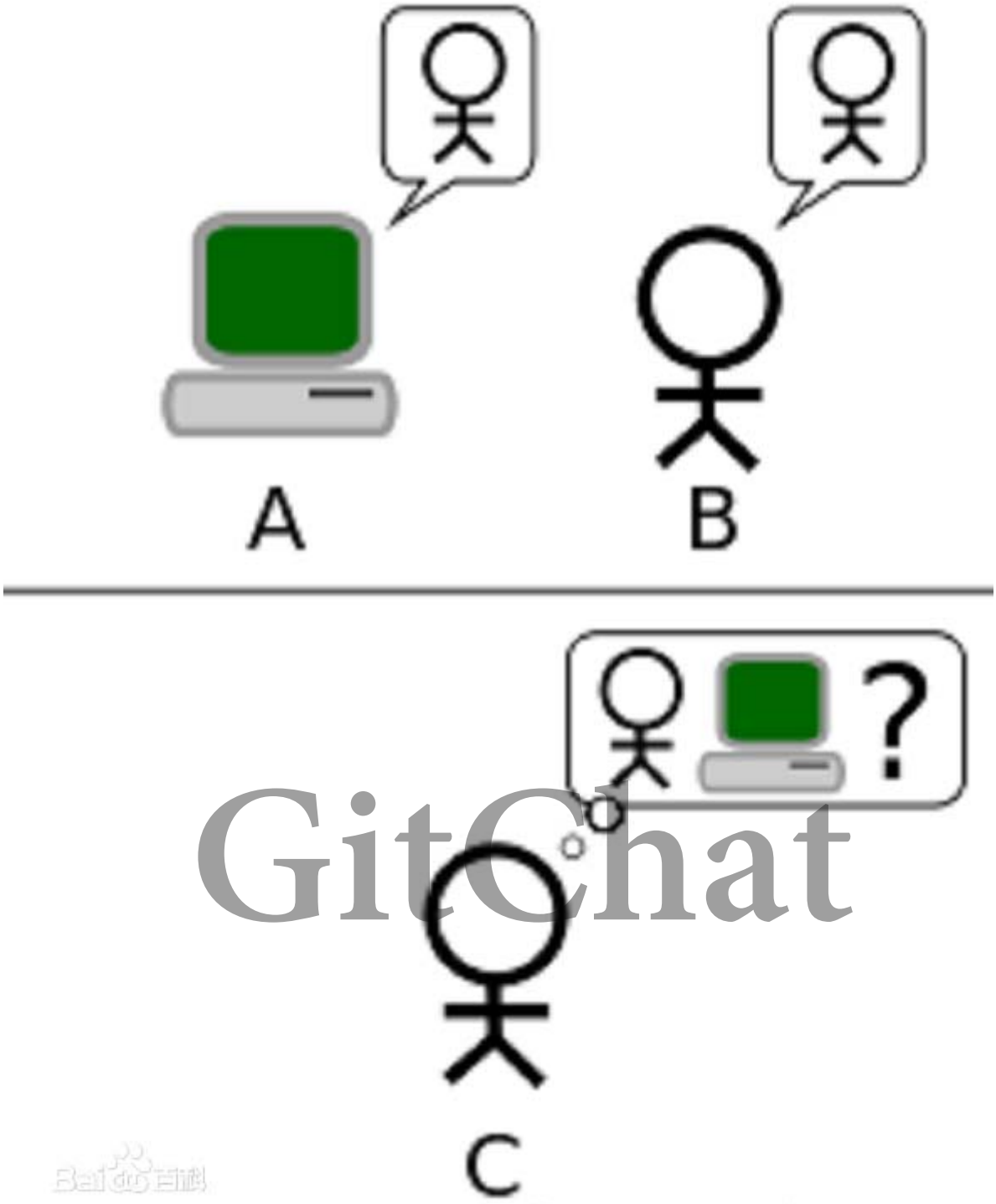
个人开发者如何通过人工智能盈利？智能原理及阿尔法狗详解

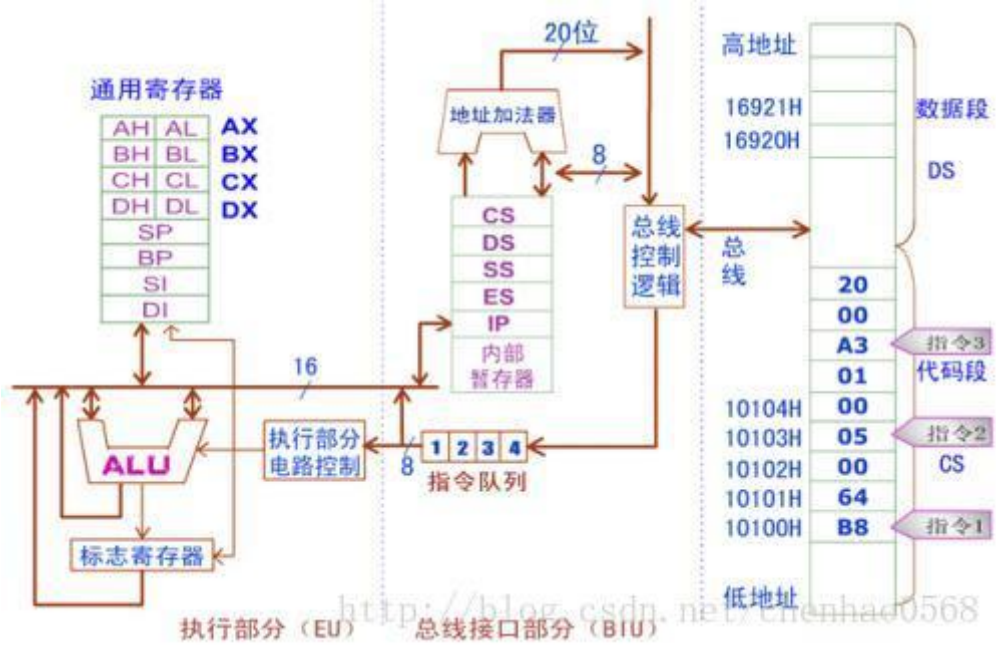
目录：

一、人工智能大背景，历史

1. 智能是什么？从模仿游戏开始

大家知道人是具有智能的，如果做一个机器，让你分辨不出它是人还是机器，就说明这个机器具有了智能（图灵测试）。有兴趣的可以看《模仿游戏》这部电影。



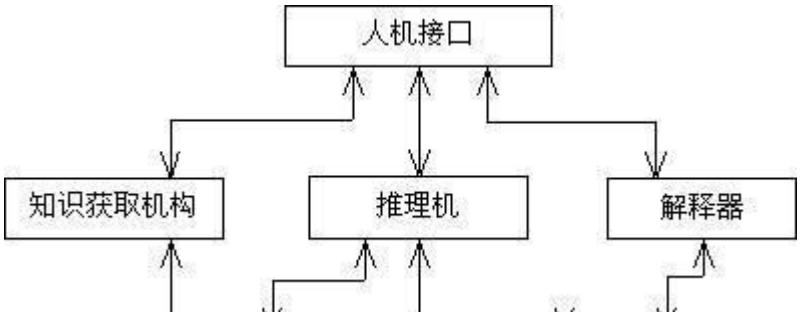


反正计算机很强大了，又快又准，算法越来越高效。机器终于能帮我们干许多活了，只是苦了我们这些码农。

大人说小孩子不能玩水，不能玩火，不能碰电，这也不能做，那也不能做，于是经过漫长的知识经验累积，形成了专家系统。有兴趣的可以看《疯狂原始人》这部电影。

信息越来越多，越来越复杂。信息的输入、维护、分析整理越来越不容易。

在很长的一段时间里，人们依赖逻辑和规则给计算机编程。不同的需求，编写不同的规则（If - Else），数据也通过人来标注好，比如这张图是猫，这张图是狗，反正只要能解决问题，再苦再累也值了。谁让机器不会自动学习呢？

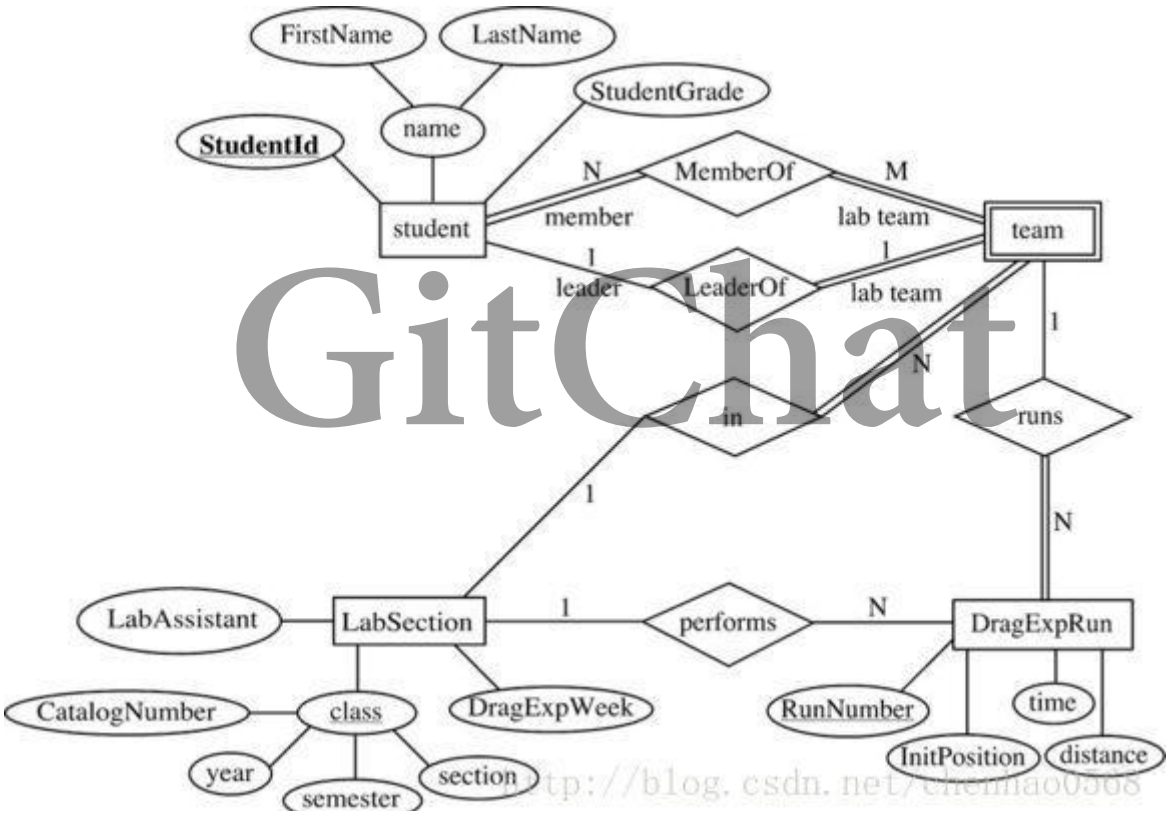


工业界开始关注深度学习，接着图像识别，然后是语音识别，在这些领域，一个一个被深度学习突破。

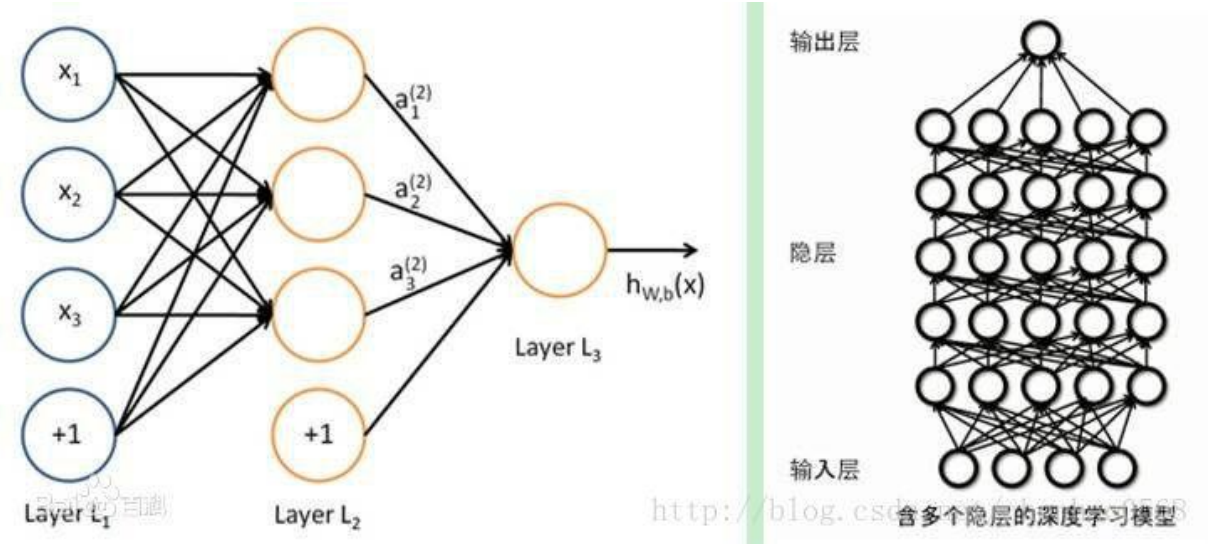
深度学习技术是怎么做的呢？它是通过像搭积木一样地搭建那些神经网络的组合，用数据灌入到网络，接下来，就是见证奇迹的时刻了，他竟然会自己一层层的逐渐寻找出最重要的特征，比人类几十年专家经验设计的特征，效果都要要好得多。这就是深度学习技术技术牛逼的地方。

2. 深度学习（人工神经网络）

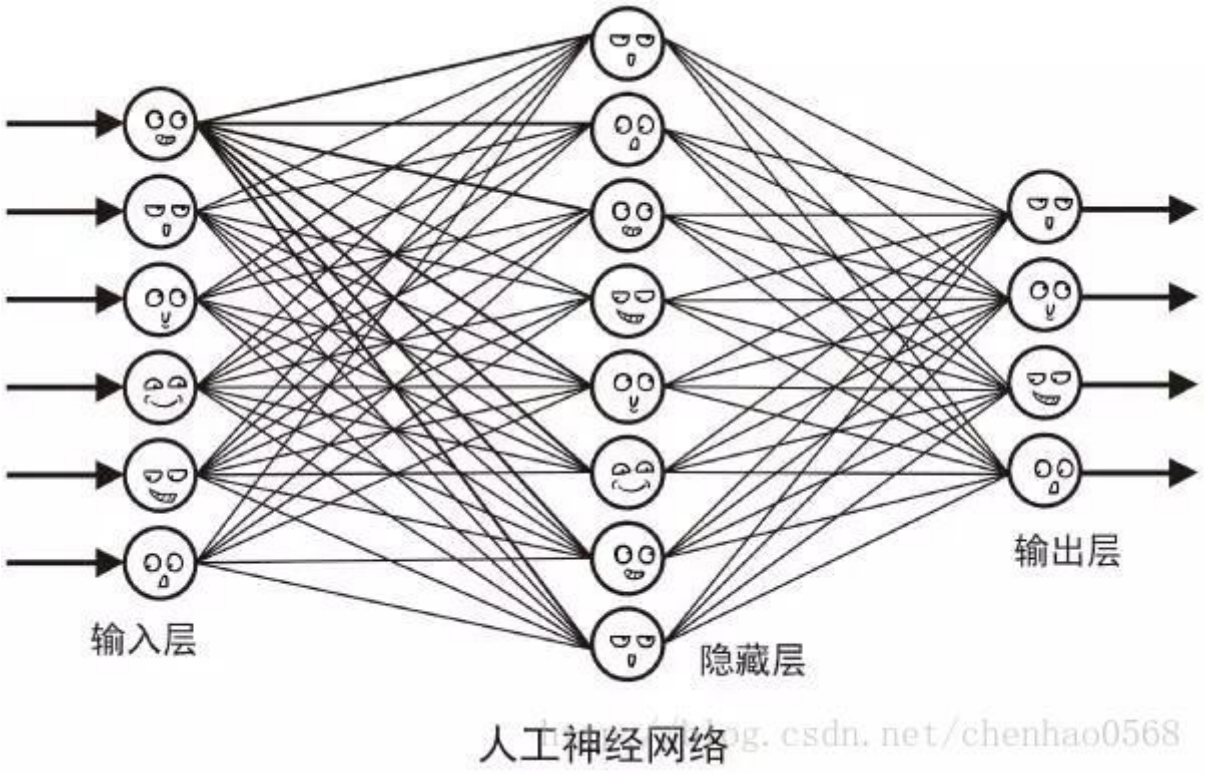
大脑储存信息的方式并非将记忆储存在一个特定的地方，而是在整个神经网络里传播（分布式表征）。举个最简单的例子。一辆 "大黑狗"，如果分布式地表达，一个神经元代表大小，一个神经元代表颜色，第三个神经元代表狗的类别。三个神经元同时激活时，就可以准确描述我们要表达的物体。



络（深度学习）。



现在在围棋中，人工神经网络终于可以自己学习，也可以对其做出反应了，效率与进度让世人吃惊。AlphaGo就是以深度学习技术为基础的电脑程序。



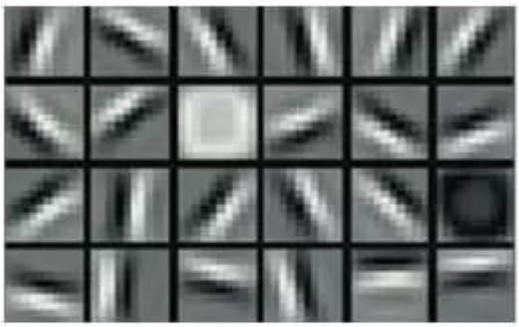
人工神经网络

在人工神经网络中，每一个小圆圈都是在模拟一个"神经元"。它能够接收从上一层神经元传来的输入信号（也就是一堆数字）；根据不同神经元在它眼中的重要性，分配不同的权重，然后将输入信号按照各自的权重加起来（一堆数字乘以权重的大小，再求和）；接着，它将加起来结果代入某个函数（通常是非线性函数），进行运算，得到最终结果；最后，它再将这个结果输出给神经网络中的下一层神经元。

所谓的人工神经网络学习，本质上是让人工神经网络尝试调节每一个神经元上的权重大小，使得整个人工神经网络在某一个任务的测试中的表现达到某个要求（例如，识别汽车的正确率达到90%以上）。

(深度的)人工神经网络如何学习识别汽车

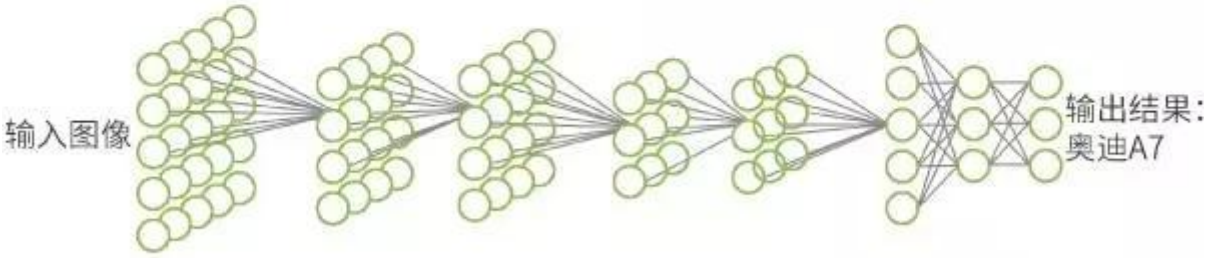
输入图像



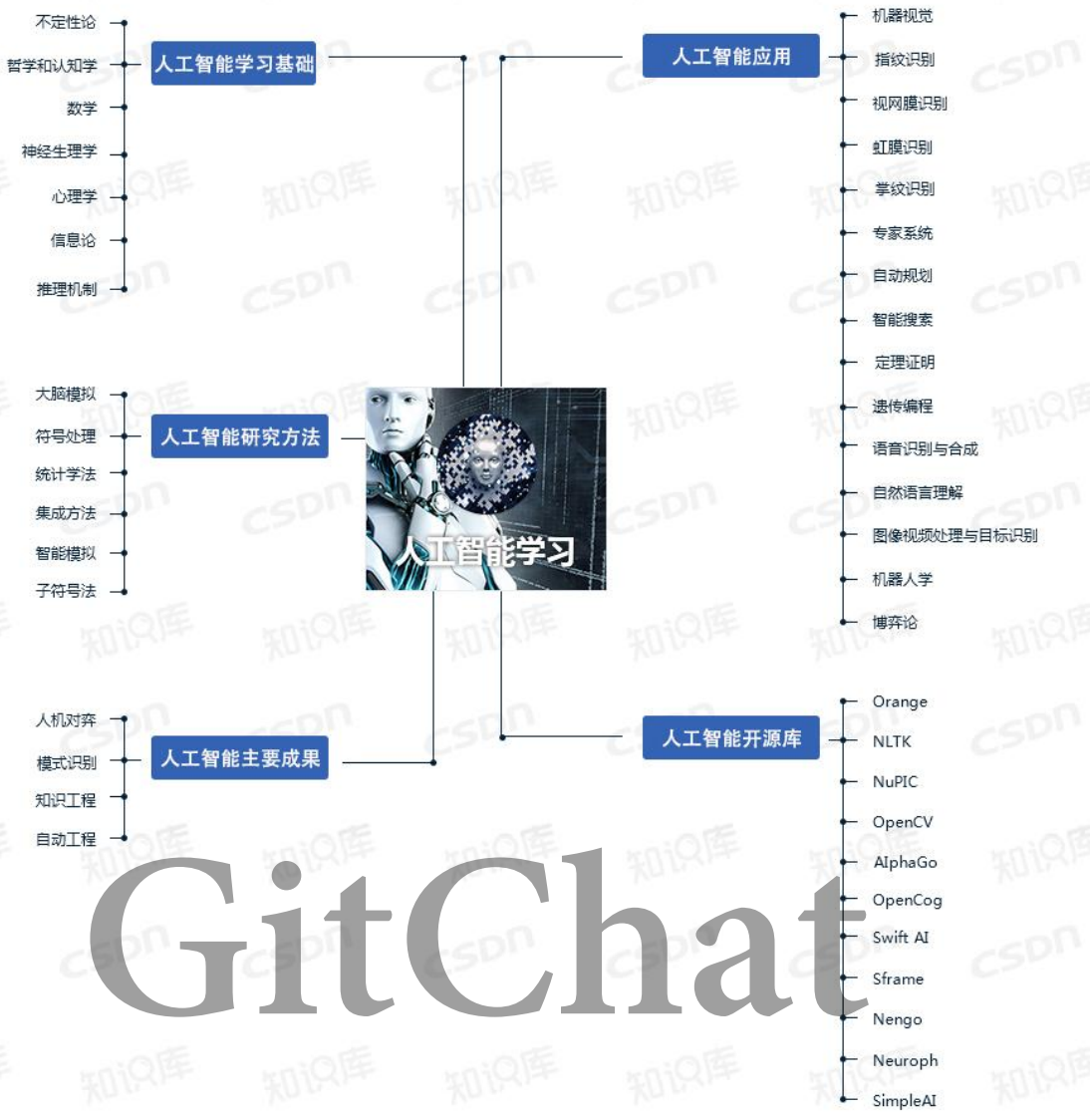
前一层神经元通过运算，识别汽车不同细节的微观特征；

中间层神经元通过运算，识别由汽车微观特征构成的中间特征；

后一层神经元通过运算，识别由汽车中间特征构成的宏观特征。



实际上每一“层”的中间结果只有计算机能够从数学角度理解，这些特征并不是人可以理解的特征。



2. 人工智能学习路线

了解行业最新动态和研究成果，比如各大牛的经典论文、博客等。比如关于AlphaGo介绍，网上就有很多。

三、阿尔法狗原理算法深入解析

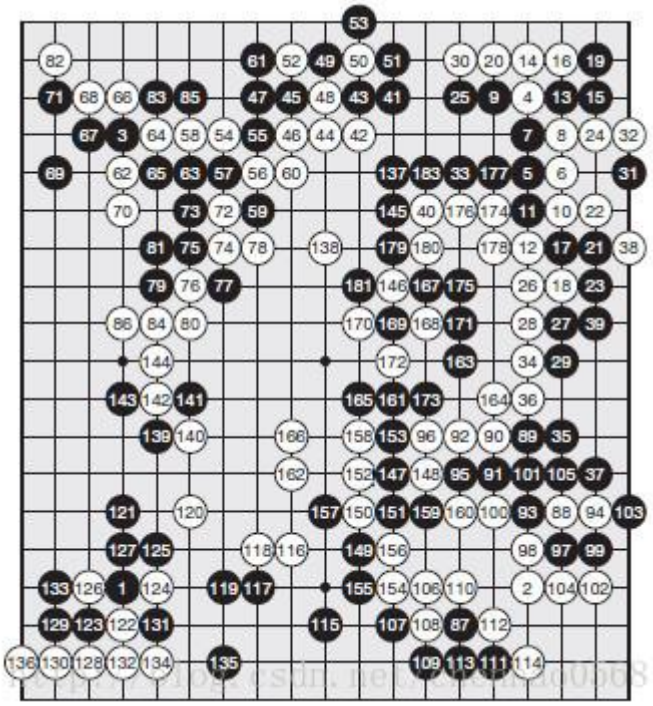
1. 机器学习的第一步先了解业务

围棋的业务特点包括其基本规则、对弈特性和下棋的思路。根据这些业务特点，我们可以分阶段理解AlphaGo算法。

从棋盘状态来看，围棋棋盘是有19*19=361的格子，每个格子有3种可能性（黑、白、空），所以总共有3的361次方个状态；从下棋步骤角度来看，即使不算吃子和打劫，第n步有361-n种选择，所以至少有361!（361的阶乘=361*360*…3*2*1）——超过10的200次方种可能性。，超过目前的计算机的计算能力，无法通过暴力穷举解决。

围棋是所有大家熟知的智力游戏中，搜索空间最大的，所需要的计算量也是最大的。

不像象棋、军棋那样盘面上的棋子越走越少，而是越走越多。所以一局棋从开始到结束，用一张标记好走棋顺序的棋谱就能保存绝大部分下棋的信息，是一个时间序列。如下图就是《Nature》论文中的樊麾与AlphaGo对弈的一个棋谱：



2. 建模

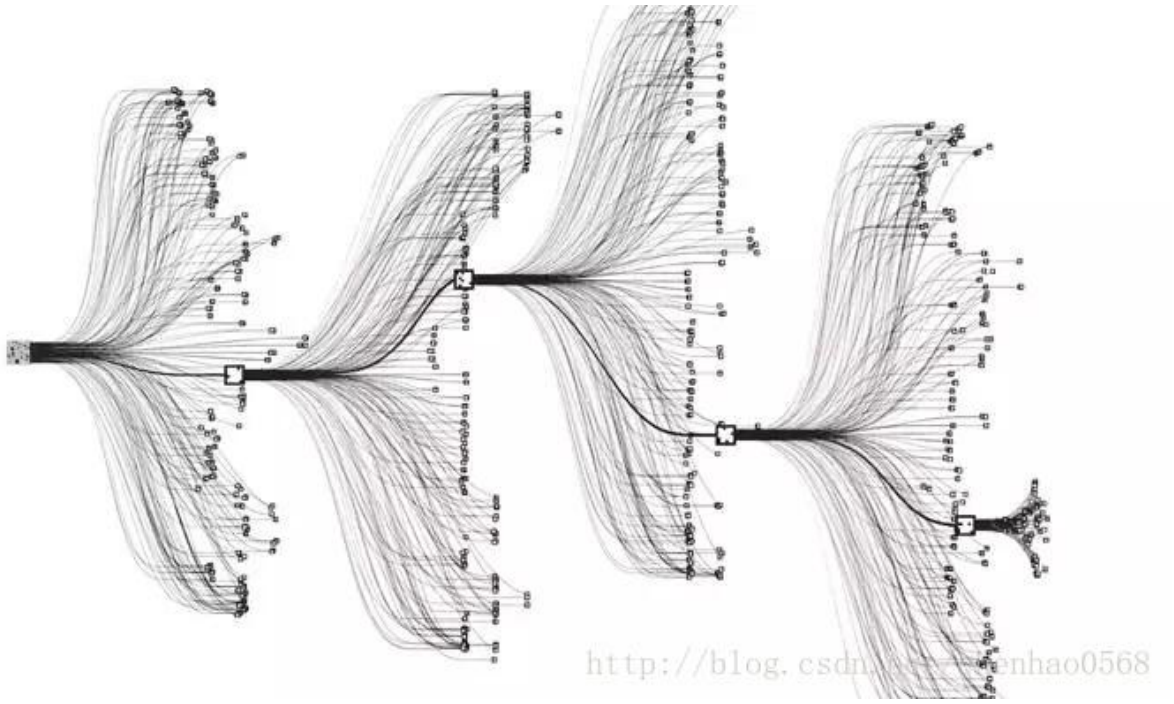
把这个游戏表示成一个机器可以处理的问题。

基本思路与难点

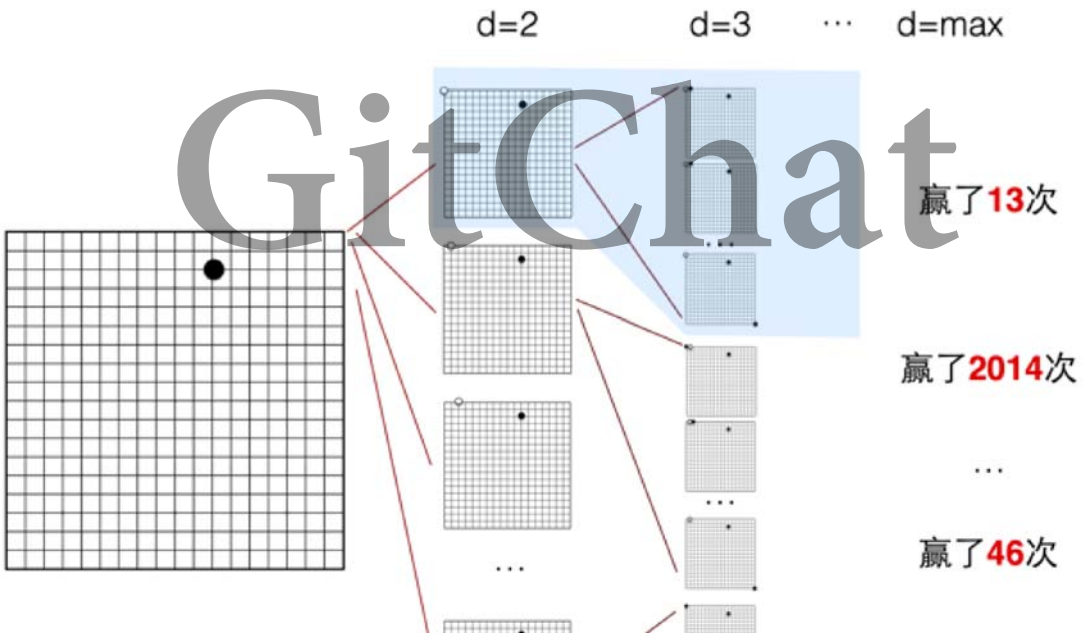
所有的问题都可以表示成一个抽象函数，有着输入和输出（数据可表示为<输入，输出>对）。而下棋，输入是棋盘状态，输出是当前状态的最佳行动（数据可表示为<棋盘状态，选择>对）。机器学习的任务就是从数据中学出来这个函数，至少越来越近似这个函数。

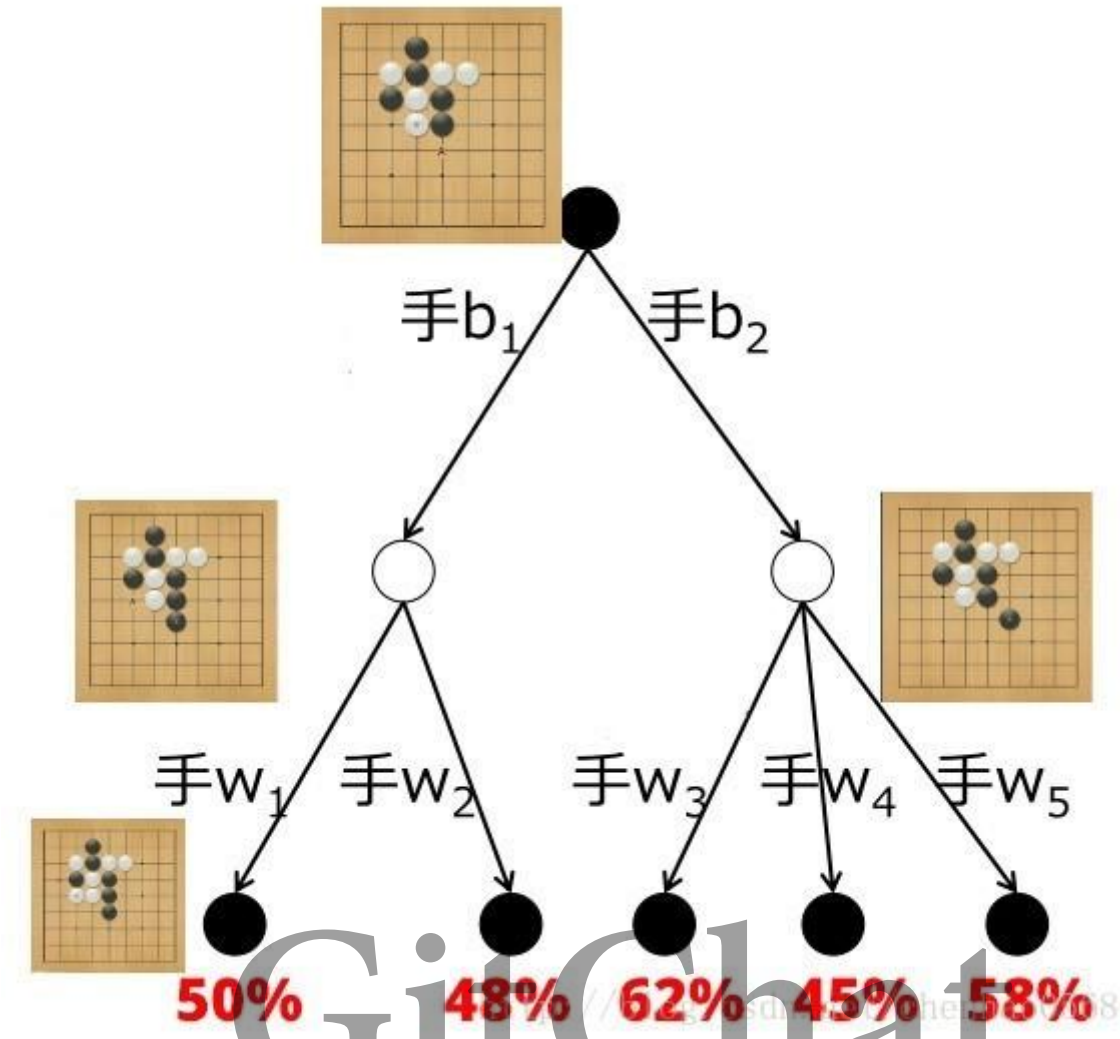
在下棋中，虽然每个棋盘状态下的最佳行动很难给出，但下完之后的输赢很容易判定。这就是奖惩机制，这就是一个典型的强化学习问题。

同西洋跳棋和国际象棋一样，围棋可以被建模成一个搜索问题。更具体一点，下棋的步骤可以表示成一个搜索树，其中每个节点代表一个棋盘状态。根节点是空白棋盘，每个



为了找到最优路径，人工智能传统的解决方案是"搜索"（如下图），搜索空间巨大，如宇宙星空。

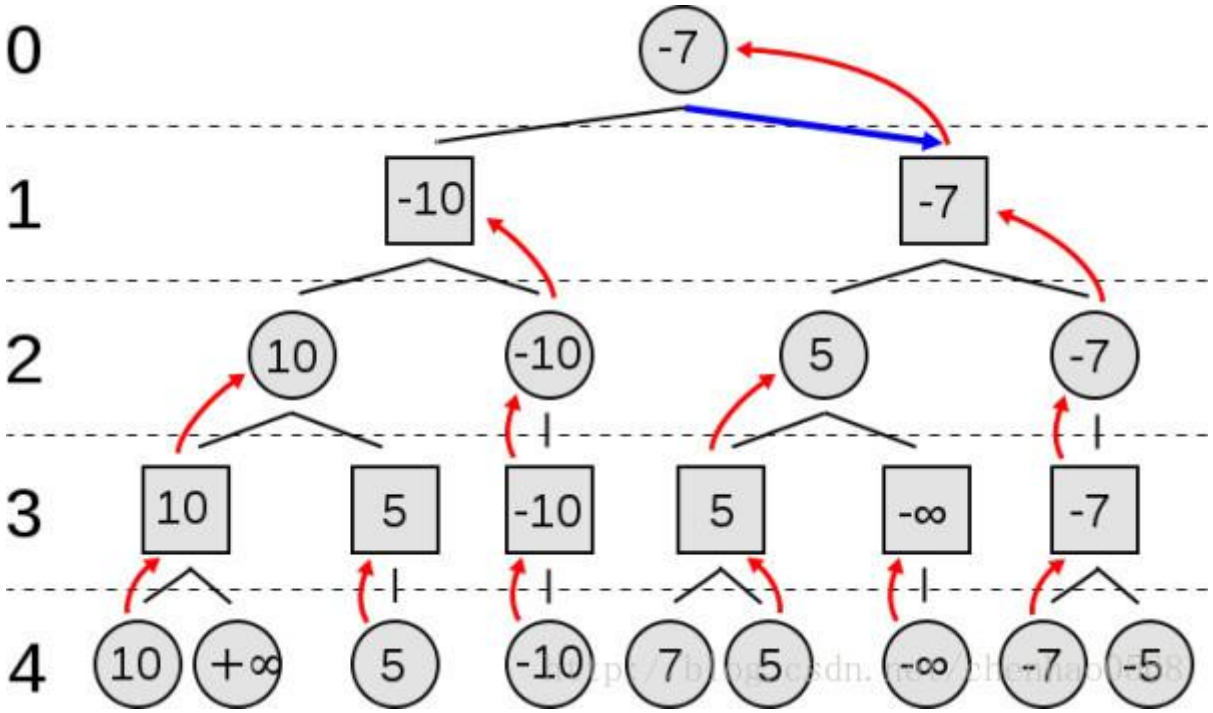




然后假设走完w1/w2/w3/w4/w5后，经过局面评估，黑棋的未来胜率分别是50%/48%/62%/45%/58%（这些胜率是怎么评估出来的？我们后文会说这个问题）。请问，黑棋此时最佳的着法是b1还是b2？

思考中……

如果白棋够聪明，会在黑棋走b1的时候回应以w2（尽量降低黑棋的胜率），在黑棋走b2的时候回应以w4（尽量降低黑棋的胜率）。所以走b1后黑棋的真实胜率是48%，走b2后黑棋的真实胜率是45%。黑棋的正解是b1。



在实际对局中，胜率评估会有不准确的地方，这就会导致"地平线效应"，即由于电脑思考的深度不够，且胜率评估不够准确，因此没有看见正解。

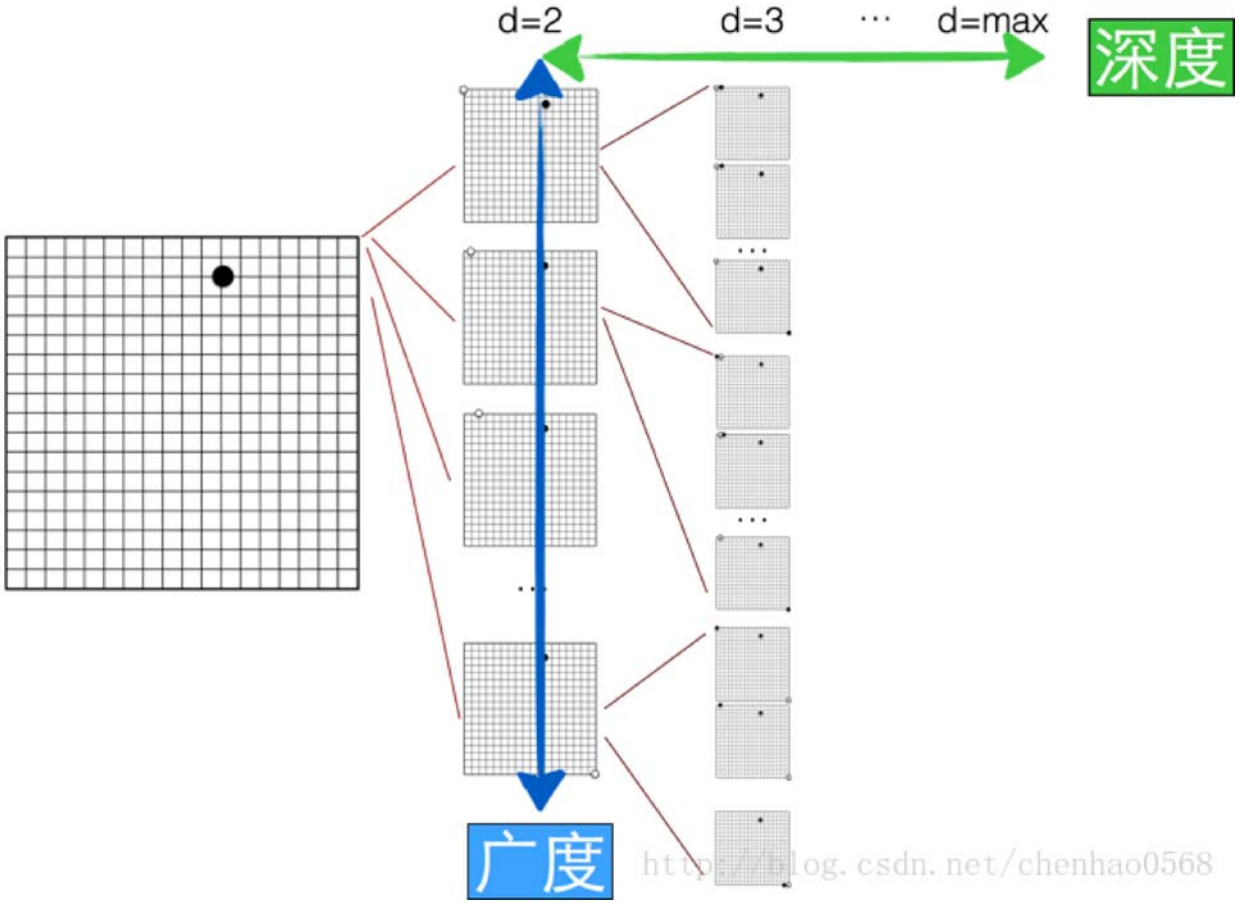
总结有两个难点：

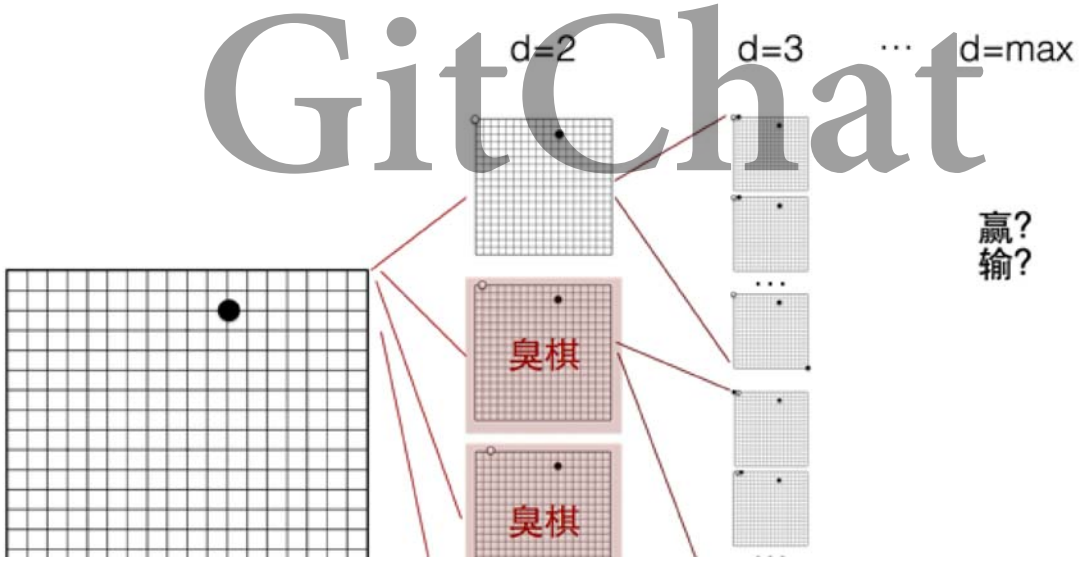
1. 搜索树太广。棋盘太大了，每一方在每一步都有很多着法可选。
2. 很难评估胜率。除非把搜索树走到终局，这意味着要走够三百多步（因为对于电脑来说，甚至很难判断何时才是双方都同意的终局，所以只能傻傻地填子，一直到双方都真的没地方可以走为止）。简单地说，搜索树也需要特别深。

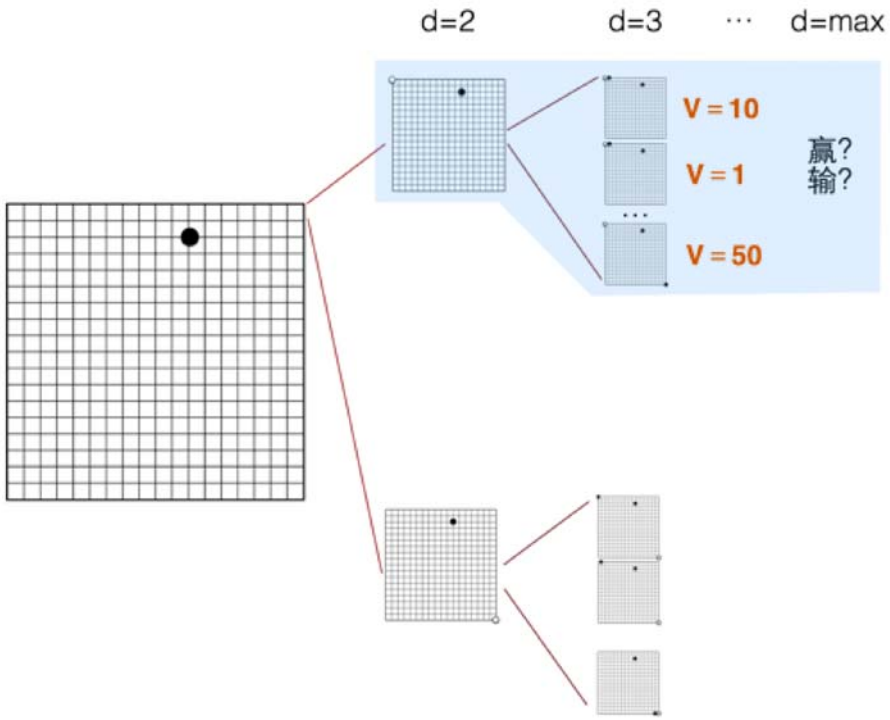
Alphago怎么做的？

人类不需要搜索这么多状态空间也能够下好围棋，说明还是有规律的。如何才能减少搜索空间呢？来看看Alphago怎么做的—- 关键是降低搜索广度与深度。

降低搜索广度：



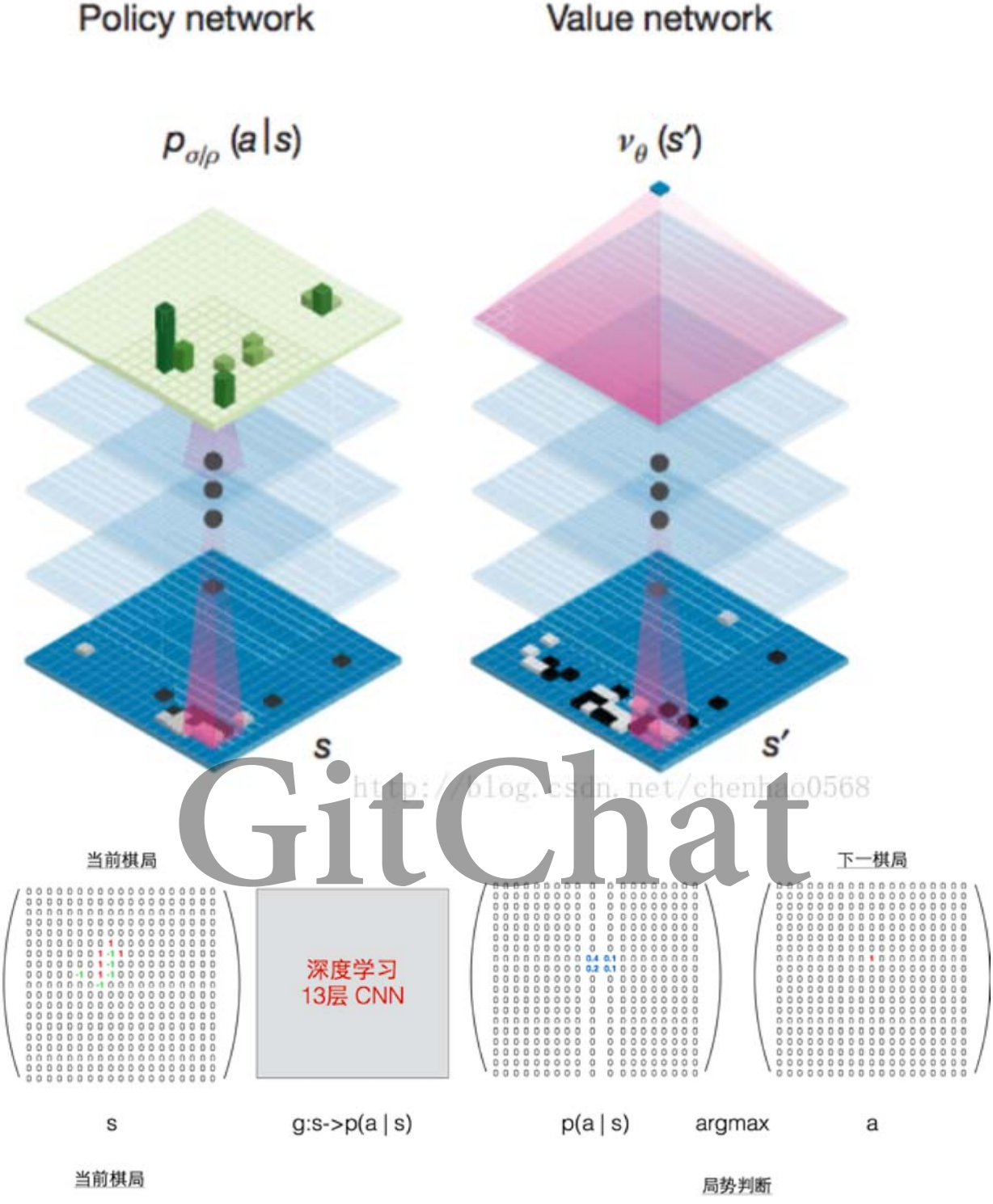


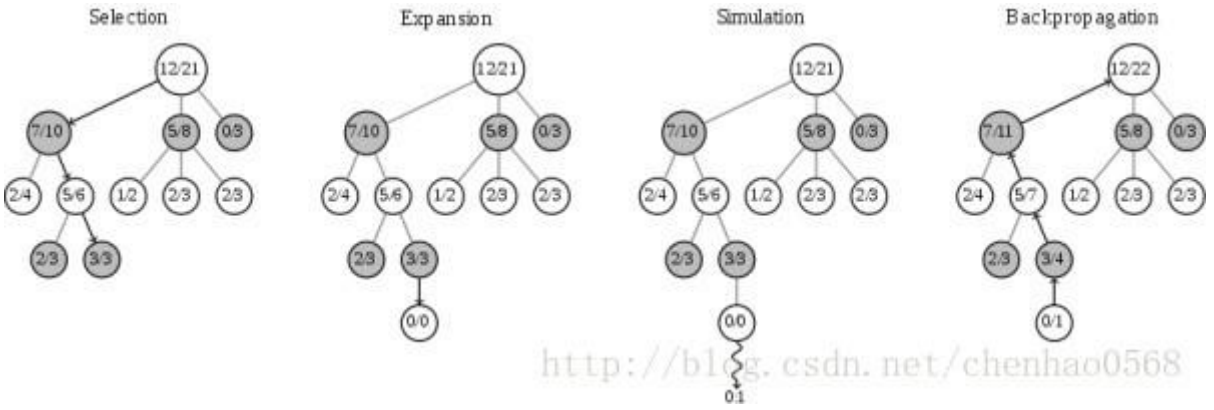


假如有另一个专家模型告诉我们当前棋局的“赢面”，就不需要“走那么深”了

Alphago构建了两种专家模型：落子预测器 + 棋盘价值评估器，论文上叫策略网络（policy networks）与估值网络（value networks）。

建模：P(落子行为| 当前棋局)





我们将不断重复一个过程（很多万次）：这个过程的第一步叫选择（Selection）。从根节点往下走，每次都选一个"最值得看的子节点"（具体规则稍后说），直到来到一个"存在未扩展的子节点"的节点，如图中的3/3 节点。什么叫做"存在未扩展的子节点"，其实就是指这个局面存在未走过的后续着法。

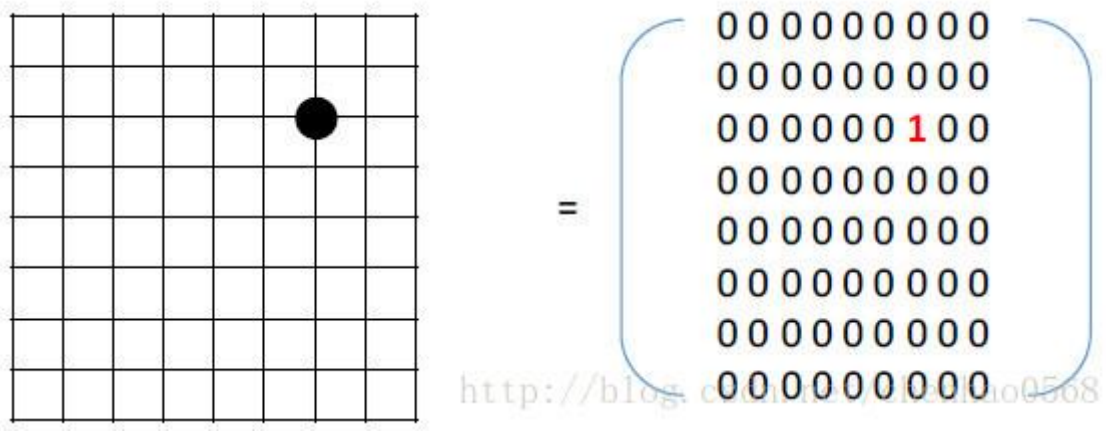
第二步叫扩展（Expansion），我们给这个节点加上一个0/0 子节点，对应之前所说的"未扩展的子节点"，就是还没有试过的一个着法。

第三步是模拟（Simluation）。从上面这个没有试过的着法开始，用快速走子策略（Rollout policy）走到底，得到一个胜负结果。按照普遍的观点，快速走子策略适合选择一个棋力很弱但走子很快的策略。因为如果这个策略走得慢（比如用AlphaGo 的策略网络走棋），虽然棋力会更强，结果会更准确，但由于耗时多了，在单位时间内的模拟次数就少了，所以不一定会棋力更强，有可能会更弱。这也是为什么我们一般只模拟一次，因为如果模拟多次，虽然更准确，但更慢。

第四步是回溯（Backpropagation）。把模拟的结果加到它的所有父节点上。例如第三步模拟的结果是0/1（代表黑棋失败），那么就把这个节点的所有父节点加上0/1。

这就是蒙特卡洛树搜索。它可以给出一个局面评估，虽然不准，但比没有强。另外，搜索树会较好地自动集中到"更值得搜索的变化"。

最后，AlphaGo 的策略网络，可以让我们更准确地选择需扩展的节点。而AlphaGo 的估值网络，可以与快速走子策略的模拟结果相结合，得到更准确的局面评估结果。

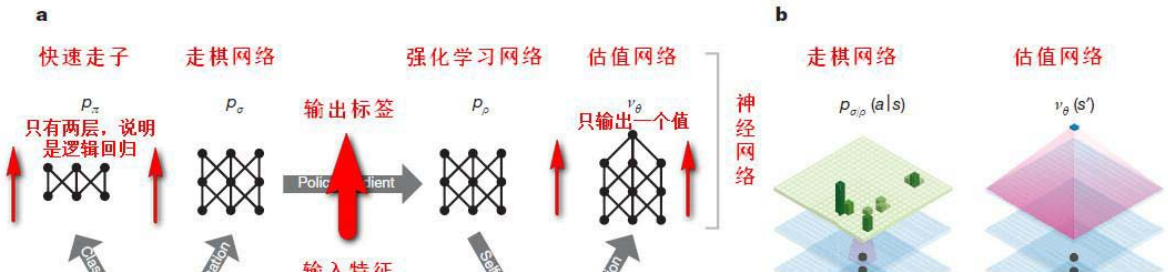


运算模型：围棋的大部分争夺是在局部区域进行的，不同的局部争夺共同组成了围棋的全局性。所以Alphago选择了适合于处理局部特征的卷积神经网络，构建了策略网络policy network、快速走子网络 playout network、估值网络 value network。

选择数据：DeepMind就是直接从围棋对战平台KGS（可以理解成外国的联众围棋游戏大厅）获得16万局6至9段人类选手的围棋对弈棋谱，总共有3000万个的< s , a >位置（s表示当前局面，a表示下一步落子位置），训练出来了一个类似人类下棋行为的模型。

DeepMind团队基于卷积神经网络和逻辑回归做了两个模型（监督学习策略网络和快速策略）来评估效果，发现结果不靠谱。赢棋者的落子不一定都是好棋（如两个臭棋篓子下棋），输棋者的落子不一定都是差棋（如两个顶尖高手的精彩对弈）。原因很可能是未穷尽棋局演化的各种可能，把臭棋也当做好棋来学了。

Alphago通过自我博弈（self-play）产生了3000万个标注样本< s' , z>（s表示当前局面，z表示这些局面对应的结果）。既然自我博弈可以自己产生数据，那么可否用自己产生的数据来训练策略网络自己呢？而这就是增强学习的思想。



AlphaGo Zero作出了很多重要的调整，去掉了人类专家对局的数据，而把剩下的合并成为一步"基于蒙特卡洛树搜索的强化学习"。还改进了蒙特卡洛树搜索方法，在每一步随机选取行动的时候，不再是完全随机，而是根据已经学习到的神经网络，尽量选取赢面大的行动。然后，强化学习又反过来使用了改进版本的蒙特卡洛树搜索方法得到的结果来调整自身参数。这样，蒙特卡洛树搜索和强化学习就很好地结合到了一起。

AlphaGo的算法原理基本介绍完了。AlphaGo，了不起！！！

四、阿尔法狗适用于哪些应用场景以及如何拿来用

能否把AlphaGo Zero中的技术推广并应用到其他领域？这还需要人工智能研究者们付出相当大的努力。其根本原因在于其他领域的基本难点于环境和围棋有本质上的区别。

下面我大致概括一下困难点，插入与AlphaGo Zero战胜人类的围棋问题作对比。

1. 建模。很多人工智能问题，连一个完整的数学模型都很难建立。例如玩星际争霸游戏、高考、自然语言理解等，虽然很容易对其中的某一部分建立一个模型，但很难把整个模型完整地统一起来。而围棋的建模非常简单。

2. 行动后果的不确定性。在围棋中，每个行动都有确定的结果，落子必定会成功。然而很多人工智能问题，行动的后果是不确定的，并不保证一定成功。例如传球的时候，球可能被敌方抢走，导致这个行动失败。

3. 环境的部分可观察性和动态性。例如星际争霸游戏中，敌方的军队位置和动向都是部分可观察的，并且不是一成不变的。而在围棋领域，这些都是完全可观察的和静态的。

4. 规则性，稳健性。比如在自然语言理解中，自然语言的规则不仅相当之多，而且很多情况下这些规则并不对所有情况都适用。而在围棋中，规则都是很简单且很通用的。

5. 意外和突发情况。例如在自动驾驶中，可能有很多意外情况，如突然蹿出来一个小孩。在这些领域中，几乎不可能列举所有的突发情况。而在围棋领域，完全没有意

AlphaGo对解决表示成超大状态空间搜索并有明显奖惩机制的问题（如很多智力游戏），有很大帮助。AlphaGo中应用的新评价函数具有一定的普适价值，因为很多人工智能中的问题都可以归结为大规模搜索的问题。

另外，可以借鉴AlphaGo的思路：根据特征，降低搜索广度与深度，减少计算量。

五、个人如何开发一款人工智能应用

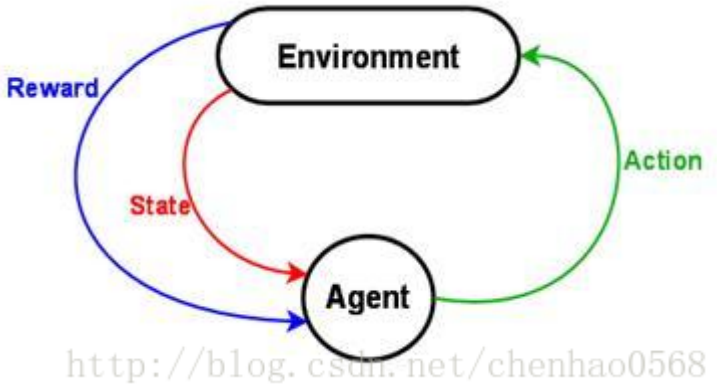
通过上面，你也了解了吧，步骤：

1. 选择目标：封闭、可控、高频、影响大。
2. 先了解业务。包括其基本规则、特性、处理的思路、目标等。
3. 建模，把这个表示成一个机器可以处理的问题。找到基本思路与难点。
4. 选择算法与实现功能。包括输入特征，运算模型，选择数据，优化等。
5. 改进、优化、扩展应用。加入其它机器学习理论和算法，或者把模型技术应用到其它领域。

示例：开发会玩游戏的人工智能应用。可以直接使用DeepMind最开始的这个：通过自我学习自动玩老的电视游戏。

1）选择一个目标，比如说玩老游戏。DeepMind最开始的这个：通过自我学习自动玩老的电视游戏。如下图

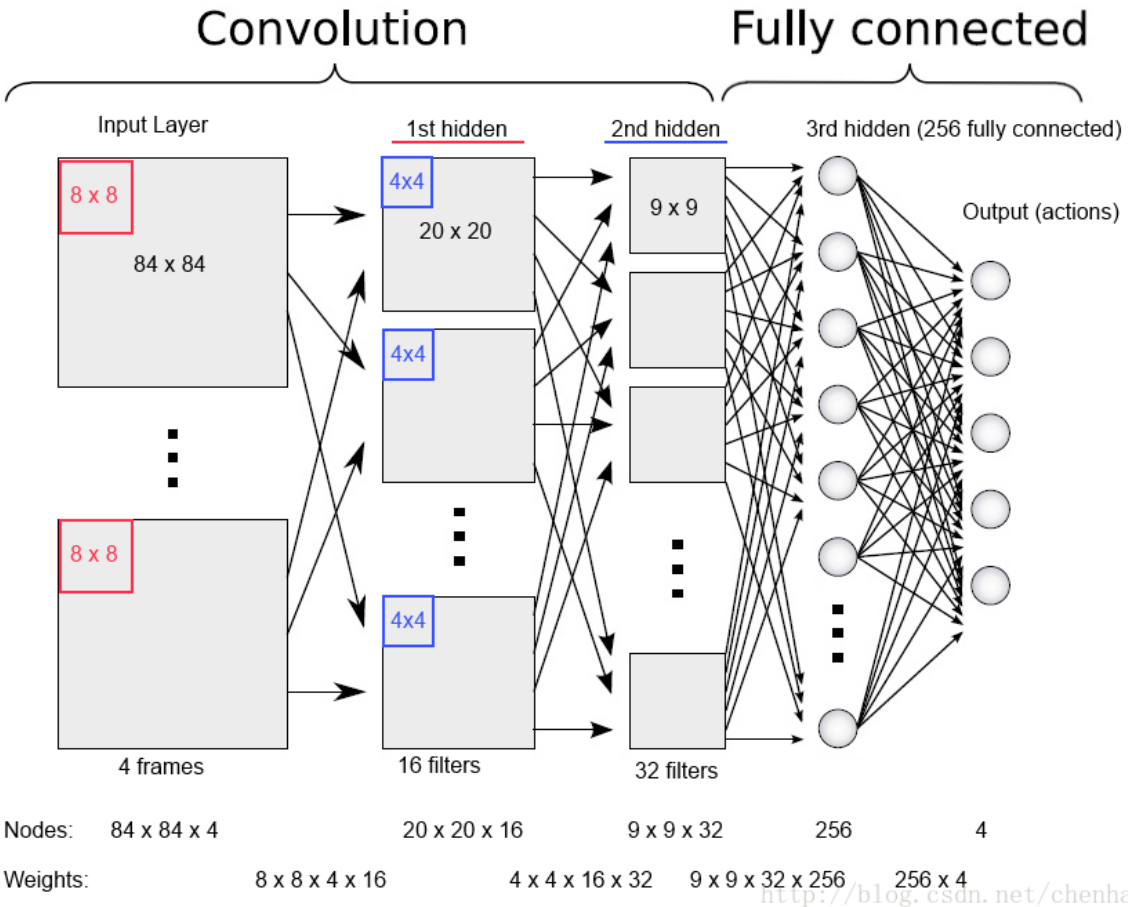
比如游戏仅仅通过屏幕上面的图像和游戏中的分数是否上升下降，从而做出选择性的动作。



4）各模块具体算法实现，用到什么学什么。

还是上面例子，在训练的一开始，这个程序对游戏一点都不了解。它并不知道这个游戏的目标，是保持生存、杀死谁或者是走出一个迷宫。它对这个游戏的影响也不清楚，并不知道它的动作会对这个游戏产生什么影响，甚至不知道这个游戏中会有哪些目标物品。通过在这个游戏中尝试并且一遍一遍失败，这个系统会逐渐学会如何表现来获得比较好的分数。

在这个系统中，一个神经网络被用来期望在当前游戏状态下每种可能的动作所得到的反馈。下图给出了文章中所提到的神经网络。这个网络能够回答一个问题，比如"如果这么做会变得怎么样？"。网络的输入部分由最新的四幅游戏屏幕图像组成，这样这个网络不仅仅能够看到最后的部分，而且能够看到一些这个游戏是如何变化的。输入被经过三个后继的隐藏层，最终到输出层。



输出层对每个可能的动作都有一个节点，并且这些节点包含了所有动作可能得到的反馈。在其中，会得到最高期望分数的反馈会被用来执行下一步动作。

他们的源代码可以在 GitHub主页上找到。

5）为最小模型加入其它机器学习理论和算法，或者把模型应用到其它领域。
DeepMind后来把这些技术应用到了围棋上，产生了AlphaGo。各种算法都有他们的用途，看你想让机器干什么。比如标准围棋是19*19，你能否扩展到M*N呢？

我们减少点运算量，以纵横各4条直线将棋盘分成16个交叉点，是不是简单多了？16！=约21万亿。有兴趣的可以做一下。听说7*7围棋在AlphaGo之前就被人用人工智能解决了。

如果你不想上班、不能坚持、不能受苦、不能受累，动不动就心存委屈，动不动就放弃，不想出力还想很赚钱，怎么办？

——买个碗，你就是老板！

警惕两坑。第一不要把AI用作噱头，比如人脸识别登录APP，这在目前依然是属于炫技噱头，完全不考虑移动端用户体验的流氓做法；第二也不要对AI有不切实际的期待，认为它能做很多复杂的事情，比如AI聊天机器人，做你的情感伴侣，要知道，即使是微软和苹果公司的AI水平，现在也很难满足你那颗寂寞的心。

首先不要选开放环境，要选一个封闭可控的环境。比如说做个仓储搬货机器人，就是在仓库里面，从A到B，完成一个固定的任务。这里面场景是封闭的，光线是可控的，路线是受约束的。这种任务，目前AI就可以做得比较好。不要去碰自动驾驶汽车这种开放场景的，有迎着阳光的，有背着阳光的，有隧道里，有高速上的。即使是现在最牛的自动驾驶汽车，现在到了下雨下雪天，基本都会歇菜。

或者你做客服机器人，专注处理顾客遇到的业务上的问题，这就是封闭场景，你可以不用解决全部问题，而只是那些高频重复的标准问题，在这样一个可控的范围内，你把体验做好。先解决一个具体的小问题。产品稳定，又能创造价值。

第二，不要完全替代人，要辅助、增强也行。人工智能只有少数领域的少数场景下，才能做到超过人类水平。所以暂时还是需要保持理性，当下这个阶段，即使把辅助、增强做好，也很不容易。

第三，不要等待大数据，要巧妙采集、草船借箭。现在大家都意识到做AI，拥有大量的数据特别重要。但对于初创团队来说，要弄到海量数据，成本极高。那要如何采集呢？

第四，不要过于相信算法，要精心设计聪明的容错方案。算法是一定会出问题的，要通过设计容错方案，保证出错情况下的用户体验。比如出错直接人工服务。

一个能把握真正价值，对人性或者业务洞悉的人才，对产品成败至关重要。

B. 如果搞出好产品了，或已经有产品了，如何赢得竞争？主要是三方面：真正懂深度学习算法的人才、领先的数据量和领先的硬件计算资源。

C. 如果技术不行或资源不够，用大平台的接口，如百度，微软，facebook，google，亚马逊等。

D. 再实在不行，销售人工智相关工具或产品，自己做不出来，卖别人的总行吧。卖软硬件。老子转换干销售，能侃吧。

- https://mp.weixin.qq.com/s/AoL5a73YNpaqmNviCLaY6g
- http://www.infoq.com/cn/news/2014/10/deepmind
- http://blog.csdn.net/longxinchen_ml/article/details/50900070
- https://yq.aliyun.com/articles/8239