

## 译 AlphaGo论文的译文，用深度神经网络和树搜索征服围棋：Mastering the game of Go with de neural networks and tree search

### 前言：

围棋的英文是 the game of Go，标题翻译为：《用深度神经网络和树搜索征服围棋》。

在AlphaGo的影响之下，全社会对人工智能的关注进一步提升。3月12日，AlphaGo 第三次击败李世石。在3月15日总比分定格为4：1，随后AlphaGo 世界来到第二。

论文的英文原文[点击这里](#)拜读

编者按：2014年5月，人们认为至少需要十年电脑才能击败职业选手。笔者在翻译的时候忠实于原文，很少加入自己的理解（本人不敢说有啥深入理解 翻译结果可能不好。但是对于本人而言，翻译这篇文论的过程大于结果：一篇一万字的中文翻译，背后是十万中英文资料的阅读。

### 译文

标题：用深度神经网络和树搜索征服围棋

作者：David Silver 1, Aja Huang 1, Chris J. Maddison 1, Arthur Guez 1, Laurent Sifre 1, George van den Driessche 1, Julian Schrittwieser Antonoglou 1, Veda Panneershelvam 1, Marc Lanctot 1, Sander Dieleman 1, Dominik Grewe 1, John Nham 2, Nal Kalchbrenner 1, Ilya Sutskever 1, Timothy Lillicrap 1, Madeleine Leach 1, Koray Kavukcuoglu 1, Thore Graepel 1, Demis Hassabis 1

他们来自 Google DeepMind 英国团队（用1表示），Google 总部（用2表示）

David Silver, Aja Huang是并列第一作者

摘要：人们长久以来认为：围棋对于人工智能来说是最具有挑战性的经典博弈游戏，因为它的巨大的搜索空间，评估棋局和评估落子地点的难度。我们引入一种新的方法，这个方法使用估值网络来评估棋局，以及使用策略网络来选择如何落子。这些深度神经网络被一种新的组合来训练：使用了人类的监督学习，以及自我对弈的强化学习。没有使用任何预测搜索的方法，神经网络下围棋达到了最先进的蒙特卡洛树搜索程序的水准，这程序模拟了数对弈的随机博弈。我们同时也引入了一种新的搜索算法，这算法把蒙特卡洛模拟和估值、策略网络结合在一起。运用了这个搜索算法，我们的程序AlphaGo在它围棋程序的对弈中达到了99.8%的胜率，并且以5：0的比分击败了欧洲冠军，这是史上第一次计算机程序在全尺寸围棋中击败一个人类职业棋手。在人们认为需要至少十年才会达成这个壮举。

### 引言

所有完全信息博弈都有一个最优估值函数 $v^*(s)$ ，它在判断了每个棋局或状态 $s$ 之后的博弈结果的优劣（在所有对手完美发挥的情况下）。解决这些博弈搜索树中递归调用最优估值函数，这个搜索树包含大约 $b^d$ 种可能的下棋序列，其中 $b$ 是博弈的广度（每一次下棋时候的合法落子个数）， $d$ 是深度（长度）。在大型博弈中，比如国际象棋（ $b \approx 35, d \approx 80$ ），和特别是围棋（ $b \approx 250, d \approx 150$ ），穷举搜索是不可行的，但是有效的搜索空间可通用的原则减少。第一，搜索的深度可以通过棋局评估降低：在状态 $s$ 时对搜索树进行剪枝，然后用一个近似估值函数 $v(s) \approx v^*(s)$ 取代状态 $s$ 下面的近似估值函数预测状态 $s$ 之后的对弈结果。这种方法已经在国际象棋，国际跳棋，黑白棋中得到了超越人类的下棋能力，但是人们认为这种方法在围棋中是不行的，因为围棋的巨大的复杂度。第二，搜索的广度可以通过来自策略 $p(a | s)$ 的采样动作来降低，这个策略是一个在位置 $s$ 的可能下棋走子 $a$ 概率分布。蒙特卡洛走子方法搜索到最大深度时候根本不使用分歧界定法，它从一个策略 $p$ 中采集双方棋手的一系列下棋走法。计算这些走子的平均数可以产生一个评估，在西洋双陆棋戏和拼字游戏中获得了超出人类的性能表现，并且在围棋中达到了业余低段水平。

蒙特卡洛树搜索使用蒙特卡洛走子方法，评估搜索树中每一个状态的估值。随着执行越来越多的模拟，这个搜索树成长越来越大，而且相关估值愈发精确。下棋动作的策略在搜索的过程中也会随着时间的推移而改进，通过选择拥有更高估值的子树。渐近的，这个策略收敛到一个最优下法，然后评估收敛到数。目前最强的围棋程序是基于蒙特卡洛树搜索的，并且受到了策略的增强，这个策略被人训练用来预测专家棋手的下法。这些策略用来缩窄搜索空间。下棋动作，和用来在走子中采集下法动作。这个方法已经达到了业余高手的级别。然而，先前的工作已经受到了肤浅策略的限制或基于输入的线性组合的限制。

最近，深度卷积神经网络已经在计算机视觉中达到了空前的性能：比如图像分类，人脸识别，和玩雅达利的游戏。它们使用很多层的神经网络，层与层叠排列在一起，用来构建图片的愈发抽象的局部代表。我们为围棋程序部署了类似的体系架构。我们给程序传入了一个19\*19大小棋局的图片，然后使用网络来构建一个位置的表征。我们使用这些神经网络来降低搜索树的有效深度和广度：通过估值网络来评估棋局，和使用策略网络来博弈取样。

我们使用一个包含多个不同阶段的机器学习方法的管道来训练神经网络。我们开始使用一个监督学习（SL）策略网络 $p$ ，它直接来自人类专家的下棋。快速高效的学习更新，拥有快速的反馈和高质量的梯度。和向前的工作类似，我们同时也训练了一个可以迅速从走子中取样的快速策略 $p$ 。其次，我们训练了一个强化学习（RL）策略网络 $p_p$ ，它通过优化自我对弈的最终结局来提升SL策略网络。这调整策略网络朝向赢棋的正确目标发展，而不是最大化提高预测。我们训练了一个估值网络 $v$ ，它预测博弈的赢者，通过和RL策略网络和自己对弈。我们的AlphaGo程序有效的把策略网络、估值网络，和蒙特卡洛搜索结合起来。

1 策略网络的监督学习

在训练管道的第一阶段，我们在先前工作的基础上，使用了监督学习来预测人类专家下围棋。监督学习（SL）策略网络 $p(a|s)$ 在重量 $\bar{\sigma}$ 的卷积层和非器中替换。策略网络的输入 $s$ 是一个棋局状态的简单代表（如扩展数据表2）。策略网络使用了随机取样状态-动作对 $(s, a)$ ，使用了随机梯度递增来在状态 $s$ 选择下棋走子 $a$ 的可能性。

$$\Delta \sigma \propto \frac{\partial \log p_{\sigma}(a|s)}{\partial \sigma} \tag{1}$$

我们用KGS围棋服务器的3千万个棋局，训练了13层的策略网络（我们称之为SL策略网络）。在输入留存测试数据的所受特征的时候，这个网络预测人的精准的达到了57%，而且在仅仅使用原始棋局和下棋记录的时候，精度达到了55.7%。与之相比，截至到本篇文论提交（2015年），其他研究团队的度是44.4%（全部结果在扩展数据表3）。在精确度方面的小提升会引起下棋能力的很大提升（图片2，a）；更大的神经网络拥有更高的精确度，但是评估速度更慢。我们也训练了一个更快的但是精确度更低的走子策略 $p(a|s)$ ，它使用了一个权重为0.1的小型模式特征的线性softmax。它达到了24.2度，每选择下一步棋只用2微秒，与之相比，策略网络需要3毫秒。

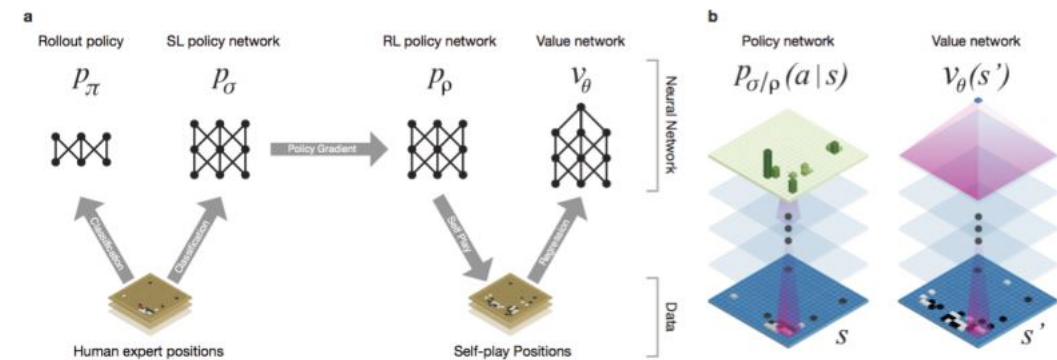


图1：神经网络训练管道和体系结构。a：在一个棋局数据集中，训练一个快速走子策略 $p$ 和监督学习（SL）策略网络 $p$ 用来预测人类专家下棋。一个（RL）策略网络 $p$ 由SL策略网络初始化，然后由策略梯度学习进行提高。和先前版本的策略网络相比，最大化结局（比如赢更多的博弈）。一个新的了，通过自我对弈结合RL策略网络。最终通过回归训练，产生一个估值网络 $v$ ，用来在自我对弈的数据集中预测期待的结局（比如当前棋手是否能赢）。AlphaGo使用的神经网络体系架构的原理图代表。策略网络把棋局状态 $s$ 当作输入的代表，策略网络把 $s$ 传输通过很多卷积层（这些卷积层是参数为 $\bar{\sigma}$ 的卷积层或者参数为 $\bar{\sigma}$ 的RL策略网络），然后输出一个关于下棋动作 $a$ 的概率分布 $p(a|s)$ 或 $p(a|s)$ ，用一个棋盘的概率地图来表示。估值网络类似的参数 $\bar{\sigma}$ 的卷积层，但是输出一个标量值 $v(s')$ 用来预测棋局状态 $s'$ 后的结局。

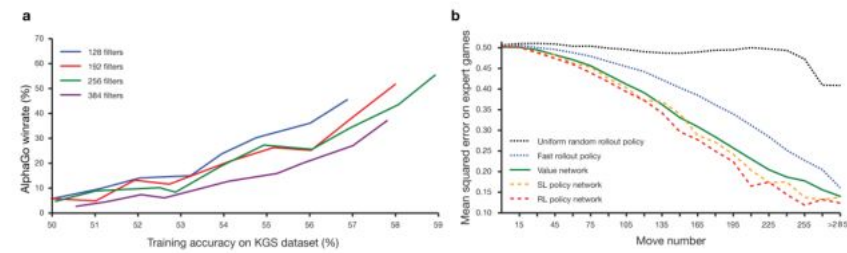


图2：策略网络和估值网络的能力和精确度。a图显示了策略网络的下棋能力随着它们的训练精确度的函数。拥有128，192，256，384卷积过滤每层的训练过程中得到周期性的评估；这个图显示了AlphaGo使用不同策略网络的赢棋概率随着的不同精确度版本的AlphaGo的变化。b：估值网络 and 不同策略对比。棋局和结局是从人类专家博弈对局中采样的。每一个棋局都是由一个单独的向前传递的估值网络 $v$ 评估的，或者100个走子的平均值，这些走子机走子，或快速走子策略 $p$ ，或SL策略网络 $p$ ，或RL策略网络 $p$ 。图中，预测估值和博弈实际结局之间的平均方差随着博弈的进行阶段（博弈总步）的变化而变化。

2 策略网络的强化学习

训练管道第二阶段的目标是通过策略梯度强化学习（RL）来提高策略网络。强化学习策略网络 $p$ 在结构上和SL策略网络是一样的，权重 $\bar{\sigma}$ 初始值也是 $\bar{\sigma}$ 。我们在当前的策略网络和随机选择某先前一次迭代的策略网络之间博弈。从一个对手的候选池中随机选择，可以稳定训练过程，防止过度拟合策略。我们使用一个奖励函数 $r(s)$ ，对于所有非终端的步骤 $t < T$ ，它的值等于零。从当前棋手在步骤 $t$ 的角度来讲，结果 $z_t = \pm r(s_T)$ 是在博弈结束时奖励，如果赢棋，结果等于+1，如果输棋，结果等于-1。然后权重在每一个步骤 $t$ 更新：朝向最大化预期结果的方向随机梯度递增

$$\Delta \rho \propto \frac{\partial \log p_{\rho}(a_t | s_t)}{\partial \rho} z_t. \quad (2)$$

我们在博弈过程中评估 RL策略网络的性能表现，从输出的下棋动作的概率分布，对每一下棋动作 $a_t \sim p_p(\cdot | s_t)$ 进行取样。我们自己面对面博弈，RL SL策略网络的胜率高于80%。我们也测试了和最强的开源围棋软件 Pachi 对弈，它是一个随机的蒙特卡洛搜索程序，在KGS中达到业余2段。在没有使情况下，RL策略网络对 Pachi的胜率达到了85%。与之相比，之前的最先进的仅仅基于监督学习的卷积网络，对 Pachi的胜率仅只有11%，对稍弱的程胜率是12%。

### 3 估值网络的强化学习

训练管道的最后一个阶段关注于棋局评估，评估一个估值函数  $v^p(s)$ ，它预测从棋局状态  $s$  开始，博弈双方都按照策略网络  $p$  下棋的结局，

$$v^p(s) = \mathbb{E}[z_t | s_t = s, a_{t...T} \sim p]. \quad (3)$$

理想情况下，我们期望知道在完美下法 $v^*(s)$ 情况下的最优值；然而在现实中，我们使用 RL策略网络，来评估估值函数 $v^p$ ，作为我们的最佳策略。我们<sup>2</sup> 的估值网络  $v(s)$ 来逼近估值函数， $v(s) \approx v^p \approx v^*(s)$ 。这个神经网络和策略网络拥有近似的体系结构，但是输出一个单一的预测，而不是一个概率。我们通过回归到状态-结果对  $(s, z)$  来训练估值网络的权重，使用了随机梯度递减，最小化预测估值 $v(s)$ 和相应的结局  $z$  之间的平均方差 (MSE)。

$$\Delta \theta \propto \frac{\partial v_{\theta}(s)}{\partial \theta} (z - v_{\theta}(s)). \quad (4)$$

这个天真的从拥有完整对弈的数据来预测博弈结局的方法导致过度拟合。问题在于，连续的棋局之间的联系十分强大，和仅单独下一步棋有差距，但是个博弈又是相通的。当通过这种方式在KGS数据集上训练是，估值网络记住了博弈的结局而不是推广出新的棋局，在测试数据上面MSE最小达到了0.19。比在训练数据集上面MSE是0.19。为了解决这个问题，我们想出了新的自我对弈的数据集合，包含了三千万个不同的棋局，每一个都是从不同盘博弈一盘博弈都是在 RL策略网络和自己之间对弈，直到博弈本身结束。在这个数据集上训练导致了MSE为0.226，和训练和测试数据集的MSE为0.234小的过度拟合。图2，b展示了估值网络对棋局评估的精确度：对比使用了快速走子策略网络 $p$  的蒙特卡洛走子的精确度，估值函数一直更加精确。而 $v(s)$ 的精确度也逼近了使用了 RL策略网络 $v(s)$ 的蒙特卡洛走子的精确度，不过计算量是原来的15000分之一。

### 4 运用策略网络和估值网络搜索

AlphaGo在把策略网络、估值网络和MCTS算法结合，MCTS通过预测搜索选择下棋动作。每一个搜索树的边  $(s, a)$  存储着一个动作估值  $Q(s, a)$ ，访问次数  $N(s, a)$ ，和先验概率  $P(s, a)$ 。这棵树从根节点开始，通过模拟来遍历（比如在完整的博弈中沿着树无没有备份地向下搜索）。在每一次模拟的时间步骤  $t$ ，选择下一个棋动作 $a_t$ ，

$$a_t = \operatorname{argmax}_a (Q(s_t, a) + u(s_t, a)), \quad (5)$$

用来最大化动作估值加上一个额外奖励 $u(s, a) \sim \frac{P(s,a)}{1+N(s,a)}$ ，它和先验概率成正向关系，但是和重复访问次数成反向关系，这样是为了鼓励更多的探索。遍历到达一个叶节点 $s_L$ 时，该叶节点可能不会被扩展。叶节点棋局 $s_L$ 仅被 SL策略网络 $p$  执行一次。输出的概率存储下来作为每一合法下法动作  $a$  的先验概率  $P(s, a) = p(a | s)$ 。叶节点通过两种方式的得到评估：第一，通过价值网络 $v(s_L)$ 评估；第二，用快速走子策略 $p$  随机走子，直到终点步骤  $T$ ，产生 $z_L$ 作为评估方法。这些评估方法结合在一起，在叶节点的评估函数 $V(s_L)$ 中使用一个混合参数  $\lambda$ ，

$$V(s_L) = (1 - \lambda)v_{\theta}(s_L) + \lambda z_L. \quad (6)$$

在模拟的结尾  $n$ ，更新所有被遍历过的边的下棋动作估值和访问次数。每一条边累加访问次数，和求出所有经过该边的模拟估值的平均值。

$$N(s, a) = \sum_{i=1}^n \mathbf{1}(s, a, i) \quad (7)$$

$$Q(s, a) = \frac{1}{N(s, a)} \sum_{i=1}^n \mathbf{1}(s, a, i) V(s_L^i), \quad (8)$$

其中 $s_L^i$ 是第  $i$  次模拟的叶节点， $\mathbf{1}(s, a, i)$ 代表一条边  $(s, a)$  在第  $i$  次模拟时是否被遍历过。一旦搜索完成，算法选择从根节点开始，被访问次数最多的边。

在AlphaGo中，SL策略网络 $p$  的表现优于 RL策略网络 $p$ ，推测可能是因为人类从一束不同的前景很好的下棋走法中选择，然而 RL优化单一最优下棋而，从更强的 RL策略网络训练出来的估值函数 $v \approx v^P(s)$  优于从 SL策略网络训练出来的估值函数 $v \approx v^P(s)$

评估策略网络和估值网络和传统的启发式搜索相比，需要多几个数量级的计算量。为了高效的把 MCTS 和深度神经网络结合在一起，AlphaGo 在很多步多线程搜索技术进行了模拟，在很多GPU上计算策略网络和估值网络。最终版本的 AlphaGo使用了40个搜索线程，48个CPU，和8个GPU。我们也分布式的AlphaGo版本，它利用的多台电脑，40个搜索线程，1202个CPU，176个GPU。在方法部分提供了关于异步和分布MCTS的全部的细节。

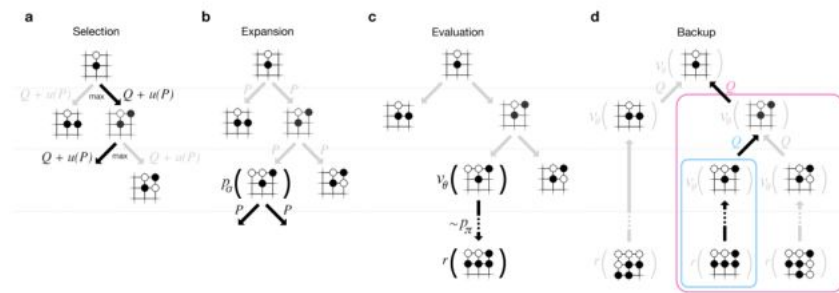


图3：AlphaGo中的蒙特卡洛树搜索。**a** 每一次模拟遍历搜索树，通过选择拥有最大下棋动作估值  $Q$  的边，加上一个额外奖励  $u(P)$ （依赖于存储的该边  $P$ ）。**b** 叶节点可能被展开，新的结点被策略网络 $p$  执行一次，然后结果概率存储下来作为每一个下棋动作的先验概率。**c** 在一次模拟的结尾，叶节点 $E$  估计：使用估值网络  $v$  和运行一个走子策略直到博弈结尾（使用了快速走子策略 $p$ ），然后对于赢者运算函数  $r$ 。**d** 下棋动作估值  $Q$  得到更新，用来跟  $r(\cdot)$  的平均值和该下棋动作的子树的  $v(\cdot)$

## 5 评估AlphaGo的下棋能力

为了评估 AlphaGo 的水平，我们举办了内部比赛，成员包括不同版本的 AlphaGo 和几个其它的围棋程序，包括最强的商业程序 CrazyStone和Zen，程序Pachi和Fuego。所有这些程序都是基于高性能蒙特卡洛树搜索算法的。此外，我们的内部比赛还包括了开源程序GnuGo，它使用了最先进搜索方树搜索。所有程序每次执行下一步棋允许5秒钟。

比赛的结果如图4,a，它预示着单击版本的AlphaGo比先前任何一个围棋程序强上很多段，在495场比赛中，AlphaGo赢了其中的494场比赛。我们也在的情况下进行了比赛：AlphaGo和CrazyStone，Zen，Pachi的胜率分别是77%，86%，99%。分布式版本的AlphaGo强大很多：和单机版本的Alpha率是77%，和其他的围棋程序对弈的胜率是100%。

我们也评估了不同版本的AlphaGo，不同版本仅仅使用估值网络（ $\gamma = 0$ ）或者仅仅使用快速走子（ $\gamma = 1$ ）（如图4,b）。即使不使用快速走子，AI能超出了所有其他围棋程序，显示了估值网络提供了一个可行的代替蒙特卡洛评估的可能。不过，其估值网络和快速走子的混合版本表现最好（ $\gamma = 0$ ）。他版本的AlphaGo的时候胜率达到了95%以上。这预示着这两种棋局评估系统是互补的：估值网络通过能力很强，但是不切实际的慢的 $p$ 来逼近博弈走子可以通过能力更弱但是更快的快速走子策略 $p$ 来精确的评估博弈的结局。图5将AlphaGo在真实博弈棋局中评估能力可视化了。

最终，我们把分布式版本的AlphaGo和樊麾进行了评估，他作为一个职业2段棋手，是2013，2014，2015年的欧洲围棋冠军。在2015年10月5-9日，樊麾在真实比赛中下了5盘棋。AlphaGo以5：0的比分赢了比赛（图6和扩展数据表1）。这是史上第一次，在人类不让子和完整棋盘的情况下，一个围了一个人类职业棋手。这个壮举之前认为需要至少十年才能达到。

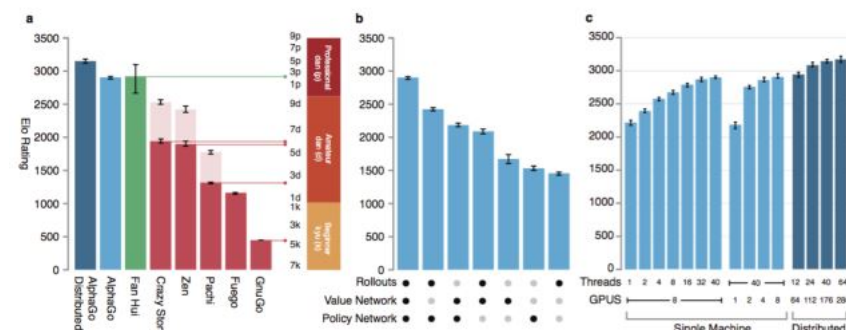


图4：AlphaGo的比赛评估。**a** 和不同围棋程序比赛的结果（见扩展数据表6-11）。每个程序使用接近5秒每走一步棋的速度。为了给AlphaGo更高的排名，有些程序得到了所有对手让4步子的优势。程序的评估基于ELO体系：230分的差距，这相当于79%的胜率差距，这大致相当于在KGS中高一个业余等级。接近的相当也显示了，水平的线显示了程序在在线比赛中达到的KSG等级。和欧洲冠军樊麾的比赛也包括在内。这些比赛使用更长的时间控制。图中显示置信区间。**b** 单机版本的AlphaGo在组成部分的不同组合下的性能表现。其中仅仅使用了策略网络的版本没有使用任何搜索算法。**c** 蒙特卡洛搜索树算法和GPU的可扩展性研究，其中使用了异步搜索（浅蓝色）和分布式搜索（深蓝色），每下一步时间两秒。



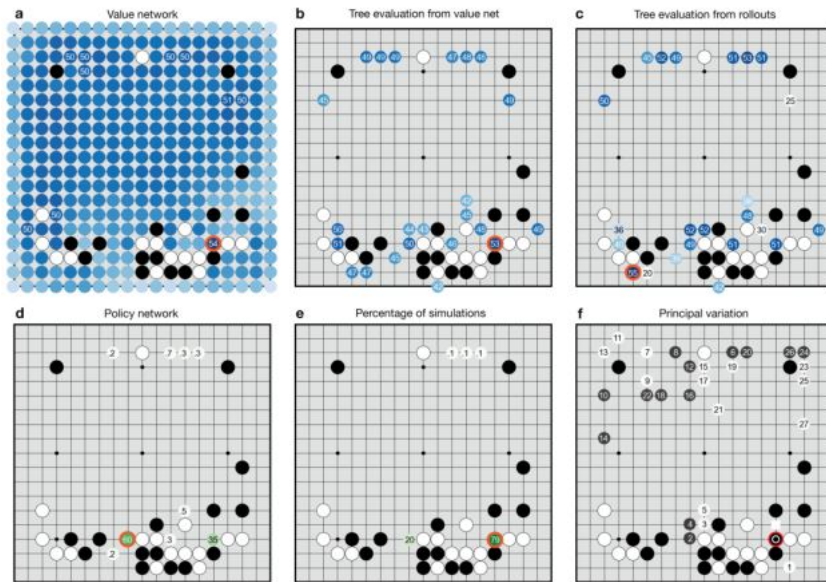


图5：AlphaGo（执黑）是在一个和樊麾的非正式的比赛中选择下棋走子的。接下来的每一个统计中，估值最大的落子地点用橘黄色标记。**a**根节点  $s$  的点  $s'$  的估值，使用估值网络  $v(s')$ ，评估很靠前的会赢的百分数显示出来了。**b** 从根节点开始的每一条边  $(s, a)$  的走子动作估值  $Q(s, a)$ ；仅仅使用估值  $v = 0$  方法的均值。**c** 下棋动作  $Q(s, a)$ ，仅仅使用快速走子 ( $v = 1$ ) 方法的均值。**d** 直接使用 SL 策略网络的下棋走子概率， $p(a|s)$ ；如果大于 0.1 百分比的形式报告出来。**e** 从根节点开始的模拟过程中下棋走子地点选择的频率百分比。**f** AlphaGo 的树搜索的理论上的走子选择序列（一个搜索过程中多的路径）。下棋走子用一个数字序列表示。AlphaGo 选择下棋的落子地点用红色圆圈标记出来；樊麾下在白色方形的地方作为回应；在他的复盘过过程：下在地点 1 应该是更好的选择，而这个落子地点正好是 AlphaGo 预测的白棋的落子地点。

## 6 讨论

在这个工作中，我们基于一个深度神经网络和树搜索的结合开发了一个围棋程序，它的下棋水平达到了人类最强的水平，因此成功战胜了一项人工智能战。我们首次，对围棋开发了一个有效的下棋走子选择器和棋局评估函数，它是基于被一个创新型的监督学习和强化学习的组合训练的深度神经网络。的搜索算法，它成功的把神经网络评估和蒙特卡洛走子结合在一起。我们的程序 AlphaGo 把这些组成部分按照比例集成在一起，成为了一个高性能的树

在和樊麾的比赛中，AlphaGo 对棋局评估的次数和深蓝对卡斯帕罗夫下国际象棋的时候的次数相比，是其千分之一。作为补偿的，是更加智能的棋局选用了更加精确的评估棋局的能力，使用了估值网络（一个也许是更加接近于人类下棋方式的方法）。此外，深蓝使用的是人类手工调参数的估值函数，AlphaGo 的神经网络是直接比赛对弈数据中训练出来的，纯通过一个通用目的的监督学习和强化学习方法。

围棋在很多方面是横亘在人工智能面前的困难：一个有挑战性的决策任务；一个难以对付的解空间；和一个非常复杂的最优解，以至于它看上去不可能或者估值函数逼近。之前的关于围棋程序的重大突破，蒙特卡洛树搜索，在其它领域导致了相应的进步：比如通用的博弈比赛，经典的规划问题，局部问题，调度问题，和约束满足问题。通过把树搜索和策略网络、估值网络结合在一起，AlphaGo 最终达到了围棋职业选手的水平，并且提供了希望：在解决的人工智能领域里，计算机现在是可以达到人类水平的。

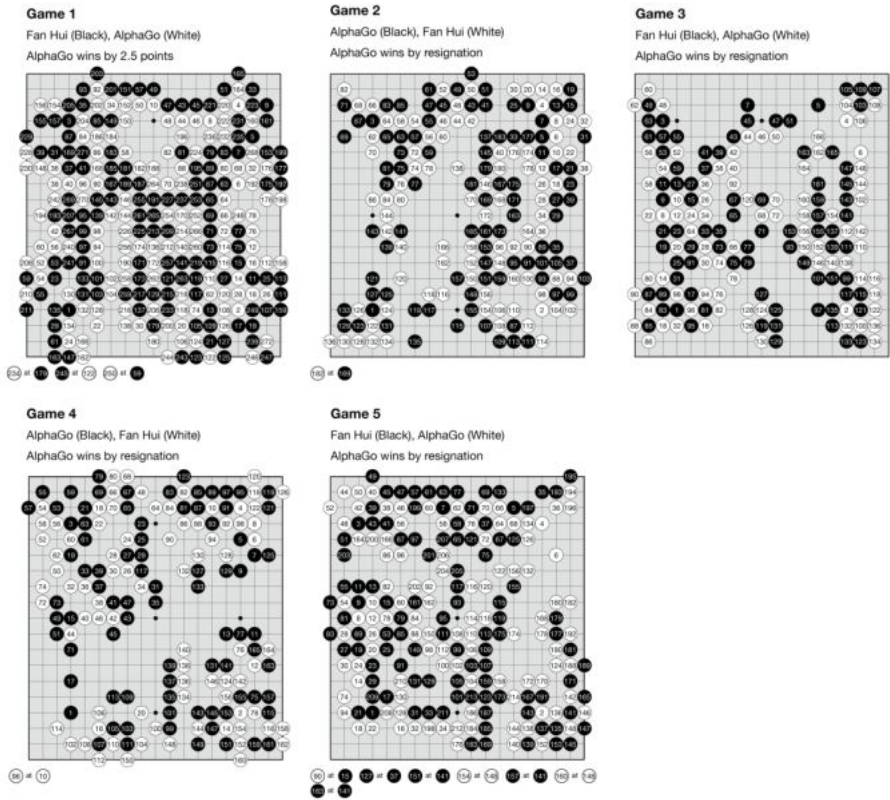


图6：AlphaGo 和欧洲冠军樊麾的博弈棋局。下棋走的每一步按照下棋顺序由数字序列显示出来。重复落子的地方在棋盘的下面成双成对显示出来。每一个数字的落子，重复到了第二个数字显示的交叉地方。

参考文献  
略

其他贡献者  
略

感谢  
我们感谢樊麾答应和AlphaGo进行比赛；感谢T.M担当比赛的裁判；感谢R.M和T.S给予有帮助的讨论和建议；感谢A.C和M.C在可视化方面的工作；感谢D.K, D.P, H.vH, A.G和G.O修订了这篇论文；感谢 DeepMing 团队其它的成员的支持，想法，和鼓励。

## 后记

理解 AlphaGo 有两个关键部分：

- 1. 深度神经网络的训练过程，文章把这个过程描述成为一个**管道**。所谓管道，很像Linux系统中的管道命令，把前者的输出作为后者的输入
- 2. 蒙特卡洛树搜索的过程，看这个搜索是如何把 SL策略网络，估值网络，快速走子策略结合在一起的。

先谈谈 1：  
输入人类的棋谱，经过监督学习，输出SL策略网络  
输入SL策略网络，经过强化学习，输出RL策略网络  
输入RL策略网络，经过强化学习，输出估值网络

再谈谈 2：  
蒙特卡洛树搜索是模拟下棋，并且评估的过程。蒙特卡洛就是“随机”的意思，只不过逼格更高而已。如果你“随机”下棋，肯定输呀，怎么办？使用**来预测人类是如何下棋的**。AlphaGo每次要下棋的时候，先运行 SL策略网络一遍，得到一个概率分布，在此基础上进行“随机”：更有可能在概率更大的子。  
AlphaGo一边模拟自己下棋，一边模拟对手下棋，最后，下完了。所谓下完了，就是在树搜索的时候达到了叶节点。下完了之后，对棋局进行评估。**结和快速走子策略，得到一个估值函数**，该函数的值越高，越好。  
**模拟很多很多遍**。模拟结束之后，进行“统计”工作。统计每一条边走过的次数，和估值函数的估值。最后，AlphaGo做出选择：现在是棋局 s，如果结合 a 在模拟过程中走过的次数，以及 a 下面的叶节点的估值函数，累加起来最高，那么AlphaGo选择在 a 地方落子。

