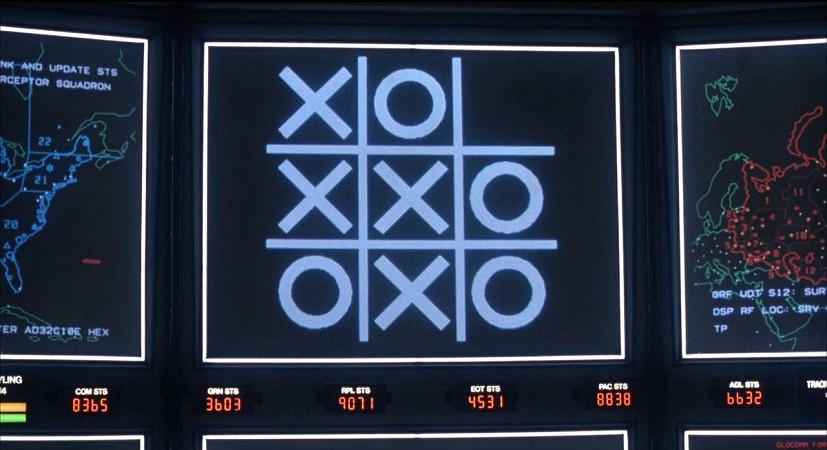
# Why AlphaGo Zero is a Quantum Leap Forward in Deep Learning 为什么AlphaGo Zero是深度学习的一个重大飞跃

原文链接：  
原文链接：



img  
img公司

Credit: War Games (1983)   
学分：战争游戏（1983）

Self-play is Automated Knowledge Creation  
自我游戏是自动的知识创造

The 1983 movie “War Games” has a memorable climax where the supercomputer known as WOPR (War Operation Plan Response) is asked to train on itself to discover the concept of an un-winnable game. The character played by Mathew Broderick asks “Is there any way that it can play itself?”  
1983年的电影《战争游戏》有一个令人难忘的高潮，超级计算机WOPR（War Operation Plan Response）被要求对自己进行训练，以发现不可赢游戏的概念。由马修·布罗德里克扮演的角色问道：“有什么方法可以让它自己发挥作用吗？”

34 years later, DeepMind has shown how this is exactly done in real life! The solution is the same, set the number of players to zero (i.e. zero humans).  
34年后，DeepMind展示了这在现实生活中是如何做到的！解决方法是相同的，将玩家数量设置为零（即零人类）。

There is plenty to digest about this latest breakthrough in Deep Learning technology. DeepMind authors use the term “self-play reinforcement learning”. As I remarked in the piece about “”, DeepMind is particularly fond of their Reinforcement Learning (RL) approach. DeepMind has taken the use of Deep Learning layers in combination with more classical RL approaches to an art form.  
关于深度学习技术的这一最新突破，有很多东西需要消化。DeepMind的作者使用“自我游戏强化学习”一词。正如我在“关于”一文中所说，DeepMind特别喜欢他们的强化学习（RL）方法。DeepMind将深层学习层与更经典的RL方法结合起来，形成了一种艺术形式。

AlphaGo Zero is the latest incarnation of its Go-playing automation. One would think that it would be hard to top the AlphaGo version that bested the human world champion in Go. AlphaGo Zero however not only beats the previous system, but does it in a manner that validates a revolutionary approach. To be more specific, this is what AlphaGo has been able to accomplish:  
AlphaGo Zero是其Go Play自动化的最新体现。有人会认为，在围棋领域击败人类世界冠军的AlphaGo版本是很难超越的。然而，AlphaGo Zero不仅击败了以前的系统，而且以一种验证革命性方法的方式做到了这一点。更具体地说，这是AlphaGo能够做到的：

1. Beat the previous version of AlphaGo (Final score: 100–0).  
   击败之前版本的AlphaGo（最终得分：100-0）。
2. Learn to perform this task from scratch, without learning from previous human knowledge (i.e. recorded game play).  
   学习从零开始执行这项任务，而不必学习以前的人类知识（即录制的游戏）。
3. World champion level Go playing in just 3 days of training.  
   世界冠军级别的比赛只需3天的训练。
4. Do so with an order of magnitude less neural networks ( 4 TPUs vs 48 TPUs).  
   使用一个数量级较少的神经网络（4个tpu对48个tpu）。
5. Do this with less training data (3.9 million games vs 30 millions games).  
   使用更少的训练数据（390万场对3000万场）。

Each of the above bullet points is a newsworthy headline. The combination of each bullet point and what it reveals is completely overwhelming. This is my honest attempt to make sense of all of this.  
以上每个要点都是有新闻价值的标题。每一个要点和它所揭示的内容的结合是完全压倒性的。这是我想弄明白这一切的真诚尝试。

The first bullet point for many will seem unremarkable. Perhaps it’s because incremental improvements in technology have always been the norm. Perhaps one algorithm besting another algorithm 100 straight times intuitively doesn’t have the same appeal of one human besting another human 100 straight times. Algorithms don’t have the kind of inconsistency that we find in humans.  
对许多人来说，第一个要点似乎并不明显。也许是因为技术上的进步一直是常态。也许一个算法在直觉上超过另一个算法100次，并不像一个人在直觉上超过另一个人100次那样有吸引力。算法没有我们在人类身上发现的那种不一致性。

One would expect though that the game of Go would have a large enough search space that there would be a chance of a less capable algorithm to be lucky enough to beat a better own. Could it be that AlphaGo Zero has learned new alien moves that its competitors are unable to reason about the same search space and thus having an insurmountable disadvantage. This apparently seems to be the case and is sort of alluded to by the fact that AlphaGo Zero requires less compute resources to best its competitors. Clearly, it’s doing a lot less work, but perhaps it is just working off a much richer language of Go strategy. Less work is what biological creatures aspire to do. Language compression is a means to arrive at less cognitive work.  
尽管围棋游戏会有足够大的搜索空间，但人们还是会期望有机会找到一个能力较弱的算法，幸运地击败一个更好的算法。可能是AlphaGo Zero学会了新的外星移动，其竞争对手无法对相同的搜索空间进行推理，因此具有不可逾越的劣势。很明显，事实上AlphaGo Zero需要更少的计算资源才能使其竞争对手达到最佳状态，这似乎是事实，也是一种暗示。显然，它做的工作要少得多，但也许它只是在制定一个更加丰富的围棋语言策略。生物渴望做的是更少的工作。语言压缩是减少认知工作的一种手段。

The second bullet point challenges our current paradigm of supervised only machine learning. The original AlphaGo was bootstrapped using previously recorded tournament gameplay. This was then followed with self-play to improve its two internal neural networks (i.e. policy and value networks). In contrast, AlphaGo Zero started from scratch with just the rules of Go programmed. It also required a single network rather than two. It is indeed surprising that it was able to bootstrap itself and then eventually learning more advanced human strategies as well as previously unknown strategies. Furthermore, the order in what strategies it learned first were sometimes unexpected. It is as if the system had learned a new internal language of how to play Go. It is also interesting to speculate as to the effect of a single integrated neural network versus two disjoint neural networks. Perhaps there are certain strategies that a disjoint network cannot learn.  
第二个要点挑战了我们目前的仅监督机器学习的范式。最初的AlphaGo是使用先前录制的锦标赛游戏进行引导的。然后，在这之后进行自我游戏，以改进其两个内部神经网络（即策略和价值网络）。相比之下，AlphaGo Zero从零开始就只使用了Go编程规则。它还需要一个网络而不是两个。确实令人惊讶的是，它能够自我引导，然后最终学习更先进的人类策略以及之前未知的策略。此外，它最初学习策略的顺序有时是出乎意料的。这就好像系统已经学会了一种新的内部语言如何玩围棋。同样有趣的是，我们可以推测一个单一的集成神经网络和两个不相交的神经网络的效果。也许有些策略是不相交的网络无法学习的。

Humans learn languages through metaphors and stories. The human strategies discovered in Go are referred to with names so as to be recognizable by a player. It could be possible that the human language of Go is inefficient in that it is unable to express more complex compound concepts. What AlphaGo Zero seems to be able to do is perform its moves in a way that satisfies multiple objectives at the same time. So humans and perhaps earlier versions of AlphaGo were constrained to a relatively linear way of thinking, while AlphaGo Zero was not encumbered with an inefficient language of strategy. It is also interesting that one may consider this a system that actually doesn’t use the implicit bias that may reside in a language. David Silver, of DeepMind, has an even more bold claim:  
人类通过隐喻和故事学习语言。在围棋中发现的人类策略是用名字来指代的，以便玩家能够识别。Go的人类语言可能效率低下，因为它无法表达更复杂的复合概念。AlphaGo Zero似乎能够做到的是以同时满足多个目标的方式执行其动作。因此，人类和早期版本的AlphaGo被限制在相对线性的思维方式上，而AlphaGo Zero并没有被低效的策略语言所困扰。同样有趣的是，人们可能会认为这是一个系统，实际上没有使用可能存在于语言中的隐含偏见。来自DeepMind的David Silver更大胆地宣称：

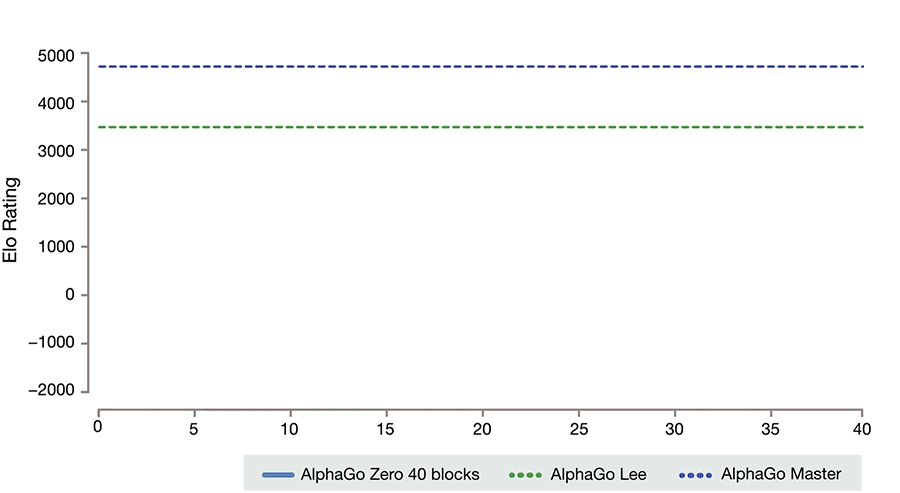
It’s more powerful than previous approaches because by not using human data, or human expertise in any fashion, we’ve removed the constraints of human knowledge and it is able to create knowledge itself.  
它比以前的方法更强大，因为通过不使用人类数据或任何方式的人类专业知识，我们已经消除了人类知识的限制，它能够自己创建知识。

The about some interesting observation of the game play of this new system:  
关于这个新系统游戏的一些有趣的观察：

Expert players are also noticing AlphaGo’s idiosyncrasies. Lockhart and others mention that it almost fights various battles simultaneously, adopting an approach that might seem a bit madcap to human players, who’d probably spend more energy focusing on smaller areas of the board at a time.  
专家玩家也注意到了AlphaGo的特质。洛克哈特和其他人提到，它几乎同时打各种各样的仗，采用一种对人类玩家来说可能有点疯狂的方法，这些玩家可能一次会花更多的精力专注于董事会较小的领域。

The learned language is devoid of any historical baggage that it may have accumulated over the centuries of Go study.  
学习的语言没有任何历史包袱，它可能已经积累了几个世纪的围棋学习。

The third bullet point says that training time is also surprisingly less than its previous incarnation. It is as if AlphaGo Zero learns how to improve its own learning.  
第三个要点是，训练时间也比前一次出人意料地少。就好像AlphaGo Zero学会了如何提高自己的学习水平。



img  
img公司

Source:   
资料来源：

It took only 3 days to get to a level that beats the best human player. Furthermore, it just keeps getting better even after it surpasses the best previous AlphaGo implementation. How is it capable of improving its learning continuously? This ability to incrementally learn and improve the same neural network is something we’ve seen in another architecture known as FeedbackNet. In the commonplace SGD based learning, the same network is fed data across multiple epochs.  
只花了3天的时间就达到了击败人类最佳选手的水平。而且，即使它超过了以前最好的AlphaGo实现，它也会变得更好。它如何能够不断地提高自己的学习水平？这种逐步学习和改进同一个神经网络的能力是我们在另一个称为反馈网的架构中看到的。在常见的基于SGD的学习中，同一个网络在多个时间段上传送数据。

Here however, each training set is entirely new and increasingly more challenging. It is also analogous to curriculum learning, however the curriculum is intrinsic in the algorithm. The training set is self generated and the calculation of the objective function is derived from the result of MCTS. The network learns by comparing itself not from external training data but from synthetic data that is generated from a previous version of the neural network.  
然而，在这里，每个训练集都是全新的，而且越来越具有挑战性。它也类似于课程学习，但课程是内在的算法。训练集是自生成的，目标函数的计算由MCTS的结果导出。该网络不是从外部训练数据中学习，而是从以前版本的神经网络生成的合成数据中学习。

The fourth bullet point, the paper reports that it took only 4 Google TPUs ( ) as compared to 48 TPUs for previous systems. Even surprisingly, the Nature paper notes that this ran on a single system and did not use distributed computing. So anyone with four Volta based Nvidia GPUs has the horse power to replicate these results. Performing a task with 1/10th the amount of compute resources should be a hint to anyone that something fundamentally different is happening over here. I have yet to analyze this in detail, but perhaps the explanation is due to just a simpler architecture.  
第四个要点是，该报报道说，与之前系统的48个tpu相比，它只使用了4个Google tpu（）。甚至令人惊讶的是，《自然》杂志的论文指出，这是在一个单一的系统上运行的，并没有使用分布式计算。所以任何拥有四个基于Volta的Nvidia gpu的人都有能力复制这些结果。用十分之一的计算资源来执行一个任务，对任何人来说都应该是一个提示：这里正在发生一些根本不同的事情。我还没有详细分析这个问题，但可能是因为一个更简单的架构。

Finally, the last bullet point where it appears that AGZ advanced its capabilities using less training data. It appears that the synthetic data generated by self-play has more ‘teachable moments’ than data that’s derived from human play. Usually, the way to improve a network is to generate more synthetic data. The usual practice is to augment data by doing all sorts of data manipulations (ex. cropping, translations, etc), however in AGZ’s case, the automation seemed to be able to select richer training data.  
最后，最后一个要点是AGZ使用较少的训练数据提高了它的能力。似乎由自我游戏生成的合成数据比由人类游戏生成的数据具有更多的“可教时刻”。通常，改进网络的方法是生成更多的合成数据。通常的做法是通过各种数据操作（例如裁剪、翻译等）来增加数据，但是在AGZ的例子中，自动化似乎能够选择更丰富的训练数据。

Almost every new Deep Learning paper that is published (or found in Arxiv) tends to show at best a small percentage improvement over previous architectures. Almost every time, the newer implementation also requires more resources to achieve higher prediction accuracies. What AlphaGo has shown is unheard of, that is, it requires an order of magnitude less resources and a less complex design, while unequivocally besting all previous algorithms.  
几乎所有出版的新的深度学习论文（或在ARXIV中发现）都倾向于显示比以前的架构小的百分比改进。几乎每次，新的实现都需要更多的资源来实现更高的预测精度。AlphaGo所展示的是闻所未闻的，也就是说，它需要的资源数量级更少，设计也不那么复杂，同时毫不含糊地胜过所有以前的算法。

Many long time practitioners of reinforcement learning applied to games have commented that the actual design isn’t even novel and has been formulated decades ago. Yet, the efficacy of this approach has finally been experimentally validated by the DeepMind team. In Deep Learning like in sports, you can’t win on paper, you actually have to play the game to see who wins. In short, no matter a simple an idea may be, you just never know how well it will work unless the experiments are actually run.  
许多应用于游戏的强化学习的长期实践者评论说，实际的设计甚至不是新颖的，而是几十年前制定的。然而，这种方法的有效性最终得到了DeepMind团队的实验验证。在像体育这样的深度学习中，你不可能在纸面上获胜，你实际上必须要玩游戏，看谁赢。简言之，不管一个简单的想法是什么，除非实验真的进行了，否则你永远不知道它会有多好。

There is nothing new about the or the architecture of the neural network. Policy iteration is a old algorithm that learns improving policies, by alternating between policy estimation and policy improvement . That is, between estimating the value function of the current policy and using the current value function to find a better policy.  
神经网络的结构没有什么新意。策略迭代是一种通过策略估计和策略改进交替学习改进策略的旧算法。也就是说，在估计当前政策的价值函数和利用当前价值函数寻找更好的政策之间。

The single neural network that it uses is a pedestrian convolution network:  
它使用的单一神经网络是行人卷积网络：

The overall network depth, in the 20- or 40-block network, is 39 or 79 parameterized layers, respectively, for the residual tower, plus an additional 2 layers for the policy head and 3 layers for the value head.  
在20或40块网络中，剩余塔的总网络深度分别为39或79个参数化层，加上策略头的额外2层和值头的3层。

Like the previous incarnations of AlphaGo, Monte Carlo Tree Search (MCTS) is used to select the next move. AlphaGo Zero takes advantage of the calculations of the tree search as a way to evaluate and train the neural network. So basically, MCTS employing a previously trained neural network, performs a search for winning moves. The policy evaluation estimates the value function from many sampled trajectories. The results of this search is then used to drive the learning of the neural network. So after every game, a new and potentially improved network is selected for the next self-play game. DeepMind calls this “Self-play reinforcement learning”:  
与AlphaGo的前一个版本一样，Monte Carlo树搜索（MCTS）用于选择下一步。AlphaGo Zero利用树搜索的计算作为评估和训练神经网络的方法。因此，基本上，MCT使用一个先前训练过的神经网络，执行一个搜索获胜的动作。策略评估从许多采样轨迹估计值函数。然后将搜索结果用于驱动神经网络的学习。所以在每一个游戏之后，一个新的和潜在的改进的网络被选为下一个自玩游戏。DeepMind称之为“自我游戏强化学习”：

A novel reinforcement learning algorithm. MCTS search is executed, guided by the neural network fθ. The MCTS search outputs probabilities π of playing each move. These search probabilities usually select much stronger moves than the raw move probabilities p of the neural network fθ(s); MCTS may therefore be viewed as a powerful policy improvement operator.  
一种新的强化学习算法。在神经网络fθ的指导下进行MCTS搜索。MCTS搜索输出播放每个动作的概率π。这些搜索概率通常选择比神经网络fθ（s）的原始移动概率p强得多的移动；因此MCTS可以被视为一个强大的策略改进算子。

Self-play with search — using the improved MCTS-based policy to select each move, then using the game winner z as a sample of the value — may be viewed as a powerful policy evaluation operator.  
Self play with search - 使用改进的基于MCTS的策略选择每个移动，然后使用游戏赢家z作为值的样本 - 可以被视为一个强大的策略评估运算符。

With each iteration of self-play, the system learns to become a stronger player. I find it odd that the exploitive search mechanism is able to creatively discover new strategies while simultaneous using less training data. It is as if self-play is feeding back into itself and learning to learn better.  
随着自我游戏的每一次迭代，系统学会成为一个更强大的玩家。我觉得奇怪的是，开发性搜索机制能够创造性地发现新的策略，同时使用较少的训练数据。这就好像自我游戏在自我反哺，学习更好地学习。

This self-play reminds me of an earlier writing about “.” I wrote about many recent advances in Deep Learning such as Ladder networks and Generative Adversarial Networks (GANs) that exploited a loop based method to improve recognition and generation. It seems that when you have this kind of mechanism, that is able to perform assessments of its final outputs, that the fidelity is much higher with less training data. In the case of AlphaGo Zero, there’s is no training data to speak of. The training data is generated through self-play. A GAN for example, collaboratively improves its generation by having two networks (discriminator and generator) work with each other. AlphaGo Zero, in contrast pits the capabilities of a network trained in a previous game against that of the current network. In both cases, you have two networks that feed of each other in training.  
这个自我游戏让我想起了之前的一篇关于“的文章。我写了很多关于深度学习的最新进展，比如阶梯网络和生成性对抗网络（GANs），它们利用基于循环的方法来提高识别和生成能力。当你有这样一种机制，能够对它的最终输出进行评估时，它的保真度似乎更高，训练数据更少。在AlphaGo Zero的例子中，没有训练数据可言。训练数据通过自玩生成。例如，GAN通过使两个网络（鉴别器和生成器）相互协作来改进其生成。相比之下，AlphaGo Zero将在前一个游戏中训练的网络的能力与当前网络的能力进行了对比。在这两种情况下，您都有两个在培训中相互补充的网络。

An important question that should be in everyone’s mind is: “How general is AlphaGo Zero’s algorithm?” DeepMind has publicly stated that they will be . Earlier I wrote about how to assess the appropriateness of Deep Learning technologies (see: ). In that assessment, there are six uncertainties in any domain that needs to be addressed: execution uncertainty, observational uncertainty, duration uncertainty, action uncertainty, evaluation uncertainty and training uncertainty.  
每个人都应该想到的一个重要问题是：“AlphaGo Zero的算法有多普遍？“DeepMind已经公开表示，他们会的。早些时候，我曾写过如何评估深度学习技术的适当性（见：）。在评估中，任何领域都有六个不确定性需要解决：执行不确定性、观察不确定性、持续时间不确定性、行动不确定性、评估不确定性和培训不确定性。

In the AlphaGo Zero, the training uncertainty, seems to have been addressed. AlphaGo Zero learns the best strategies by just playing against itself. That is, it is able to “imagine” situations and then discover through self-improvement the best strategies. It can do this efficiently because all the other uncertainties are known. That is, there is no indeterminism in the results of a sequence of actions. There is complete information. The effects of actions are predictable. There is a way to measure success. In short, the behavior of the game of Go is predictable, real world systems however are not.  
在alphago0中，训练的不确定性似乎已经得到了解决。AlphaGo Zero只需与自身对抗，就能学会最佳策略。也就是说，它能够“想象”情况，然后通过自我改进发现最佳策略。它可以有效地做到这一点，因为所有其他的不确定性都是已知的。也就是说，一系列行动的结果没有不确定性。有完整的信息。行动的效果是可以预见的。有一种方法可以衡量成功。简言之，围棋游戏的行为是可预测的，而现实世界的系统却不是。

In many real world contexts however, we can still build accurate simulations or virtual worlds. Certainly the policy iteration methods found here may seem to be applicable to these virtual worlds. Reinforcement learning has been applied to virtual worlds (i.e. video games and strategy games). DeepMind has not yet reported experiments of using policy iteration in Atari games. Most games of course don’t need this sophisticated look ahead that requires MCTS, however there are some games like Montezuma’s Revenge that do. DeepMind’s Atari game experiments were like AlphaGo Zero, in that there was no need for human data to teach a machine.  
然而，在许多现实世界中，我们仍然可以建立精确的模拟或虚拟世界。当然，这里找到的策略迭代方法似乎适用于这些虚拟世界。强化学习已经被应用于虚拟世界（如电子游戏和策略游戏）。DeepMind还没有报道在Atari游戏中使用策略迭代的实验。当然，大多数游戏不需要这种复杂的前瞻性，需要MCT，但也有一些游戏，像Montezuma的复仇，这样做。DeepMind的Atari游戏实验就像AlphaGoZero一样，不需要人工数据来教机器。

The difference between AlphaGo Zero and the video game playing machines is that the decision making at every state in the game is much more sophisticated. In fact there is an entire spectrum of decision making required for different games. Is MCTS the most sophisticated algorithm that we will ever need?  
AlphaGoZero和电子游戏机的区别在于，游戏中每个州的决策都要复杂得多。事实上，不同的游戏需要一整套的决策过程。MCTS是我们需要的最复杂的算法吗？

There is also a question on strategies that require remembering one’s previous move. AlphaGo Zero appears to only care about the current board state and does not have a bias on what it moved previously. A human sometimes may determine its own action based on its previous move. It is a way of telegraphing actions to an opponent, but it usually is more like a head fake. Perhaps that’s a strategy that only works on humans and not machines! In short, a machine cannot see motion if it was never trained to recognize its value.  
还有一个关于策略的问题，需要记住自己之前的动作。AlphaGo Zero似乎只关心当前的板状态，对它之前移动的内容没有偏见。一个人有时会根据自己先前的行动来决定自己的行动。这是一种用电报向对手汇报行动的方式，但通常更像是一个假头。也许这是一个只对人类而不是机器有效的策略！简而言之，如果机器从未接受过识别其价值的训练，它就看不到运动。

This lack of memory affecting strategy may in fact be advantageous. Humans when playing a strategy game will stick to a specific strategy until an unexpected event disrupts that strategy. So long as an opponent’s moves are as expected, there is no need to change a strategy. However, as we’ve seen in the most advanced Poker playing automation, there is a distinct advantage of always calculating strategy from scratch with every move. This approach avoids telegraphing any plans and therefore a good strategy. However, misdirection is a strategy that is effective against humans but not machines that are not trained to be distracted by them. (Editors Note: Apparently previous board states are used as input to the network, so appears this lack of memory observation is incorrect).  
这种缺乏记忆的影响策略实际上可能是有利的。人类在玩一个策略游戏时会坚持一个特定的策略，直到一个意外事件破坏了这个策略。只要对手的动作符合预期，就没有必要改变策略。然而，正如我们在最先进的扑克游戏自动化中看到的，每次移动都会从头开始计算策略，这是一个明显的优势。这种方法避免了电报任何计划，因此是一个好的战略。然而，误导是一种对人类有效的策略，而不是那些没有被训练来分散注意力的机器。（编者注：显然以前的电路板状态被用作网络的输入，所以看来这种缺乏记忆的观察是不正确的）。

Finally, there is a question about the applicability of a turn based game to the real world. Interactions in the real world are more dynamic and continuous, furthermore the time of interaction is unbounded. Go games have a limited number of moves. Perhaps, it doesn’t matter, after all, all interactions require two parties that act and react and predicting the future will always be boxed in time.  
最后，还有一个关于回合制游戏在现实世界中的适用性的问题。现实世界中的交互是动态的、连续的，而且交互的时间是无限的。围棋游戏的动作有限。也许，这并不重要，毕竟，所有的互动都需要双方的行动和反应，预测未来总是会在时间上受到限制。

If I were to pinpoint the one pragmatic Deep Learning discovery in AlphaGo Zero then it would be the fact that Policy Iteration works surprisingly well using Deep Learning networks. We’ve have hints in previous research that incremental learning was a capability that existed. However, DeepMind has shown unequivocally that incremental learning indeed works effectively well.  
如果我要在AlphaGo Zero中精确指出一个实用的深度学习发现，那就是策略迭代在使用深度学习网络时非常有效。我们在先前的研究中已经发现，增量学习是一种存在的能力。然而，DeepMind明确地表明，渐进式学习确实有效。

AlphaGo Zero appears also to have evolutionary aspects. That is, you select the best version of the newly latest trained network and you discard the previous one. There is indeed something going on here that is eluding a good explanation. The self-play is intrinsically competitive and the MCTS mechanism is an exploratory search mechanism. Without exploration, the system will eventually not be able to beat itself in play. To be effective, the system should be inclined to seek out novel strategies to avoid any stalemate. Like nature’s own evolutionary process that abhors a vacuum, AGZ seems to discover unexplored areas and somehow take advantage of these finds.  
AlphaGo Zero似乎也有进化的方面。也就是说，选择最新训练网络的最佳版本，然后放弃前一个版本。这里确实发生了一些事情，无法得到很好的解释。自我游戏具有内在的竞争性，MCTS机制是一种探索性的搜索机制。如果没有探索，这个系统最终将无法在游戏中击败自己。为了有效，该体系应倾向于寻找新的战略，以避免任何僵局。就像自然自身的进化过程中厌恶真空一样，AGZ似乎发现了未探索的区域，并以某种方式利用了这些发现。

One perspective to think about these systems as well as the human mind is in terms of the language that we use. Language is something that you layer more and more complex concepts on top of each other. In the case of AlphaGo Zero, it learned a new language that doesn’t have legacy baggage and it learned one that is so advanced that it is incomprehensible. Not necessarily mutually exclusive. As humans, we understand the world with concepts that originate from our embodiment with our world. That is we have evolved to understand visual-spatial, sequence, rhythm and motion. All our understanding is derived from these basic primitives. However, a machine may possibly discover a concept that may simply not be decomposable to these basic primitives.  
思考这些系统和人类思维的一个视角是我们使用的语言。语言是一种你把越来越复杂的概念层层叠加在一起的东西。在AlphaGo Zero的例子中，它学习了一种新的语言，它没有遗留的包袱，它学习的语言是如此先进，以至于无法理解。不一定互相排斥。作为人类，我们用源于我们与世界的化身的概念来理解世界。也就是说，我们已经进化到理解视觉空间、序列、节奏和运动。我们所有的理解都源于这些基本的原语。然而，机器可能会发现一个概念，这个概念可能根本无法分解为这些基本原语。

Such irony, when DeepMind trained an AI without human bias, humans discovered they didn’t understand It! This in another dimension of incomprehensibility. The concept of “incomprehensibility in the large” in that there is just too much information. Perhaps there is this other concept, that is “incomprehensibility in the small”. That there are primitive concepts that we simply are incapable of understanding. Let this one percolate in your mind for a while. For indeed it is one that is fundamentally shocking and a majority will overlook what DeepMind may have actually uncovered!.