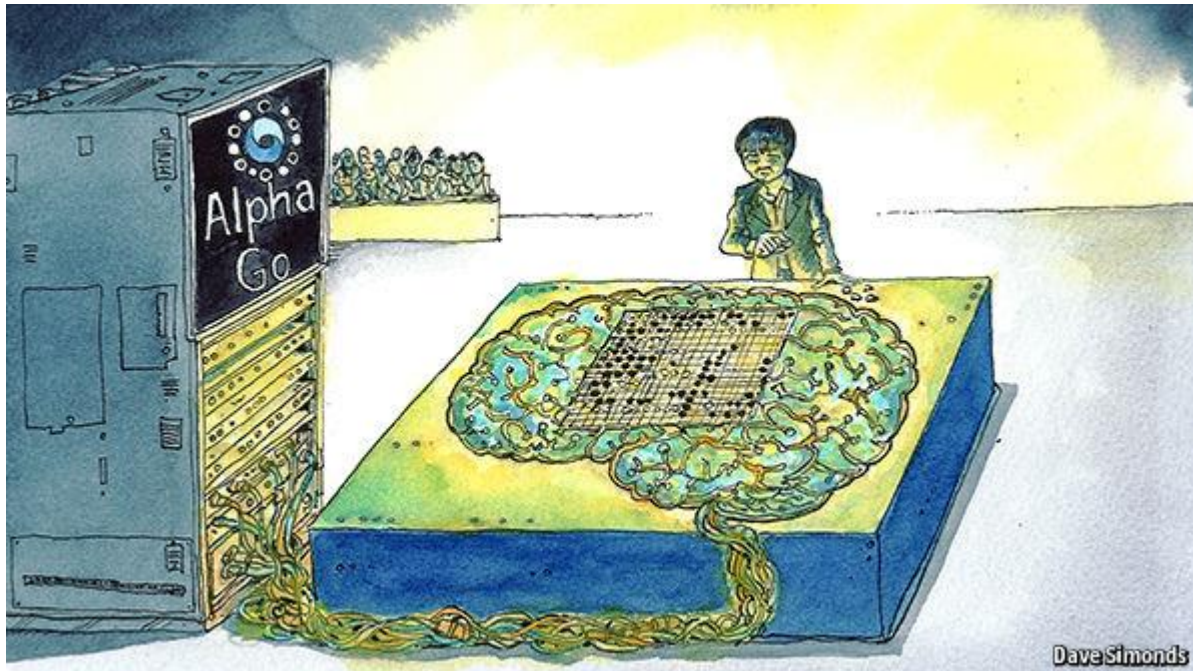


Artificial intelligence and Go

Showdown

Win or lose, a computer program's contest against a professional Go player is another milestone in AI
Mar 12th 2016 | SEOUL | From the print edition



UPDATE Mar 12th 2016: AlphaGo has won the third game against Lee Sedol, and has thus won the five-game match.

TWO : NIL to the computer. That was the score, as *The Economist* went to press, in the latest round of the battle between artificial intelligence (AI) and the naturally evolved sort. The field of honour is a Go board in Seoul, South Korea—a country that cedes to no one, least of all its neighbour Japan, the title of most Go-crazy place on the planet. To the chagrin of many Japanese, who think of Go as theirs in the same way that the English think of cricket, the game's best player is generally reckoned to be Lee Sedol, a South Korean. But not, perhaps, for much longer. Mr Lee is in the middle of a five-game series with AlphaGo, a computer program written by researchers at DeepMind, an AI software house in London that was bought by Google in 2014. And, though this is not an official championship series, as the scoreline shows, Mr Lee is losing.

In this section

- **Showdown**
- Without fire?

- Crucible
- Cool beans

Reprints

Go is an ancient game—invented, legend has it, by the mythical First Emperor of China, for the instruction of his son. It is played all over East Asia, where it occupies roughly the same position as chess does in the West. It is popular with computer scientists, too. For AI researchers in particular, the idea of cracking Go has become an obsession. Other games have fallen over the years—most notably when, in 1997, one of the best chess players in history, Garry Kasparov, lost to a machine called Deep Blue. Modern chess programs are better than any human. But compared with Go, teaching chess to computers is a doddle.

At first sight, this is odd. The rules of Go are simple and minimal. The players are Black and White, each provided with a bowl of stones of the appropriate colour. Black starts. Players take turns to place a stone on any unoccupied intersection of a 19x19 grid of vertical and horizontal lines. The aim is to use the stones to claim territory. In the version being played by Mr Lee and AlphaGo each stone, and each surrounded intersection, is a point towards the final score. Stones surrounded by enemy stones are captured and removed. If an infinite loop of capture and recapture, known as Ko, becomes possible, a player is not allowed to recapture immediately, but must first play elsewhere. Play carries on until neither player wishes to continue.

Go forth and multiply

This simplicity, though, is deceptive. In a truly simple game, like noughts and crosses, every possible outcome, all the way to the end of a game, can be calculated. This brute-force approach means a computer can always work out which move is the best in a given situation. The most complex game to be “solved” this way is draughts, in which around 10^{20} (a hundred billion billion) different matches are possible. In 2007, after 18 years of effort, researchers announced that they had come up with a provably optimum strategy.

But a draughts board is only 8x8. A Go board’s size means that the number of games that can be played on it is enormous: a rough-and-ready guess gives around 10^{170} . Analogies fail when trying to describe such a number. It is nearly a hundred of orders of magnitude more than the number of atoms in the observable universe, which is somewhere in the region of 10^{80} . Any one of Go’s hundreds of turns has about 250 possible legal moves, a number called the branching factor. Choosing any of those will throw up another 250 possible moves, and so on until the game ends. As Demis Hassabis, one of DeepMind’s founders, observes, all this means that Go is impervious to attack by mathematical brute force.

But there is more to the game’s difficulty than that. Though the small board and comparatively restrictive rules of chess mean there are only around 10^{47} different possible games, and its branching factor is only 35, that does, in practice, mean chess is also unsolvable in the way that draughts has been solved. Instead, chess programs filter their options as they go along, selecting promising-

looking moves and reserving their number-crunching prowess for the simulation of the thousands of outcomes that flow from those chosen few. This is possible because chess has some built-in structure that helps a program understand whether or not a given position is a good one. A knight is generally worth more than a pawn, for instance; a queen is worth more than either. (The standard values are three, one and nine respectively.)

Working out who is winning in Go is much harder, says Dr Hassabis. A stone's value comes only from its location relative to the other stones on the board, which changes with every move. At the same time, small tactical decisions can have, as every Go player knows, huge strategic consequences later on. There is plenty of structure—Go players talk of features such as ladders, walls and false eyes—but these emerge organically from the rules, rather than being prescribed by them.

Since good players routinely beat bad ones, there are plainly strategies for doing well. But even the best players struggle to describe exactly what they are doing, says Miles Brundage, an AI researcher at Arizona State University. "Professional Go players talk a lot about general principles, or even intuition," he says, "whereas if you talk to professional chess players they can often do a much better job of explaining exactly why they made a specific move." Intuition is all very well. But it is not much use when it comes to the hyper-literal job of programming a computer. Before AlphaGo came along, the best programs played at the level of a skilled amateur.

Go figure

AlphaGo uses some of the same technologies as those older programs. But its big idea is to combine them with new approaches that try to get the computer to develop its own intuition about how to play—to discover for itself the rules that human players understand but cannot explain. It does that using a technique called deep learning, which lets computers work out, by repeatedly applying complicated statistics, how to extract general rules from masses of noisy data.

Deep learning requires two things: plenty of processing grunt and plenty of data to learn from. DeepMind trained its machine on a sample of 30m Go positions culled from online servers where amateurs and professionals gather to play. And by having AlphaGo play against another, slightly tweaked version of itself, more training data can be generated quickly.

Those data are fed into two deep-learning algorithms. One, called the policy network, is trained to imitate human play. After watching millions of games, it has learned to extract features, principles and rules of thumb. Its job during a game is to look at the board's state and generate a handful of promising-looking moves for the second algorithm to consider.

This algorithm, called the value network, evaluates how strong a move is. The machine plays out the suggestions of the policy network, making moves and countermoves for the thousands of possible daughter games those suggestions could give rise to. Because Go is so complex, playing all

conceivable games through to the end is impossible. Instead, the value network looks at the likely state of the board several moves ahead and compares those states with examples it has seen before. The idea is to find the board state that looks, statistically speaking, most like the sorts of board states that have led to wins in the past. Together, the policy and value networks embody the Go-playing wisdom that human players accumulate over years of practice.



As Mr Brundage points out, brute force has not been banished entirely from DeepMind's approach. Like many deep-learning systems, AlphaGo's performance improves, at least up to a point, as more processing power is thrown at it. The version playing against Mr Lee uses 1,920 standard processor chips and 280 special ones developed originally to produce graphics for video games—a particularly demanding task. At least part of the reason AlphaGo is so far ahead of the competition, says Mr Brundage, is that it runs on this more potent hardware. He also points out that there are still one or two hand-crafted features lurking in the code. These give the machine direct hints about what to do, rather than letting it work things out for itself. Nevertheless, he says, AlphaGo's self-taught approach is much closer to the way people play Go than Deep Blue's is to the way they play chess.

One reason for the commercial and academic excitement around deep learning is that it has broad applications. The techniques employed in AlphaGo can be used to teach computers to recognise faces, translate between languages, show relevant advertisements to internet users or hunt for subatomic particles in data from atom-smashers. Deep learning is thus a booming business. It

powers the increasingly effective image- and voice-recognition abilities of computers, and firms such as Google, Facebook and Baidu are throwing money at it.

Deep learning is also, in Dr Hassabis's view, essential to the quest to build a general artificial intelligence—in other words, one that displays the same sort of broad, fluid intelligence as a human being. A previous DeepMind paper, published in 2015, described how a computer had taught itself to play 49 classic Atari videogames—from “Space Invaders” to “Breakout”—simply by watching the screen, with no helpful hints (or even basic instructions) from its human overlords. It ended up doing much better than any human player can. (In a nice coincidence, *atari* is also the name in Go for a stone or group of stones that is in peril of being captured.)

Games offer a convenient way to measure progress towards this general intelligence. Board games such as Go can be ranked in order of mathematical complexity. Video games span a range of difficulties, too. Space Invaders is a simple game, played on a low-resolution screen; for a computer to learn to play a modern video game would require it to interpret a picture much more subtle and complicated than some ugly-looking monsters descending a screen, and in pursuit of much less obvious goals than merely zapping them. One of DeepMind's next objectives, Dr Hassabis says, is to build a machine that can learn to play any game of cards simply by watching videos of humans doing so.

Go tell the Spartans

For now, he reckons, general-purpose machine intelligence remains a long way off. The pattern-recognising abilities of deep-learning algorithms are impressive, but computers still lack many of the mental tools that humans take for granted. A big one is “transfer learning”, which is what AI researchers call reasoning by analogy. This is the ability to take lessons learned in one domain and apply them to another. And machines like AlphaGo have no goals, and no more awareness of their own existence than does a word processor or a piece of accounting software.

In the short term, though, Dr Hassabis is optimistic. At a *kiwon*, or Go parlour, in Seoul, the day before the match, the 30 or so players present were almost unanimous in believing that the machine would fall short. “Lee is a genius who is constantly creating new moves; what machine can replicate that?” asked one. At a pre-match press conference Mr Lee said he was confident he would win 5-0, or perhaps 4-1.

He was, plainly, wrong about that, although it is not over yet. “He's a very good player,” said a diplomatic Dr Hassabis before the match. “But our internal tests say something different.” Even if Mr Lee does manage to pull off an improbable victory, though, humans are unlikely to stay on top for long. As AlphaGo's algorithms are tweaked, and as it gathers more data from which to learn, it is only going to get better. Asked whether there was a ceiling to its abilities, Dr Hassabis said he did not know: “If there is, we haven't found it yet.”

Correction: *An earlier version of this story suggested that 10^{170} was the number of possible positions of stones on a Go board; in fact it is an estimate of the number of possible Go games. Sorry*