

COMPUTING

20 Years after Deep Blue: How AI Has Advanced Since Conquering Chess

IBM AI expert Murray Campbell reflects on the machine's long, bumpy road to victory over chess champ Garry Kasparov

By Larry Greenemeier on June 2, 2017



World Chess Champion Garry Kasparov (L) makes a move during his fourth game against the IBM Deep Blue

chess computer. Credit: Stan Honda Getty Images

Twenty years ago IBM's Deep Blue computer stunned the world by becoming the first machine to beat a reigning world chess champion in a six-game match. The supercomputer's success against an incredulous Garry Kasparov sparked controversy over how a machine had managed to outmaneuver a grand master, and incited accusations—by Kasparov and others—that the company had cheated its way to victory. The reality of what transpired in the months and years leading up to that fateful match in May 1997, however, was actually more evolutionary than revolutionary—a Rocky Balboa-like rise filled with intellectual sparring matches, painstaking progress and a defeat in Philadelphia that ultimately set the stage for a triumphant rematch.

Computer scientists had for decades viewed chess as a meter stick for artificial intelligence. Chess-playing calculators emerged in the late 1970s but it would be another decade before a team of Carnegie Mellon University graduate students built the first computer—called Deep Thought—to beat a grand master in a regular tournament game. This success was short-lived—later that same year, 1989, Kasparov beat Deep Thought handily in the two games. IBM was impressed enough with the C.M.U. team's technology to bring its researchers onboard to develop an early version of Deep Blue—Deep Thought's successor. The Deep Blue team lost again to Kasparov in 1996 at a tournament in Philadelphia but managed to win one game out of six against the world champ.

That seemingly small victory “was very important to us to show that we were on the right track,” says Deep Blue AI expert Murray

Campbell, now a distinguished research staff member in the AI Foundations group within IBM T. J. Watson Research Center's Cognitive Computing organization. "By the time of our final match in 1997, we had made enough improvements to the system based on our experience that we were able to win." *Scientific American* spoke with Campbell about computer scientists' long obsession with chess, how IBM was able to turn the tables on the reigning chess champ and the challenges that lie ahead for AI.

[An edited transcript of the interview follows.]

How did you first get involved in the Deep Blue project?

I was part of a group of graduate students at Carnegie Mellon University that IBM approached. I had had a long interest in computer chess and had even written a chess program as an undergraduate. At C.M.U. I was working on artificial intelligence more generally and not exactly on building a high-performance chess computer that could play against a world champion. But as a side project a number of us [including Feng-hsiung Hsu and Thomas Anantharaman] did develop the machine that became known as Deep Thought, which became the first program to defeat a grand master, a professional level player in a tournament.

IBM noticed the successes that we were having building this machine on a shoestring budget and thought it would be interesting to have a group of us join IBM Research [in late 1989] to develop the next generation of this machine, called Deep Blue. They wanted to know if there was something special about the very best chess players in the world that was beyond what computers were capable of for the foreseeable future. Our feeling was that it was within a few

years of being done, although other researchers thought it was still decades away.

What is it about chess that makes an especially interesting problem for a computer scientist?

Hundreds of millions of people around the world play chess. It's known as a game that requires strategy, foresight, logic—all sorts of qualities that make up human intelligence. So it makes sense to use chess as a measuring stick for the development of artificial intelligence.

When we look at a game like chess, we say, “Well, yes, of course computers do well because it's a well-defined game—the rules, the moves, the goals.” And it's a constrained problem where you know all the information. Still, in spite of all those simplifications you could say chess is an enormously complex game, and that's why it took us, as a field, 50 years of development to finally beat the world champion.

What was your role specifically on the Deep Blue team?

I was the AI expert. AI was quite different in 1989 and early 1990. The dominant part in those days was what we now called good old-fashioned AI, or symbolic AI, which was based less on machine learning. Certainly machine learning was a serious field in those days but nothing like what it is today, where we have massive data sets and large computers and very advanced algorithms to churn through the data and come up with models that can do some amazing things. When I started with IBM, AI was fairly primitive. We worked on algorithms for searching databases as well as logic, planning and reasoning, all of which we knew Deep Blue would need

in order to compete.

What were the most significant limitations on AI back then?

The hardware didn't really support building the kinds of large networks that have proven useful today in making big data models. And the data itself wasn't necessarily there to the extent that we needed it at that point. Any time you go back and look at the most popular computer systems from 20 or 25 years ago you're shocked at how you could get anything done on a system like that. But of course, we did—we didn't know what we were missing, I guess, because we had never experienced it.

As far as data, I don't think anybody had a clear idea back then that there was a big benefit. It wouldn't have paid to build a really large data set because in part the processing power wouldn't have been enough to use it anyway. So, we made do with much smaller data sets.

How useful was your own chess expertise in building Deep Blue?

Not as useful as you might think. I was able to, in the early stages, identify problems with the system and suggest approaches that I felt would be able to fix one problem without creating a host of other problems. That was probably good enough to get us to a certain point. Eventually, though, if you're going to be playing competitions there's a host of really game-specific knowledge you need to have. When we got closer to the point where we would actually be playing against a world champion we brought in grand masters—Joel Benjamin, in particular—to help us.

How did the grand masters help raise Deep Blue's game?

There were two parts to how they helped. One, in particular, was to help with the opening library, which every chess program uses in order to save time and make sure it gets into reasonable positions. Humans have been studying chess openings for centuries and developed their own favorite [moves]. The grand masters helped us choose a bunch of those to program into Deep Blue.

They also were, you could say, sparring partners for Deep Blue. They would play against the computer and try and pinpoint weaknesses of the system. And then we would sit around with them and with the rest of the Deep Blue team and try to articulate what that weakness actually was and if there was a way to address it. Sometimes, given the limitations we had—we were programming part of the computer's instructions directly onto a piece of hardware called a chess accelerator chip rather than writing software—there were some problems we couldn't easily fix. But often there was some way we could improve its ability to deal with a problem we had identified.

How did Deep Blue decide which moves to make?

Deep Blue was a hybrid. It had general-purpose supercomputer processors combined with these chess accelerator chips. We had software that ran on the supercomputer to carry out part of a chess computation and then hand off the more complex parts of a move to the accelerator, which would then calculate [possible moves and outcomes]. The supercomputer would take those values and eventually decide what route to take.

How did Deep Blue advance from 1996 to 1997 in order to beat Kasparov?

We did a couple of things. We more or less doubled the speed of the system by creating a new generation of hardware. And then we increased the chess knowledge of the system by adding features to the chess chip that enabled it to recognize different positions and made it more aware of chess concepts. Those chips could then search through a tree of possibilities to figure out the best move in a position. Part of the improvement between '96 and '97 is we detected more patterns in a chess position and could put values on them and therefore evaluate chess positions more accurately. The 1997 version of Deep Blue searched between 100 million and 200 million positions per second, depending on the type of position. The system could search to a depth of between six and eight pairs of moves—one white, one black—to a maximum of 20 or even more pairs in some situations. Still, *while we were confident that the 1997 Deep Blue was much better than the 1996 version, in my mind the most probable outcome of the match was a draw. Even going into the final game of the match, I was expecting a draw, and a likely rematch.*

Why didn't IBM grant Kasparov's request for a rematch after the 1997 competition?

We felt we had achieved our goal, to demonstrate that a computer could defeat the world chess champion in a match and that it was time to move on to other important research areas.

How has AI changed over the two decades since that match?

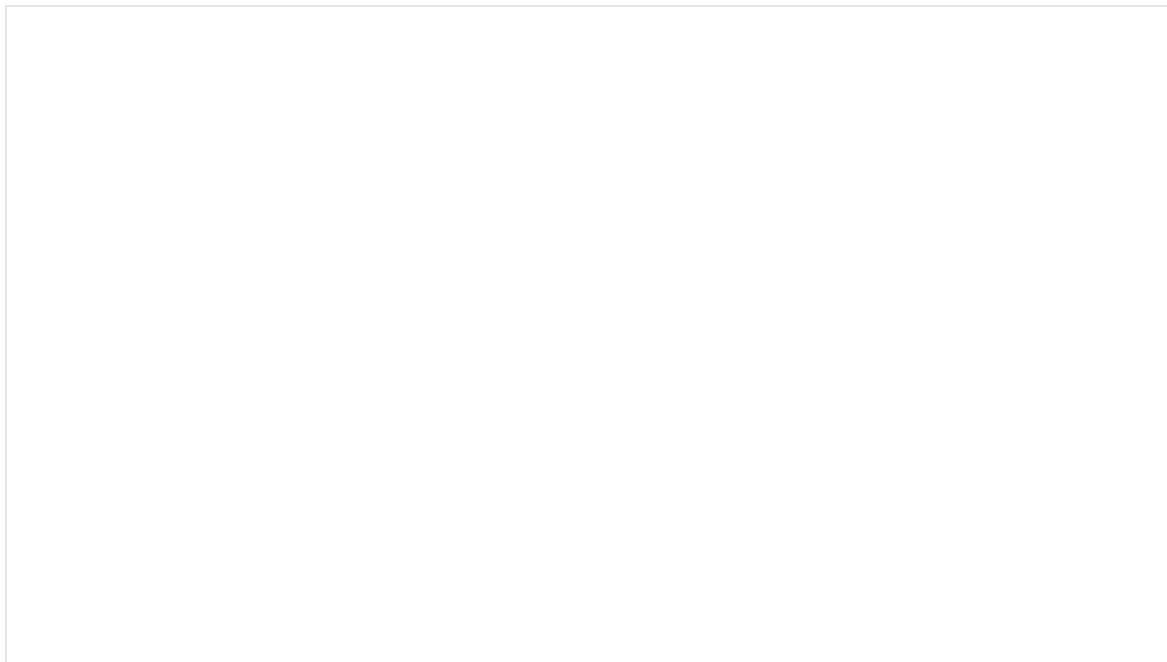
Of course, machines have improved in processing speed and memory and so on. People also started gathering—just as part of their business—a lot more data that provided fodder for the machine-learning algorithms of the day. Eventually we started realizing that combining all these things could produce some remarkable results. The IBM Watson system that played *Jeopardy!* used a machine-learning-based system that took a lot of data that existed in the world—things like Wikipedia and so on—and used that data to learn how to answer questions about the real world. Since then we have moved on to learn how to do certain kinds of perceptual tasks like speech recognition and machine vision. That has led to Watson performing more business-related tasks such as analyzing radiology images and sharing that information with physicians.

How did your experience working on Deep Blue help influence your work on AI going forward?

One thing in particular we learned is that there's more than one way to look at a complex problem. For example, in chess there's the human way, which is very pattern recognition–based and intuition-based, and then there's the machine way, which is very search intensive and looks through millions or billions of possibilities. Often these approaches are complementary. That's definitely true in chess but also in many real-world problems—that computers and humans together are better than either one alone. We wouldn't want, for example, computers to take over diagnosis and treatment of patients by themselves because there are a lot of intangibles in diagnosing a patient that are hard to capture in the data. But in terms of making recommendations about options to consider—perhaps those that are from very recent technical papers or clinical

trials that maybe the doctor isn't aware of—a system like that can be very valuable.

An important part of what we're doing right now is taking very advanced artificial neural network–based systems that tend to be very black box—they aren't particularly good at explaining why they're recommending what they're recommending—and giving them the capability to explain themselves. How can you really trust a recommendation coming out of system if it can't explain it? These black box neural network systems are enormously complex, with millions of parameters in them. Part of overcoming that [complexity] may be along the lines of training a system by giving it examples of good explanations. This is particularly obvious in the health care space when a computer makes a diagnosis or recommends a treatment. If there is a reasonable explanation, then we could probably more appropriately give it the weight it deserves to help a doctor make a final decision.



ADVERTISEMENT | REPORT AD

ABOUT THE AUTHOR(S)



Larry Greenemeier

Larry Greenemeier is the associate editor of technology for Scientific American, covering a variety of tech-related topics, including biotech, computers, military tech, nanotech and robots.

Credit: Nick Higgins

Scientific American is part of Springer Nature, which owns or has commercial relations with thousands of scientific publications (many of them can be found at www.springernature.com/us). Scientific American maintains a strict policy of editorial independence in reporting developments in science to our readers.

© 2017 SCIENTIFIC AMERICAN, A DIVISION OF NATURE AMERICA, INC.

ALL RIGHTS RESERVED.