MATTHEW SADLER & NATASHA REGAN

GAME CHANGER

NEW IN CHESS

AlphaZero's Groundbreaking Chess Strategies and the Promise of AI

With a foreword by Garry Kasparov

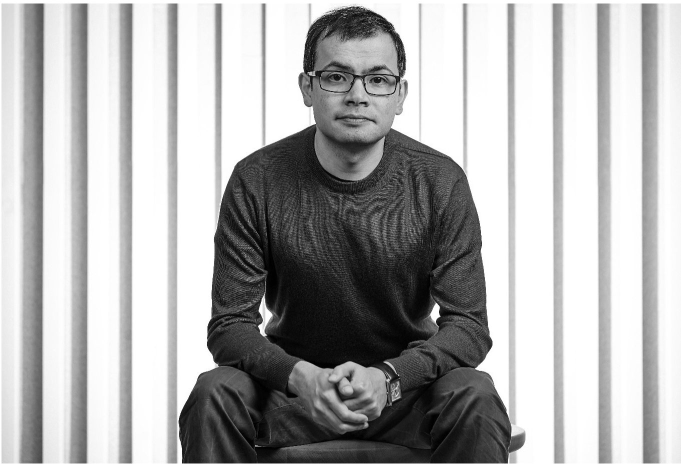
Introduction by DeepMind CEO Demis Hassabis

CHAPTER 3

Demis Hassabis, DeepMind and AI

DeepMind was set up to solve intelligence and use it to solve everything else. Spending time with the DeepMind team, the authors were struck by the depth and diversity of challenges being met by the company. We both first knew DeepMind CEO Demis Hassabis as an up-and-coming chess talent in the English junior chess circuit, and as a frequent medal winner in London's Mind Sports Olympiad, an international festival with over 60 different board game competitions. That early experience would ultimately prove the cornerstone of one of modern science's most fascinating careers.

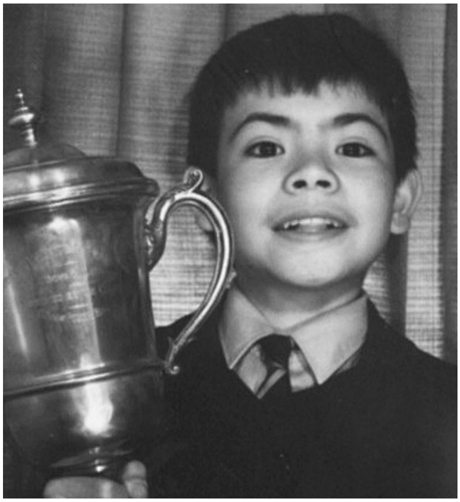
After a degree in computer science from Cambridge, and several successful ventures in computer games development – including helping program and design the best-selling title Theme Park and setting up the developer Elixir Studios – Demis went on to complete a PhD in neuroscience at UCL and conduct research at top labs including at MIT and Harvard. In 2010 he founded DeepMind, having acquired the relevant expertise in neuroscience, computing and business.



Demis Hassabis, CEO of DeepMind.

Here Demis tells us about his unique journey, as well as the origins of AlphaZero, the thinking behind it, and how, one day, it might be used to assist humanity in making crucial scientific discoveries.

When did you start down the path that would lead you to becoming CEO of DeepMind?
Thinking back, chess is a core part of my identity in many ways. Like with a lot of chess players who started young (I was about four years old), chess became a key element of the way I think and approach problems. I was quite an introspective kid who spent a lot of time trying to improve my chess, like all junior players. I liked the competitive aspect from winning tournaments, but the most satisfying thing was measuring your own self-improvement and seeing how far you can push yourself to reach your true potential.
But I was also spending a lot of time reflecting on what my brain was doing. During a game I would often wonder, 'How's my brain doing this? How's it coming up with these plans at that moment? What is this process of thinking?' That got me interested in the mind, the muscle we were using to play chess, how it works and how to improve it. I really believe you can try and understand this process mechanistically.
Then around the age of eight I got my first computer, a ZX Spectrum 48k, and I loved it from the moment I unwrapped the box. Even that was indirectly influenced by chess. I can't remember any of my friends having computers at that time and my parents are complete technophobes so I don't know where I got the idea. But I decided I'd like a computer and my parents couldn't object because I used my winnings from an under-10s chess tournament to buy it – it cost about a hundred pounds. I bought some programming books as well and just started playing and modifying games that came with the computer.



When did your interest switch to computers?
It was already starting to switch quite quickly. At that time, I was equally obsessed with chess and computers. I was teaching myself how to program from books.
In those days you could readily access the code to a game to start tinkering with – giving yourself extra lives, changing the sprites, things like that – and before you knew it, you had a different game. From there it's a small leap to creating your own games from scratch.

What were your first steps with AI?

I started my journey into AI when I was about 12. I bought a Commodore Amiga 500, which was an amazing machine that you could write a lot more complex and demanding programs on. I got one of my first books on AI, the *Computer Chess Handbook* by David Levy, which explained concepts like alpha-beta search and evaluation functions, and I used the ideas in it to program my Amiga to play Othello[4]. I tested it on my kid brother, George, and it managed to beat him. Admittedly he must have only been about 6 at the time! But it was still a huge thrill and it started me thinking about both the potential of AI and using games as a testing ground for it.

As I got progressively more drawn into computer games programming, my desire to become a professional chess player diminished. I loved playing chess and I still do, but I felt it would have been too narrow a pursuit to spend my entire career on. There are so many exciting things in life to discover, learn about and master! Even just in the domain of games there are so many brilliant ones with ingenious game designs and mechanics. A lot of chess player friends I know only like chess, whereas I've always liked a whole range of games from board games such as Go, shogi, Diplomacy, poker, Settlers of Catan, to computer games like Civilization and Starcraft… I'm yet to see a good game I didn't enjoy!

Then at 16 I got my first job as a professional programmer at Bullfrog Productions, which I would say was the number one games development house in Europe at the time. Theme Park was my first big game and it became a no.1 best-selling title and a huge commercial success. Among other things, I wrote the AI that ran the simulation and characters. The idea behind the game was that you designed and built a complete amusement park and then thousands of little people would come to play on the rides. If they enjoyed themselves they would 'tell their friends' and that would result in more visitors and revenue, which you could then use to buy bigger and better rides. You played how you liked and the game would adapt to the way you played. And it was sort of magical because the game experience would be different and individual for every player even though it was the same program. And that already really struck me as something quite interesting and powerful about emergent simulations and AI.

How did you plan your career?

My time at Bullfrog was quite formative, there was a very

4 Called Reversi in the U.S.

interesting set of talented people working there from very diverse backgrounds, led by the mercurial and world-famous game designer Peter Molyneux.

At the same time I was voraciously reading lots of books like Douglas Hofstadter's *Gödel, Escher, Bach*, Isaac Asimov's *Foundation* series, and Iain Banks' *Culture* series, all of which had a big influence on me. It was during this time that I decided I was going to dedicate my career to working on AI; it was the most important and interesting thing I could possibly imagine working on, and I already had the kernel of the idea for what eventually became DeepMind.

The rest of my early career was then about collecting the right knowledge and experience to be ready to run an ambitious project like DeepMind. That led to my degree in Computer Science at Cambridge, and after graduation I founded my own games company, Elixir Studios, which gave me invaluable experience at a young age of how to run large engineering teams, manage businesses, raise money and work with big publishers. Following that, I did a PhD in neuroscience to better understand and get inspiration from how our own brains work and solve some of the problems we wanted our AIs to be able to do, connecting back to the thoughts that fascinated me as a kid playing chess.

What was your PhD about?

I worked on a small but crucial part of the brain called the hippocampus. We've known since the 1950s that it is critical for episodic memory – the type of memory that helps us recall events in our everyday lives. But I decided to investigate whether it was also involved in supporting imagination and planning for the future. It turned out that – surprisingly – it was, and this ended up becoming a major finding for the hippocampus and memory field, as well as opening up imagination (or 'mental simulation') as a legitimate scientific topic of neuroscience study.

Our work demonstrated a systematic connection between the reconstructive process of memory recall, and the constructive process of imagination, and the fact that they both rely on the same underlying mechanisms and are dependent on the fast binding capabilities of the hippocampus.

I decided to study memory and imagination because they seemed like two key components of intelligence, and yet we had no idea how to build and integrate those capabilities into our AI systems at the time.

In addition to learning about the brain and using it as a source of inspiration for new types of algorithms, I was also learning about the scientific method in practice: how to come up with and test your hypotheses, how to design good experiments with proper controls so one can draw valid conclusions, how to engage with the scientific discourse, review the literature, publish papers and present at conferences. If you're going to manage the complex research programmes that we have here at DeepMind, you have to understand all those processes fully.

I always try to do things for multiple reasons: life's so short, ideally you want to do activities that have more than one purpose. My PhD is a good example, you're learning about neuroscience to understand more about the mind, while also learning about how to conduct good research.

And throughout all of this time, I was making a mental note of all the amazing people I met who might have useful skills for, and be interested in, an AI adventure one day – neuroscientists, engineers, game designers – and we brought them all back together for DeepMind.

Was developing a strong chess computer something you thought about a lot in your career?

Early on in my career I actually thought that chess was sort of 'done' because of Deep Blue's achievements in the late 90s. But the game of Go was something I thought about a lot. It presented a unique challenge. We could see that the method that worked for chess – hardcoded heuristics combined with brute force search – just wouldn't work for Go because of its incredible complexity and highly esoteric nature.

Dave Silver (the lead researcher on AlphaGo) and I used to discuss this a lot back when we were making games at Elixir (in fact we often played Go in the evenings there!). The possibilities excited us; we could see the limitations of the hand-coded brute force methods, and we felt that cracking Go would necessarily involve having to invent some truly interesting and novel algorithms that might also be a step towards flexible and general AI.

In the end it would be another 20 years – and long research journeys respectively – before finally we felt like we were ready to tackle such an immense challenge. But it was always there, in the back of our minds, gestating, waiting for the right time and right combination of ideas.

How did you set up your Go playing project?

After DeepMind's acquisition by Google in early 2014, I gave a talk about DQN – a system that could learn to play any Atari game just from observing the pixels on the screen – to Google's executive team. One of them asked me afterwards if we could apply these techniques to Go and how long it would take to beat a world champion. Without thinking, I said it might take two years.

In a way, saying that out loud served as the final impetus to actually begin the project that would become known as AlphaGo – the first computer program to defeat a professional human Go player.

When I got back, I talked to Dave and some of the other members of the team about what I'd said. There was a collective feeling that maybe now, nearly 20 years after we had first dreamed about it, we finally had the right ideas to tackle this incredible challenge. That was when we decided it was the right time to show the world that it was in fact possible to build a system that could learn how to play and master Go.

Initially we started the project very small, with just Dave, Aja Huang, lead programmer and a computer Go expert, and a single intern, Chris Maddison, working on it. To begin with we just wanted to establish whether it was even possible to train a neural network – later called the 'policy net' – to predict what a strong player's next move might be in any given board position.

In late 2014, after around 6 months' work, we got our first big result. The network was guessing the move of a professional player correctly with around 55% accuracy, so it was far from perfect but very promising. This was very exciting as it gave us our first empirical evidence that a neural net approach could really maybe work for Go. And we knew, from hard experience acquired through building these types of self-learning systems many times, that once you have a foothold like this, usually you can rapidly scale and improve them through careful optimisation, better algorithms, more training and more compute power.

The game of Go

Go is a beautiful and elegant strategy game that originated in China around 3,000 years ago. Players take turns to place black or white stones on a 19x19 board in an effort to surround more territory than their opponent. Despite the simple rules, Go is incredibly complex, with more than 10 to the power of 170 (1 followed by 170 zeroes) possible board configurations, more than there are atoms in the universe. Go players are ranked using kyu and *dan* grades, the same system as used in martial arts. Beginners progress through kyu grades, which decrease as playing level increases until a player achieves 1 kyu. These are followed by the amateur dan grades, which usually increase from 1 dan to 6 or 7 dan. Professional Go players have a separate ranking system that runs from 1 dan to 9 dan.

What did the AlphaGo project team do?
After that result we really accelerated things. Around another 15 people joined the project, brilliant specialists with a wide array of skills, and some of the best in the world at what they do, brought together in what we term internally a 'Strike Team'. This involves cross-cutting reporting lines, so that each person reports out of their normal teams together especially for this one project.
It's like launching a product even though it's still research. The momentum once a project like this gets going is incredible, and it was a really amazingly creative period. There are milestones and you have leaderboards to keep track of how good the estimated strength of the program (Elo rating) is at every step of the way.
The first milestone was beating the current top handcrafted Go programs: Zen and Crazy Stone. Both programs were strong amateur level (around 5 dan amateur), but neither were at pro level. In fact, famously, no machine had ever beaten a pro at Go. The last big advance in computer Go had been a decade prior, with the use of Monte Carlo tree search rather than alpha-beta search, but they still used handcrafted evaluation functions and had only been improving very slowly over the past few years. Our next innovation was creating a second neural network, called the 'Value Net', that learns through experience to estimate the probability of winning from the current position. We put the Policy Net and Value Net together with Monte Carlo tree search to create our first fully-functional Go program.
To our pleasant surprise we reached our first milestone just 6 months after our initial results, so we had the world's strongest Go program at that point. AlphaGo continued to rapidly improve with more training and soon it was able to beat Aja, who is about 6 dan amateur himself. This was a particularly poignant moment for Aja personally because he had achieved his lifelong dream of writing a Go program that could beat him!
We were now ready for our second milestone, which was to take on and be the first program to beat a professional player. We emailed the 3-times European Champion Fan Hui, who turned out to be a truly wonderful and warm human being, and eventually became an advisor to the team. He was based in France so he could get over here easily and we could try playing him first behind closed doors.
We won that match 5-0 and with that became the first program to ever beat a professional at Go. As you can imagine that was an awesome moment, one that would go down in history, and off the back of that we submitted a paper to the scientific journal *Nature*, chronicling our results so far.

We then turned our attention to the third and final step, attempting to beat a World Champion. We wanted someone who had been established at the top of the game for a long time, but was still at their peak. In the end we decided to challenge the legendary 9-dan South Korean grandmaster Lee Sedol, winner of 18 world titles, and acknowledged as the greatest player of the past decade.

We wanted to announce the match at the same time as the publication of the Nature paper, but this was a big risk as at the time of making the announcement, we knew that AlphaGo was still considerably weaker than Lee Sedol, and there would only be about three months to go. However, the performance graph was rising inexorably higher, and unless it was to asymptote for some reason, we predicted it would cross-over to be stronger than Lee Sedol before the match.

History shows that its rise continued, but how did you rate AlphaGo's chances going into that match?

Lee Sedol was pretty confident, and predicted a 5-0 or 4-1 victory for him, because he had seen the Fan Hui games that we published in Nature. In the documentary – *AlphaGo* – that was made about the matches, he says that he believed that human intuition was still 'too advanced for AI to have caught up'. At that point AlphaGo was better than a 2-dan pro, but it was nowhere close to his strength. Fan Hui is a top-500 player. Lee Sedol is a 9-dan pro, and one of the greatest players of all time, he would have thrashed the version of AlphaGo we played against Fan Hui. I guess he quite reasonably thought: 'how much could it really have improved given it's only three to four months later?' Perhaps a few levels, but surely not 7 dan rankings!

For our part we were quietly confident but also nervous. By the time of the match, our internal tests were telling us that the program should be stronger than Lee Sedol overall, but there was a high degree of uncertainty. AlphaGo was evaluated by testing it against earlier versions of itself, so in machine learning terms it could have overfit, in other words it could have learnt to beat itself very efficiently, but somehow that capability would maybe not generalise to a totally new opponent, especially someone who was famed for his creativity and fighting spirit.

Even more worryingly we also knew that AlphaGo suffered from what we dubbed 'delusion' problems. We didn't know exactly why, but in certain highly complex fighting situations, where precise timing was important in a very long sequence of moves, sometimes the system would misevaluate the position, and thus would incorrectly assess the position as being good for it, when in fact it was totally losing.

We tried dozens of things to fix the problem, but we were not able to fix it in time for the match (we did manage to resolve this problem in later versions of AlphaGo). Our tests showed this type of position would occur at a frequency of roughly one time in every five games, and so based on this we were expecting a 4-1 victory. Incredibly, that's exactly what happened, with Lee Sedol managing to win game four by playing a genius move that was so unexpected it triggered one of these misevaluations.

And so we had won this historic match, stunning AI and Go experts, with many proclaiming that the achievement was 'a decade ahead of its time'. Of course winning was the main aim, but in fact the most important thing was the *way* AlphaGo won. During the match AlphaGo played many highly original, creative and beautiful moves, most famously move 37 in game 2, which in many cases overturned centuries of received wisdom. It wasn't just regurgitating or copying human knowledge. Subsequently Fan Hui and our many friends in the Go world have told us this has revolutionised how the game is played, and many books have now been written about AlphaGo's unique playing style.

It sounds like AlphaGo's development went reasonably smoothly?

Progress often looks smooth in retrospect, but at the time the outcome of each step was totally uncertain. A few months into the project I remember mentioning to Aja that we should be aiming one day to take on a 9-dan professional, and he thought I'd completely lost my mind! But that's how it always is with any truly cutting-edge research: if you know for sure how a branch of research is going to go, then it isn't really research. That's what is so exciting about scientific research, every day you wake up and take a step into the unknown.

Has your approach to AI changed as you developed AlphaGo and AlphaZero?

It's very interesting if you stand back and look at the whole body of work. First, we started with AlphaGo and we were trying to beat this game that was thought uncrackable for AI. Then, once we had achieved that, we tried to make the system progressively more efficient and general.

The original AlphaGo was initially trained using hundreds of thousands of human amateur games to help it develop an understanding of what reasonable human play looks like. We also built in a very small amount of Go-specific information, not major things like rules or heuristics, but very high-level things like the board has four-way symmetry.

AlphaGo Zero was the next stage in its evolution, which you can think of as AlphaGo but without any of those crutches. We removed any domain knowledge and built a system that learns how to play the game itself, starting from completely random play with no human games as input. Amazingly, that also worked, so the next stage was to build AlphaZero, which also learns from random self-play but can generalise beyond Go to all sorts of two-player perfect information games[5], including of course chess!

When you decided to build AlphaZero – a more general system than AlphaGo – was it that you had built up more confidence in how to do it? Or had you discovered some new techniques along the way?
It's both of those things. This is why it's so much easier to follow than to innovate. Once you land on the moon, then other people can land on the moon. It's similar in science: things never work the first time. The question is then: should I push on this harder, or is this a brick wall? If someone has done it before then it's just a question of pain and effort and the will to do it, because you already know that it is definitely possible.
Conversely, when something is unknown, it's not just a question of will. Because sometimes, the right thing to do is to stop and do something else. If you just will it, you could end up doing the wrong thing for 30 years and getting nowhere.
Without belief and perseverance you can't achieve anything in science, but you also need to know when you are going in the wrong direction. That is something that I learned through my earlier career experiences. It's a very hard thing to teach. It's like a smell you get, a taste. They often say that about scientists: the best scientists have really good taste. What they mean by that is they know how to hone into the right problem and what the right complexity of problem is to tackle next. They know the right question to ask.

Your games playing programs are known for having a creative and attacking style. What kind of games player are you?
I'm more of an all-round games player these days. Familiarity with many games means there are strategies and motifs you recognise and then you start seeing all the connections between the different games, which can help. It also means that you can learn new games very quickly.
In most games, I generally have quite a controlled and calm style, and I always like to have something in reserve in case something unexpected happens!

5 Perfect information games are those where any player is perfectly informed of all the events that have previously occurred, including how the game started. Examples include Go, chess, shogi and Backgammon.

Can games help people with real life?
That's the other cool thing about games. You can think of them as a gym for the mind. In life, big decision moments happen only rarely and often they are extremely critical. Where do you get to practise your decision-making under pressure? In life you usually don't get a second chance. Games provide a safe training ground where you can get immediate feedback on your performance and then use that information to improve. Then you can draw on all that experience and training when it's time to face the real event.
I always used to joke that if I were to design an MBA course I'd design one around games where you get a world expert in each of the different major games and you learn the meta-skills trained by those games.

Can AlphaZero be used to teach humans?
There are certain things that machines do that you can't learn from. It depends on why they are better at doing them. Let's take chess computers as an example. The top programs are all stronger than the best human chess player. We can analyse as many of their games as we want, but we won't become as strong. The reason is that they are stronger because they calculate more lines and don't make tactical blunders. We, as human chess players, are not intentionally making tactical blunders, it's just that our brains aren't evolved to do that type of calculation. So those solutions are of limited help to us. However, what we are finding with AlphaZero is that some of the improvements or the advancements are actually strategic in nature, and that is something we can potentially incorporate in our own play. Go players already found that with AlphaGo and have incorporated some of its strategies into their own games. I think the same will happen with AlphaZero and chess. Maybe this book is the start of that!
Related to this, we are also trying to build analytical and visualisation tools that offer an insight into how these systems make their decisions and allow us to better understand what factors they are weighing up. Work in this area is still at a nascent stage but I think we will see huge advances in this, and therefore in our understanding of these systems, in the next few years.

Is it hard to analyse results (such as from AlphaZero) because the systems are so complicated?
It's very complicated, but certainly no more complicated than the brain. Probably substantially less complicated because these systems are still a lot smaller in terms of the number of neurons and connections.
We also have full access and control over every moment-by-moment

thing that the machine is doing, which we don't even have with brain imaging. So my argument is, our understanding should be at least as good as with the brain, and I would argue that we should be in a better position than we are with the brain, because we have all these extra controls over what the system is doing.

What are the main differences between the way that humans learn chess, and the way that AlphaZero learns?
People are able to apply abstract knowledge from various sources, including books, learning from teachers, or even watching AlphaZero games.
A person doesn't have to play millions of games to learn, but this raw experience is the only way that AlphaZero can learn. It can't be taught and it can't read books, it has to learn from first principles. But we are trying to build machines that are capable of learning concepts or abstract knowledge. Nobody in the world has cracked that yet, that's one of the next big challenges in AI.

Are the techniques you used for AlphaZero applicable in other areas?
Ultimately the whole point of building general learning systems like AlphaZero is so they can be applied in all sorts of ways to creating solutions for real world problems that will be of huge benefit to everyone in society. Games are a very convenient platform on which to test AI. In my opinion, they are the perfect proving ground for developing and testing AI algorithms, and that's why we love using them – from board games to computer games and virtual environments.
There are an almost limitless number of things that general AI could eventually be applied to, but my personal passion is to use these kinds of AI systems to help scientists make critical research breakthroughs and discoveries more quickly, in fields where we urgently need advances such as climate science, material science and drug discovery. I believe that machine learning and AI have got a huge part to play in accelerating science and we want to be at the forefront of that.

What are the opportunities for applying AI in other sectors?
There are so many opportunities for applying AI to certain sectors. Think of things like healthcare, logistics, energy, transport, education, insurance, robotics and many others.
I believe there are multiple multi-billion-dollar businesses to be built by combining existing sector expertise and optimizing it with what is now relatively off-the-shelf AI.
We've already had a lot of success applying these kinds of techniques ourselves. For example we used

ideas similar to AlphaZero to control the cooling systems in Google's huge data centres, saving a huge amount of the energy they use, which of course is very valuable commercially in terms of saving money, and is also great for the environment.

What tools are available for an enthusiastic amateur to get started in AI?
There's a big open-source community and you can freely download almost all the library tools that are built by the big companies and do quite impressive things out of the box. There are also good books and tons of great online courses.
If you are motivated enough and you have good maths and programming skills, then you can dive into that – you could try it in insurance, Natasha! There is plenty to be done – it's an incredibly exciting time!

If you could achieve one thing in AI, what would that be?
I think general AI, to which I have devoted my life's work, is going to be the most important technology humanity will ever invent.
There are so many problems in the world that remain intractable, from climate change to diseases such as Alzheimer's to macro-economic problems. Everywhere we look there are huge and complex challenges for society, and the speed at which we are able to solve these problems will affect the lives and well-being of billions of people. This is where I believe AI can – and will – help society in a profound way. AI is the meta-solution to all of these problems. General AI will be a tool that will act as a multiplier for human ingenuity, allowing us to rapidly discover new knowledge and make progress on these complex challenges at a rate that we have never seen before.