位置落子的概率a）
  预测准确率（25%~55%）（6d水平）
- CNN + MCTS（蒙特卡洛树搜索）
- Value network + Policy network + MCTS （Alpha GO）

介绍


x1
x2
x3
+1
+1
+1
+1
hW,b(x)
Layer L1
Layer L2
Layer L3
Layer L4


程序初始化
输入参数及
形成网络结构
取一输入模式计算各
隐层处理单元的输出
计算输出层各
处理单元的误差
是否小于
设定误差
是
否
计算隐层误差
修改权值
是否所有
模式都小于
设定误差
否
是
停止训练返回

介绍

- CNN（卷积神经网络）：应用于图像识别中的神经网络，作者使用了与CNN网络结构相似的Policy network和Value network。
- Policy network（策略网络）：以当前盘面为输入，输出下一步棋在棋盘其他空位上的落子概率。
- Rollout policy（快速走棋策略）:利用局部特征和线性模型训练出来,相比于策略网络速度快但精度低。
- Value network（价值网络）：以当前盘面作为输入，输出当前盘面胜利和失败的概率。

AlphaGO下棋思想

- Step 1：分析判断全局的形势
- Step 2：分析判断局部的棋局找到几个可能的落子点
- Step 3：预测接下来几步的棋局变化，判断并选择最佳的落子点。

人

- ----------

- Step 1：基于策略网络来预测未来的下一步走法，直到L步。
- Step 2：结合两种方式来对未来到L的走势进行评估，一个是使用价值网络进行评估，判断赢面，一个是使用快速走棋策略做进一步的预测直到比赛结束得到模拟的结果。综合两者对预测到未来L步走法进行评估。
- Step 3：评估完，将评估结果作为当前棋局下的下一步走法的估值。
- Step 4：结合下一步走法的估值和策略网络进行再一次的模拟，如果出现同样的走法，则对走法的估值取平均，反复循环上面的步骤到n次。然后选择选择次数最多的走法作为下一步。

AlphaGO

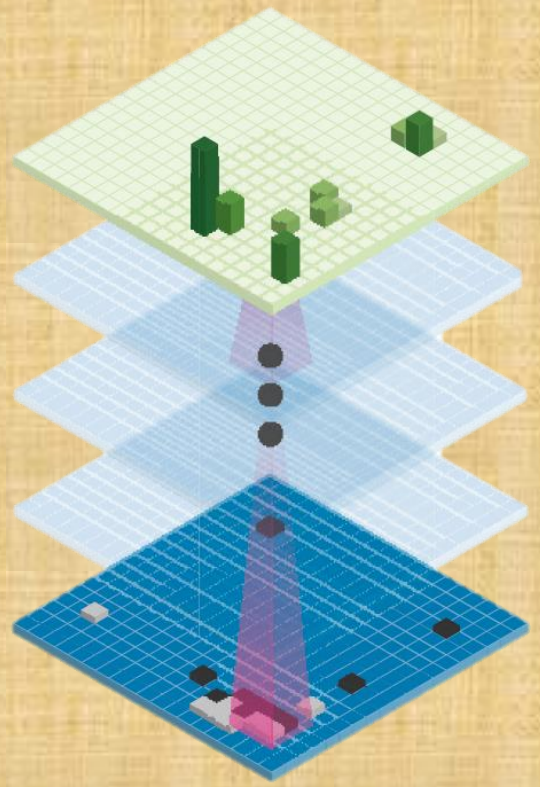
AlphaGO下棋思想

- “策略网络”观察棋盘布局企图找到较好的下法。

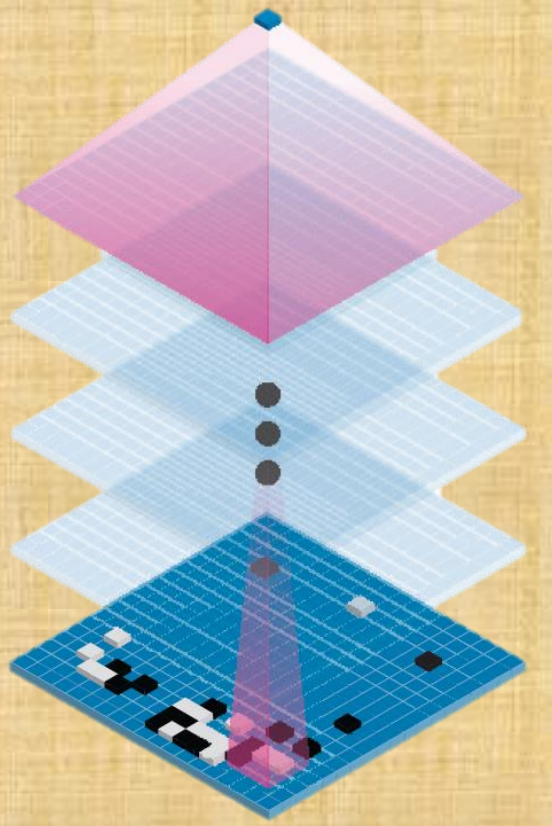
- “价值网络”和“快速走棋策略”则预测这样下的话对方棋手赢棋的可能。

策略网络预测下一步落子的位置及概率

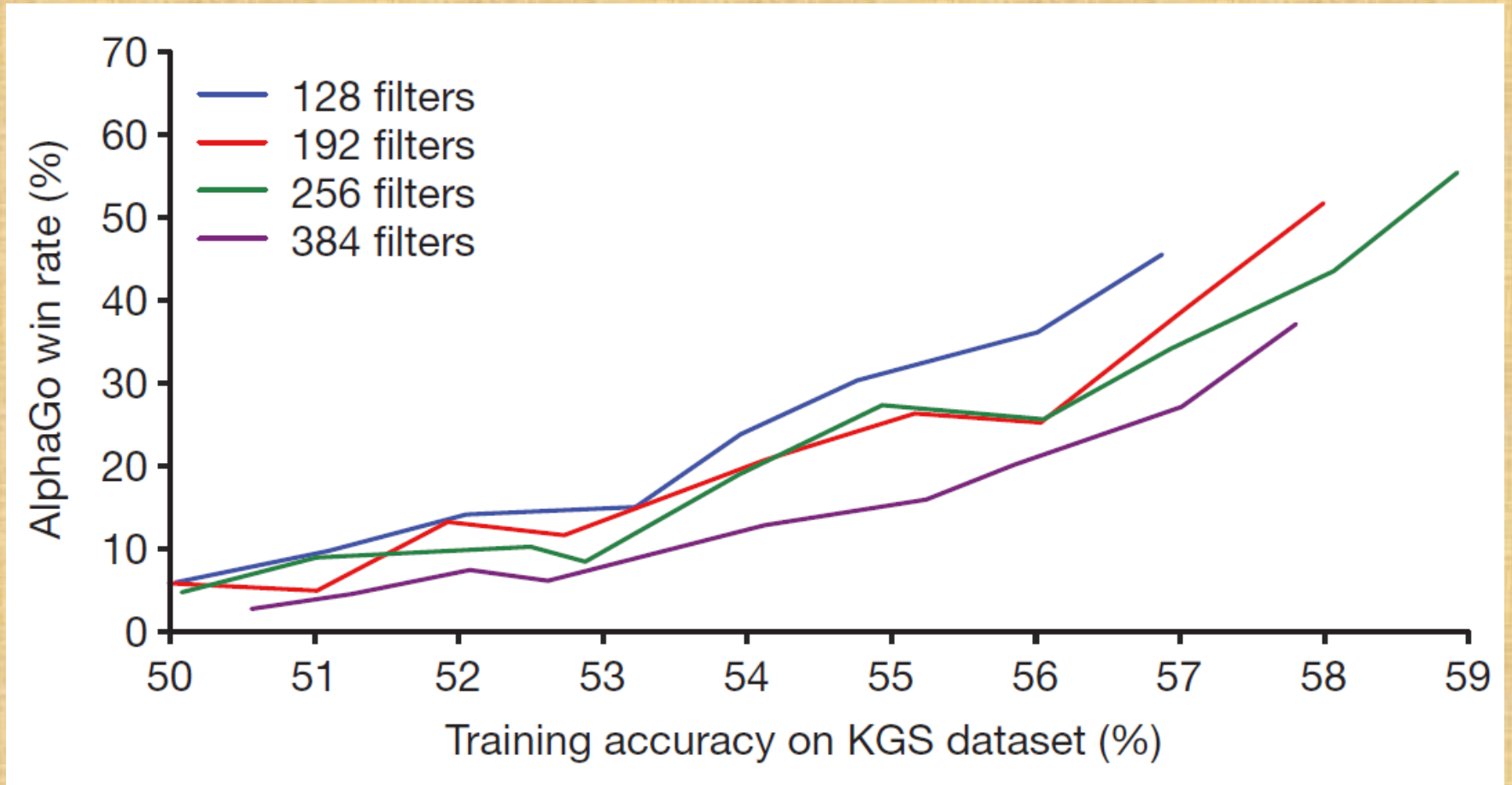
Policy network



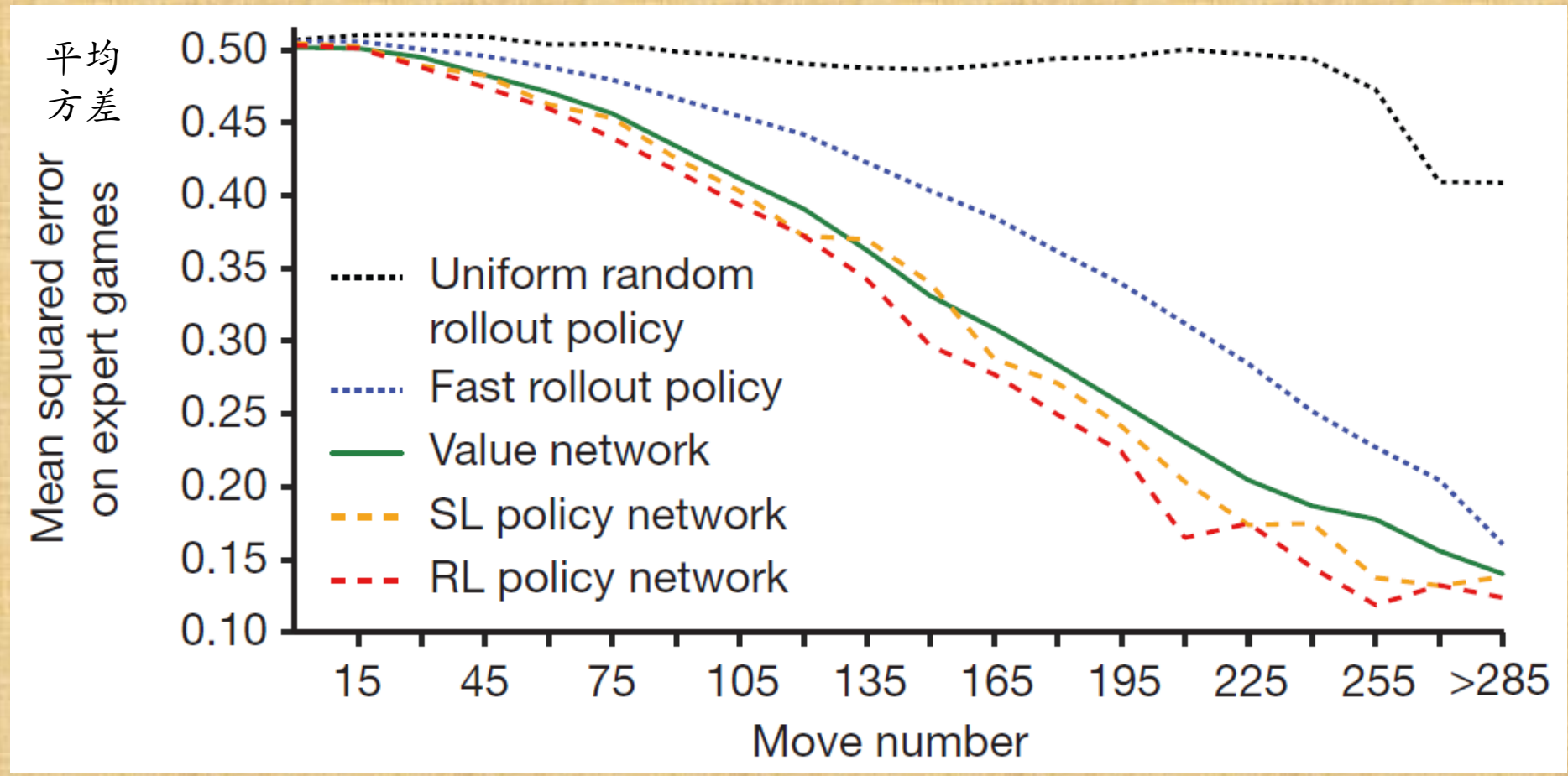
Value network



价值网络评估当前盘面胜利和失败的概率

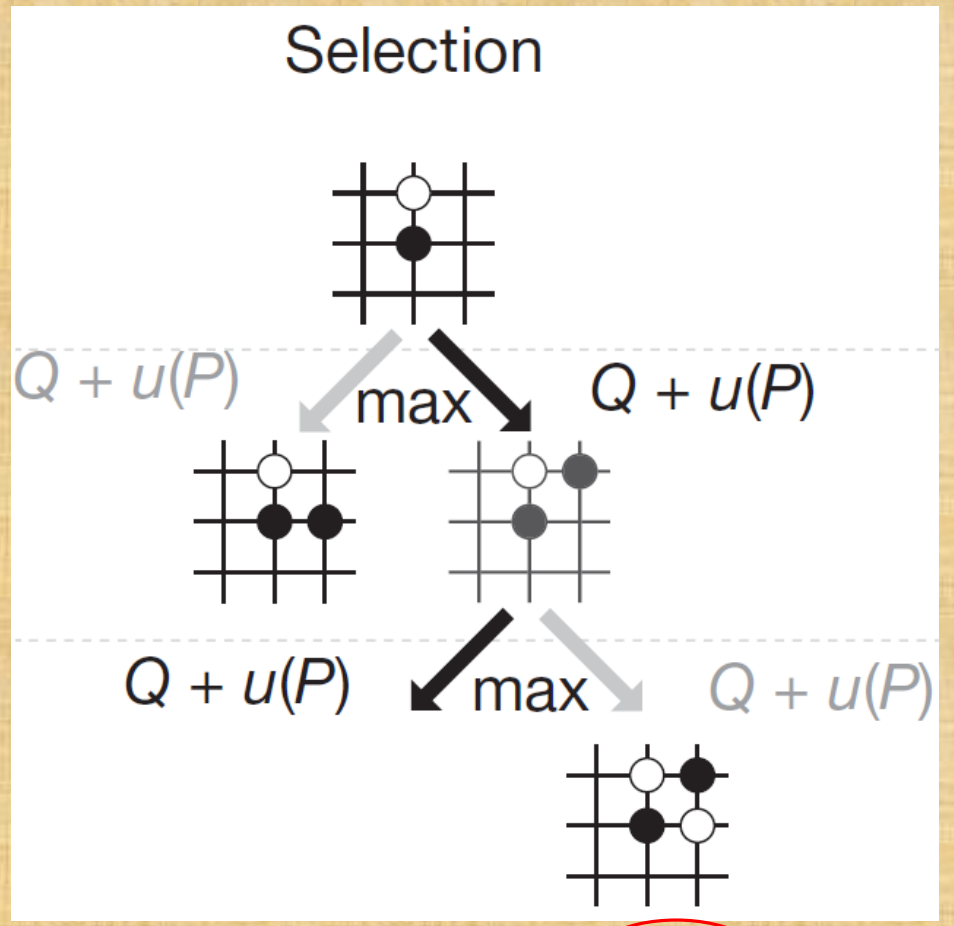


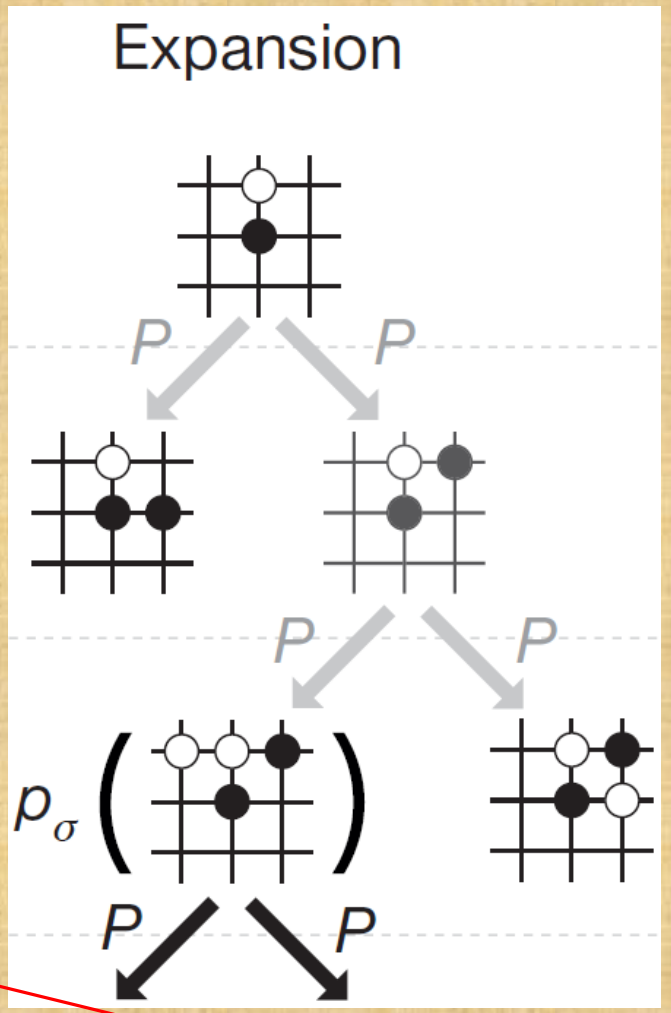
训练的准确率和AlphaGO的胜率的关系（策略网络）



不同的神经网络在给定的盘面（X轴）评估的准确率（Y轴）

蒙特卡洛数搜索（MCTS）





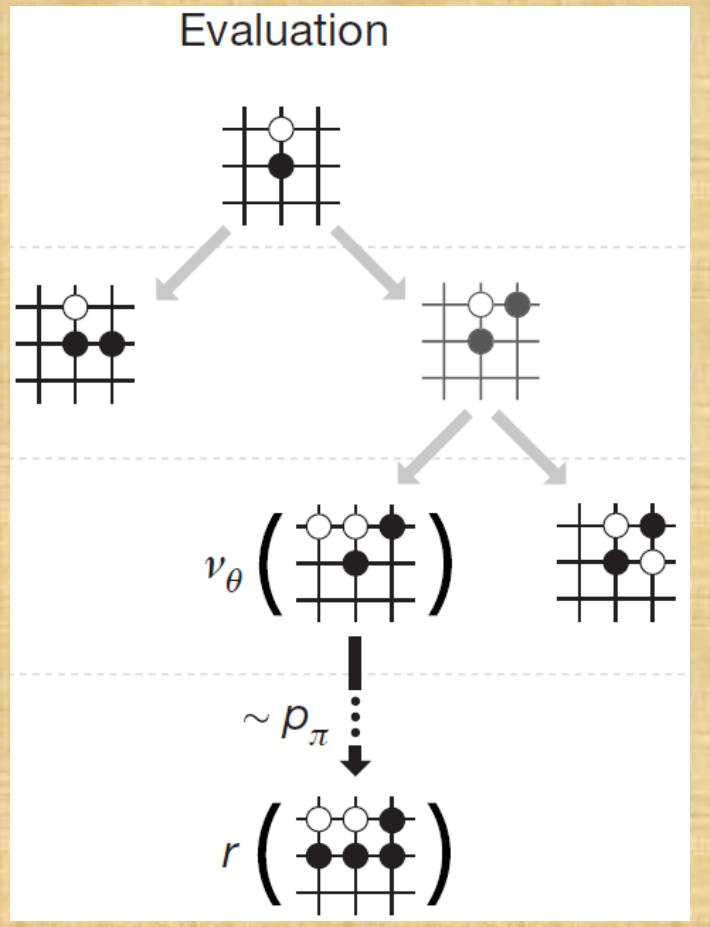
t	模拟的第t步
a_t	第t步选择的落子
$Q(s_t,a)$	行动价值
$u(s_t,a)$	额外奖励
$N(s,a)$	访问次数
$P(s,a)$	先验概率
$V(s_L)$	盘面总价值
$v_\theta(s_L)$	盘面价值
z_L	盘面价值

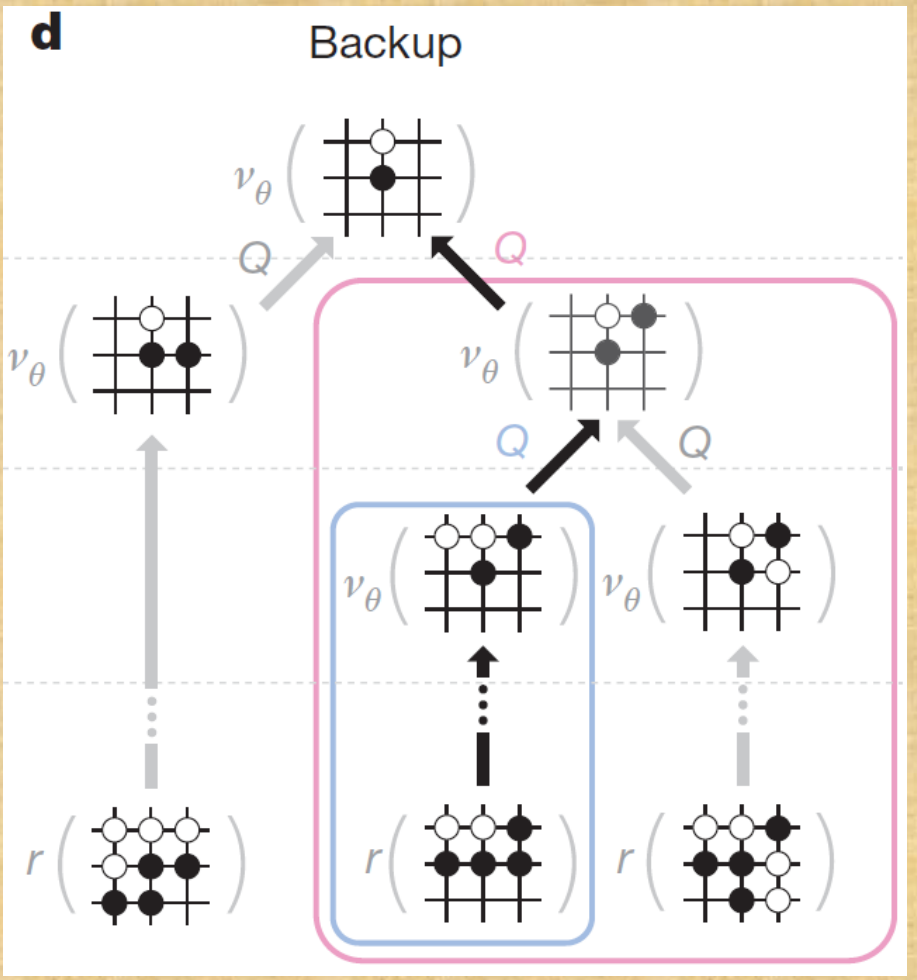
$$a_t = \underset{a}{\operatorname{argmax}}(Q(s_t,a) + u(s_t,a))$$

$$u(s,a) \propto \frac{P(s,a)}{1+N(s,a)}$$

$$N(s,a) = \sum_{i=1}^{n} 1(s,a,i)$$

蒙特卡洛数搜索（MCTS）

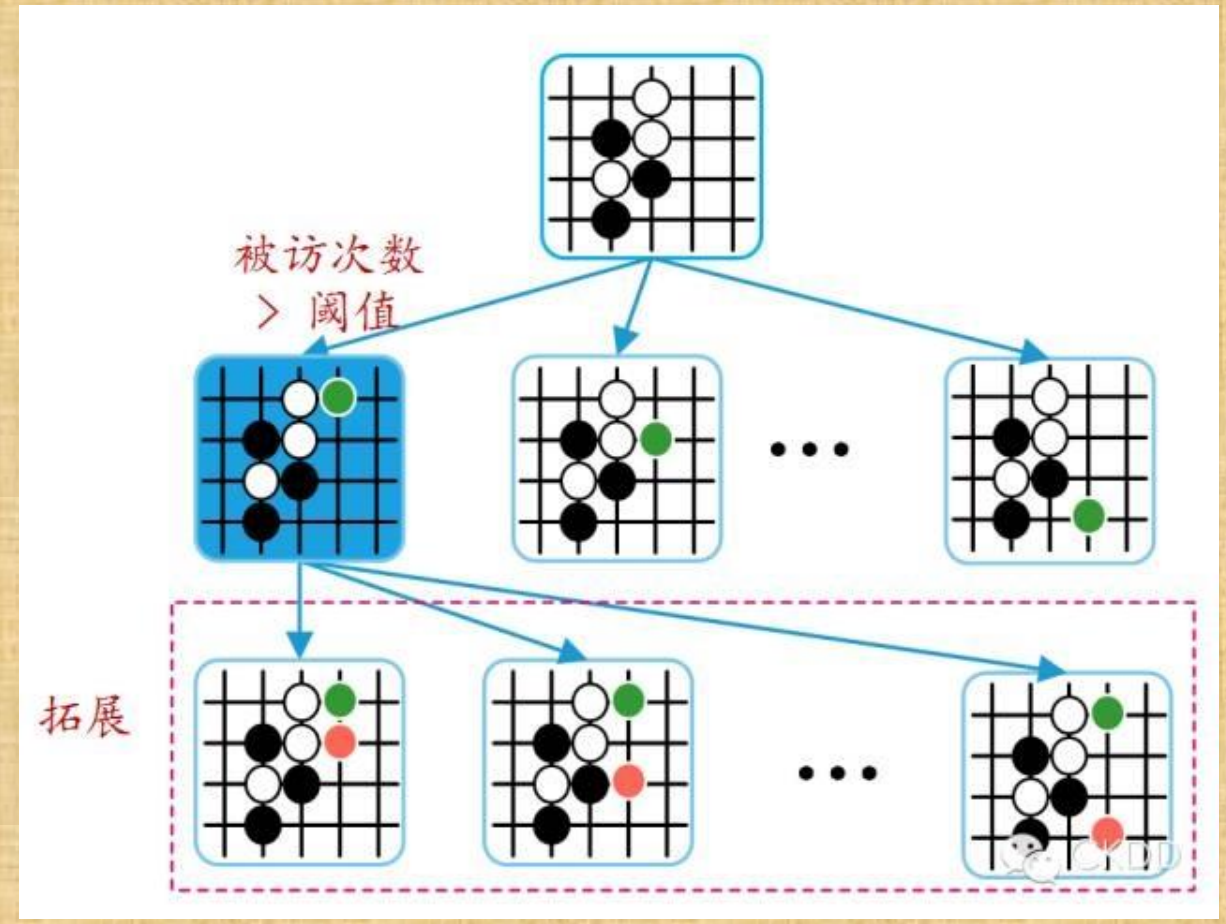


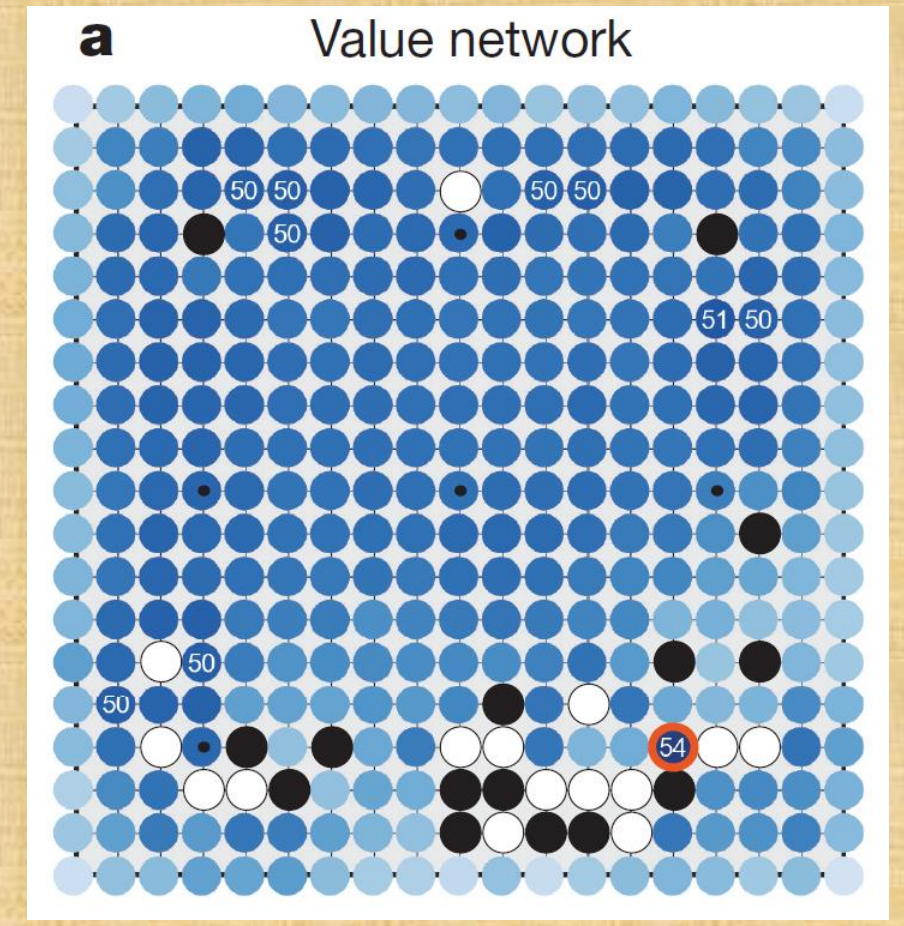


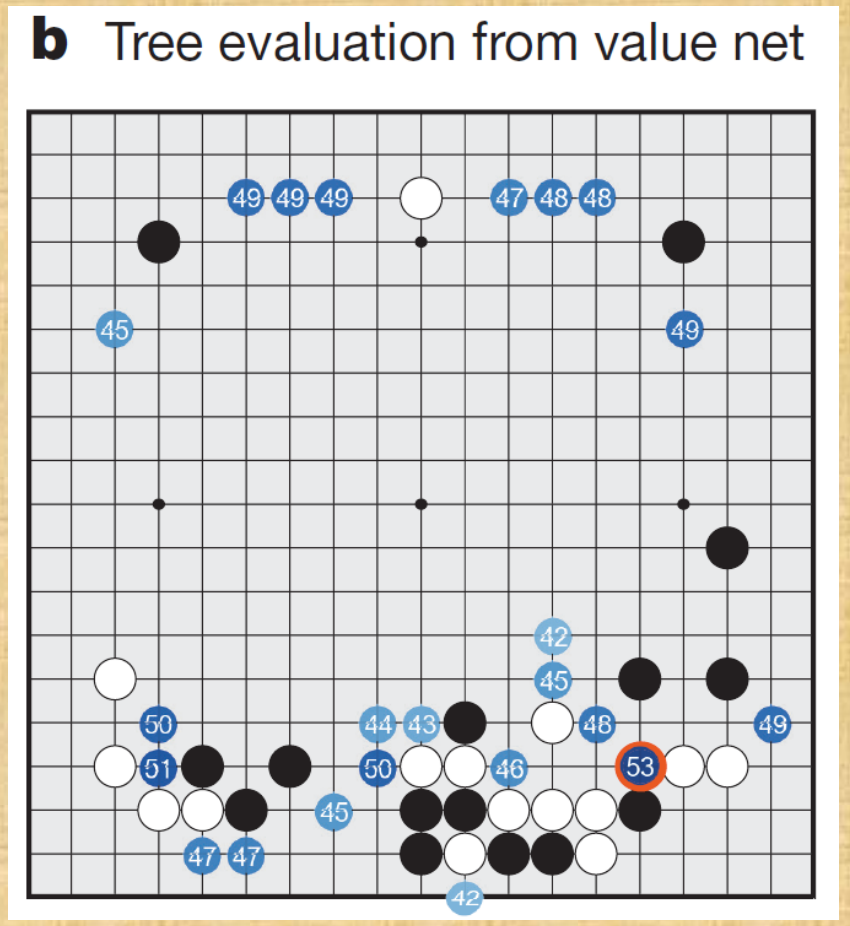
t	模拟的第t步
a_t	第t步选择的落子
$Q(s_t, a)$	行动价值
$u(s_t, a)$	额外奖励
$N(s, a)$	访问次数
$P(s, a)$	先验概率
$V(s_L)$	盘面总价值
$v_\theta(s_L)$	盘面价值
z_L	盘面价值

$$V(s_L) = (1-\lambda)v_\theta(s_L) + \lambda z_L$$

$$Q(s,a) = \frac{1}{N(s,a)} \sum_{i=1}^{n} 1(s,a,i) V(s_L^i)$$

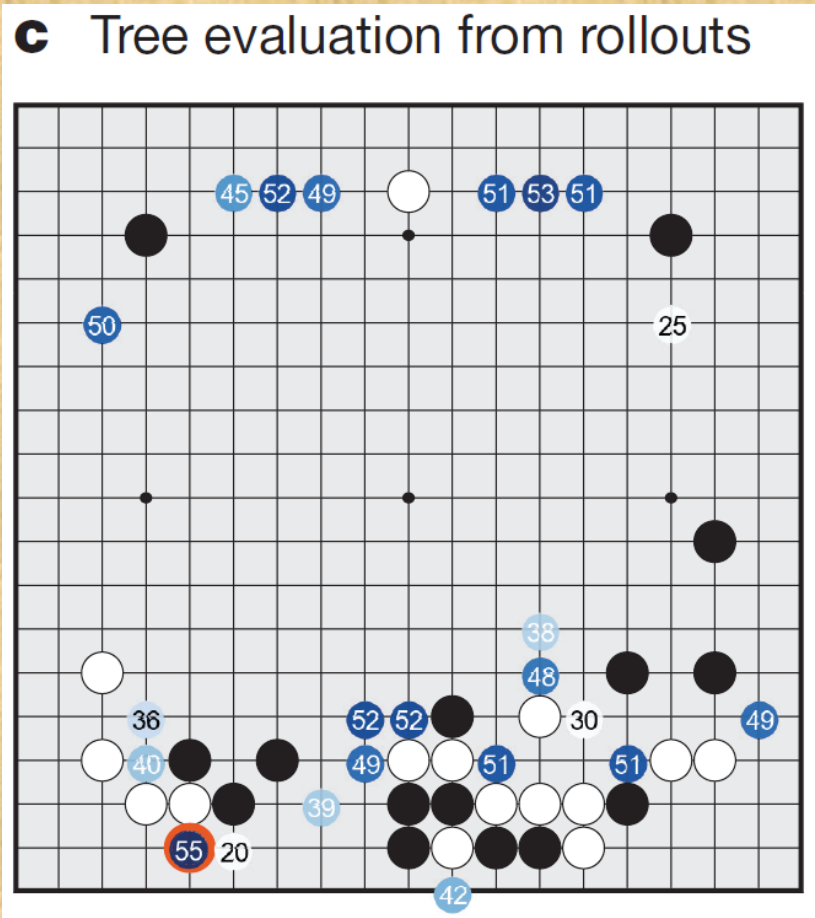

被访次数
> 阈值
拓展

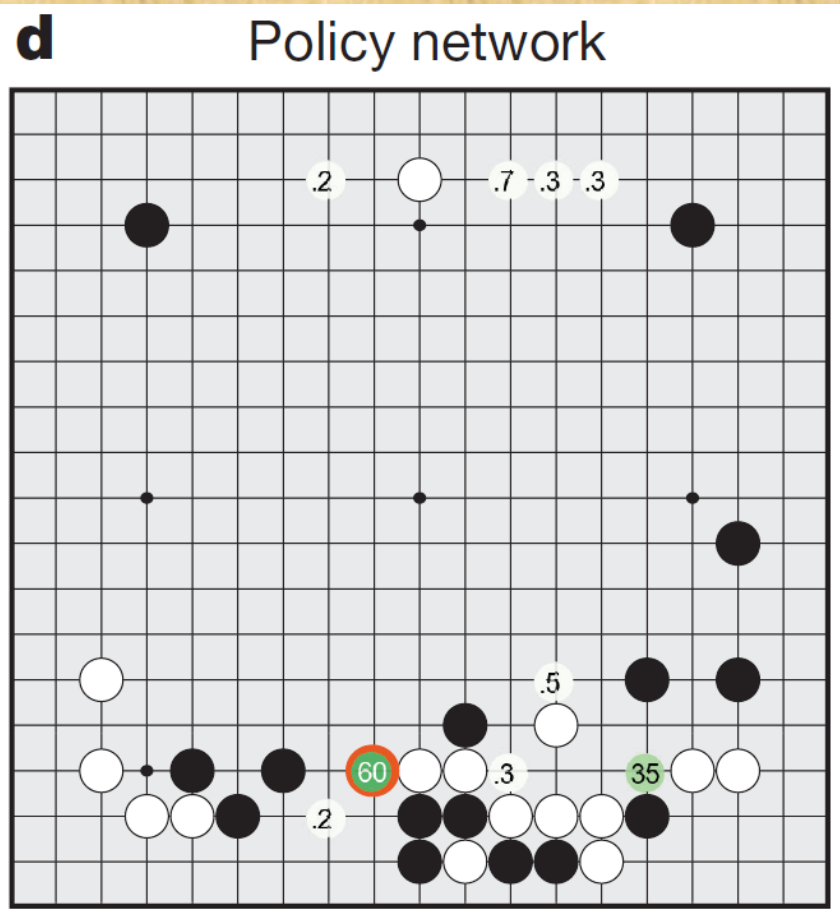




V值λ=0

$$V(s_L) = (1-\lambda)v_\theta(s_L) + \lambda z_L$$





V値λ=1

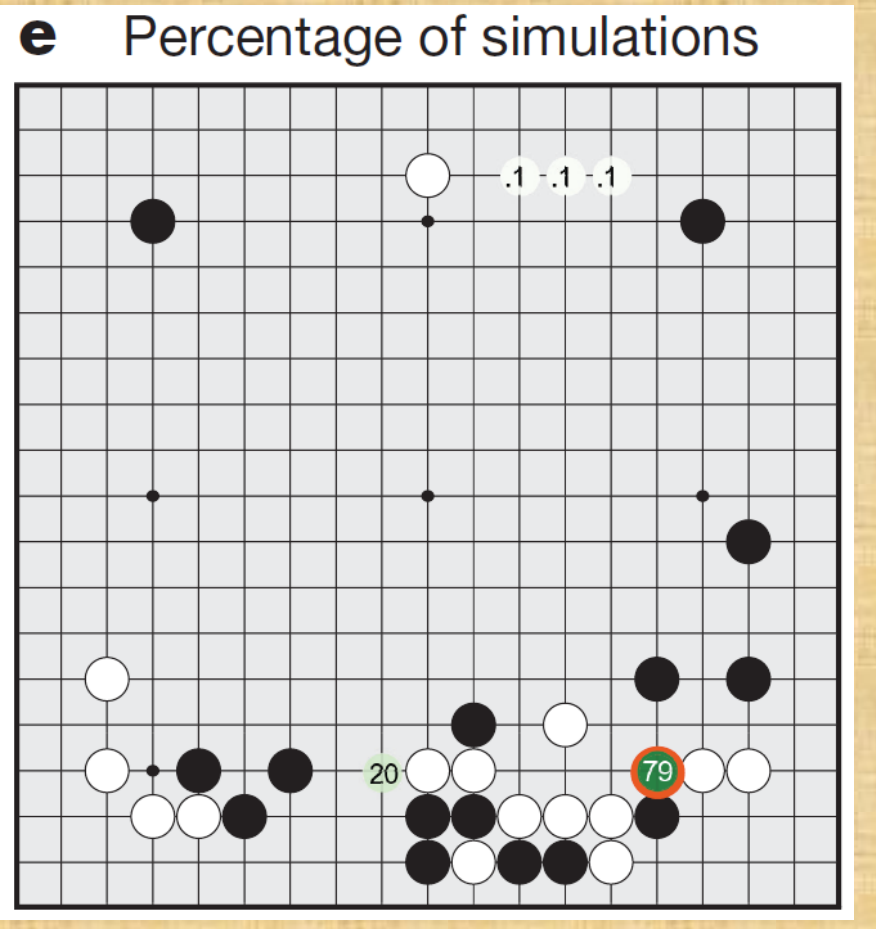
$$V(s_L) = (1-\lambda)v_\theta(s_L) + \lambda z_L$$

模拟百分比

数字表示蒙特卡洛搜索树遍历
到每个点的次数的百分比

$$a_t = \underset{a}{\operatorname{argmax}}(Q(s_t, a) + u(s_t, a))$$

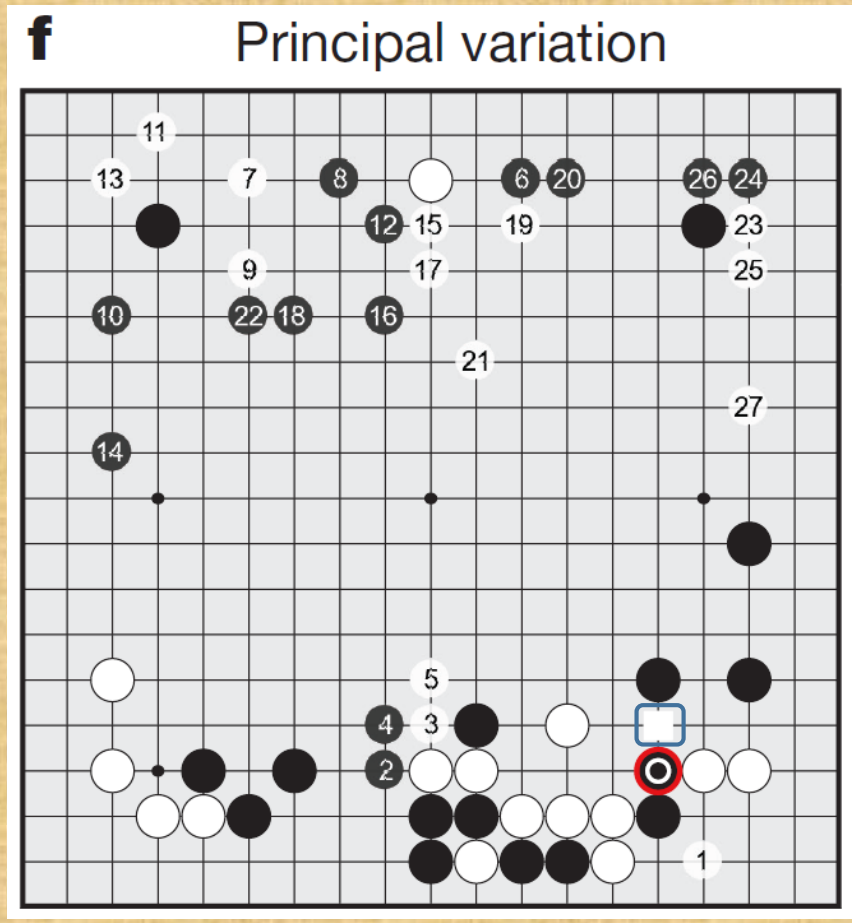
$$u(s, a) \propto \frac{P(s, a)}{1 + N(s, a)}$$



主要变化

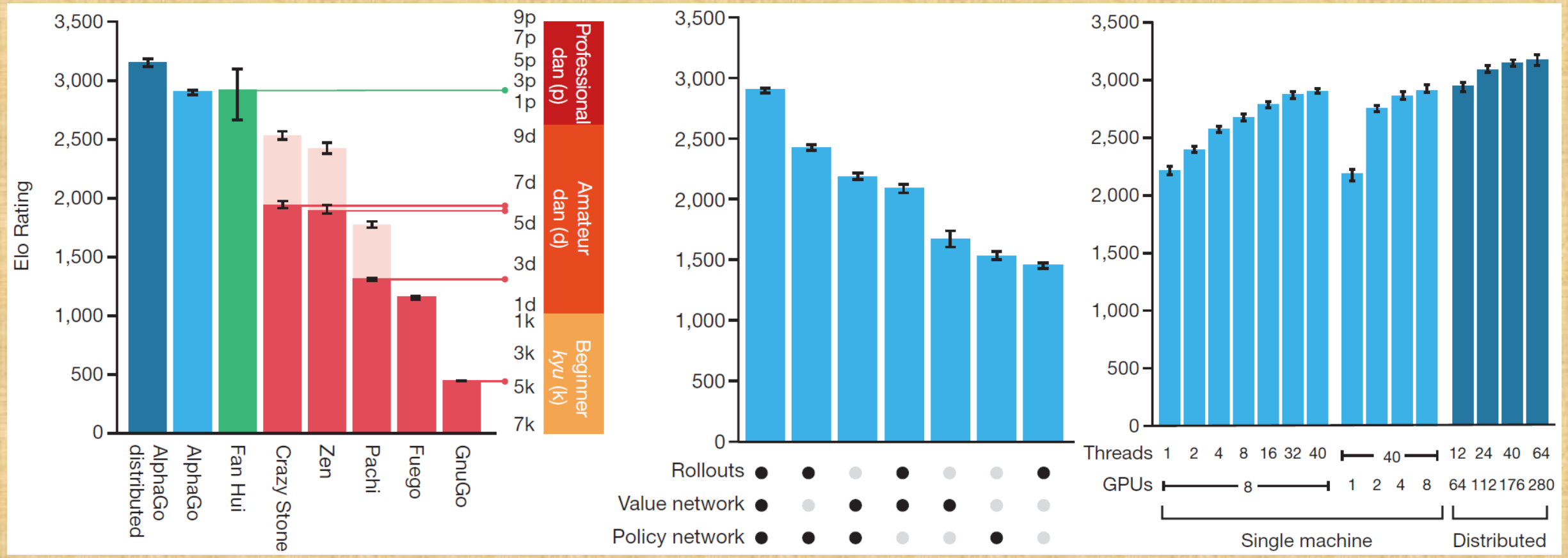
AlphaGO如何选择落子的?

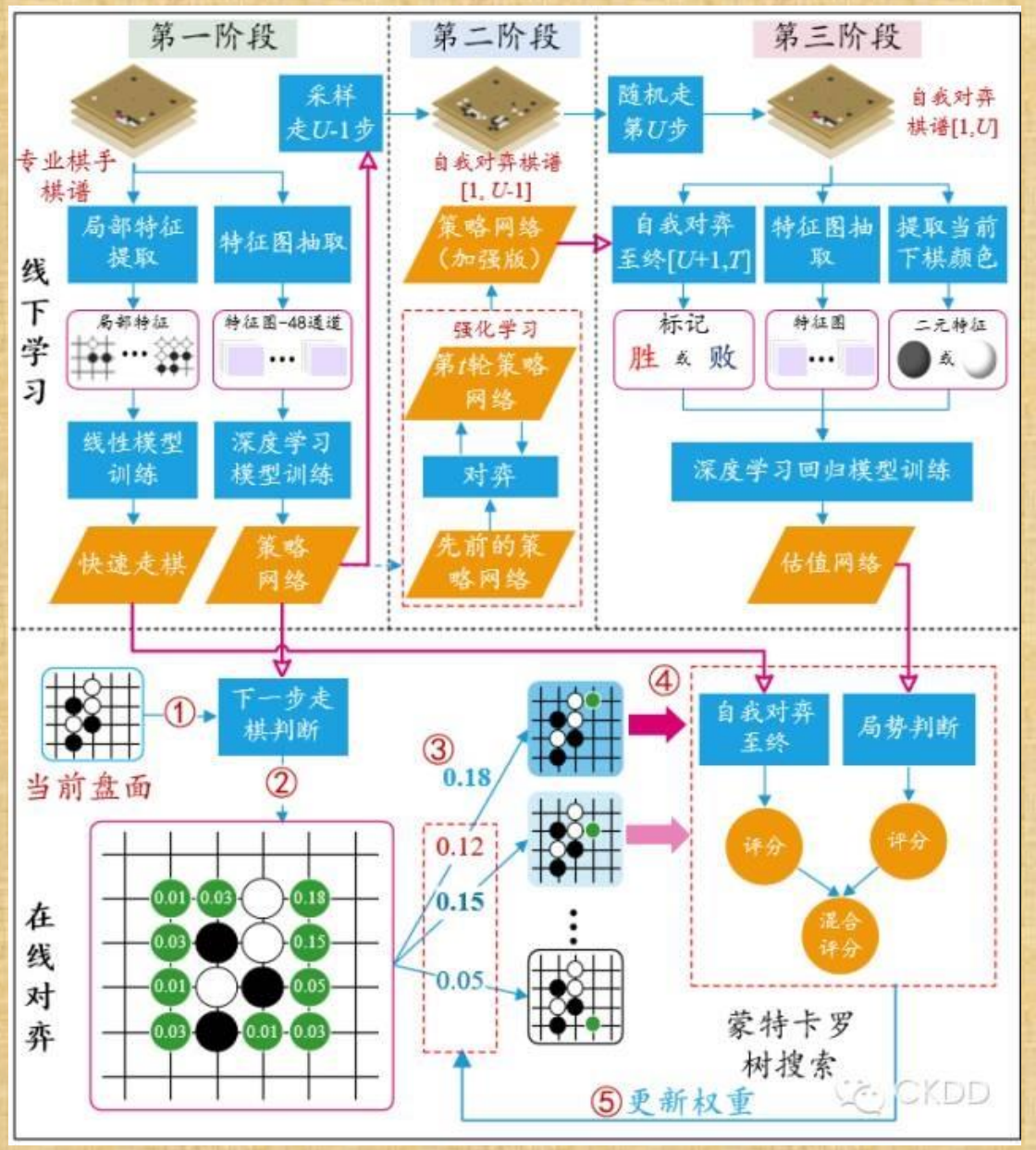
选择蒙特卡洛搜索树中遍历次数最多的点



红圈代表AlphaGO选择的落子点，白色方块表示樊麾的落子点，数字序列表示AlphaGO预测的最可能的变化。

AlphaGO的实力


Elo Rating
3,500
3,000
2,500
2,000
1,500
1,000
500
0
9p
7p
5p
3p
1p
9d
7d
5d
3d
1d
1k
3k
5k
7k
Professional dan (p)
Amateur dan (d)
Beginner kyu (k)
AlphaGo distributed
AlphaGo
Fan Hui
Crazy Stone
Zen
Pachi
Fuego
GnuGo
Rollouts
Value network
Policy network
Threads 1 2 4 8 16 32 40 40 12 24 40 64
GPUs 8 1 2 4 8 64 112 176 280
Single machine
Distributed



http://blog.csdn.net/xuxiatian/article/details/53155576

- 优点：

作者开发出了之前围棋算法中缺少但却极其重要的价值网络模型，并结合策略网络、快速下棋策略糅合到蒙特卡洛树搜索上，使计算机的围棋棋力大增。

- 不足：

文章虽然每个部分讲的很详细，但没能总体的说明AlphaGO的下棋机制，以及对图片的解释比较简略，整篇文章的逻辑不是很连贯，而且这种模型是否适用于其他领域文章也没有提及。

- 启发：

随着计算机和算法的发展，在面对NP难问题上已经不在需要穷举来求解，深度学习可以在很多方面将我们的经验充分利用来解决问题。