Science | DOI:10.1145/2949662

Marina Krakovsky

Reinforcement Renaissance

The power of deep neural networks has sparked renewed interest in reinforcement learning, with applications to games, robotics, and beyond.

Each time DeepMind has announced an amazing accomplishment in game-playing computers in recent months, people have taken notice.

First, the Google-owned, London-based artificial intelligence (AI) research center wowed the world with a computer program that had taught itself to play nearly 50 different 1980s-era Atari games—from Pong and Breakout to Pac-Man, Space Invaders, Boxing, and more—using as input nothing but pixel positions and game scores, performing at or above the human level in more than half these varied games. Then, this January, DeepMind researchers impressed experts with a feat in the realm of strategy games: AlphaGo, their Go-playing program, beat the European champion in the ancient board game, which poses a much tougher AI challenge than chess. Less than two months later, AlphaGo scored an even greater victory: it won 4 games in a best-of-5 series against the best Go player in the world, surprising the champion himself.

IMAGE BY MAXUSER

The idea that a computer can learn to play such complex games from scratch and achieve a proficient level elicits gee-whiz reactions from the general public, and DeepMind's triumphs have heightened academic and commercial interest in the AI field behind DeepMind's methods: a blend of deep neural networks and reinforcement learning called "deep reinforcement learning."

"Reinforcement learning is a model of learning where you're not given a solution—you have to discover it by trial and error," explains Sridhar Mahadevan, a professor at the University of Massachusetts Amherst, a longtime center of research into reinforcement learning.

The clearest contrast is with supervised learning, the kind used to train image recognition software, in which the supervision comes in the form of labeled examples (and requires people to label them). Reinforcement learning, on the other hand, "is a way of not needing labels, or labeling automatically by who's winning or losing—by the rewards," explains University of Alberta computer scientist Rich Sutton, a co-founder of the field of reinforcement learning and co-author of the standard textbook on the subject. In reinforcement learning, the better your moves are, the more rewards you get, "so you can learn to play the Go game by playing the moves and winning or losing, and no one has to tell you if that was a good move or a bad move because you can figure it out for yourself; it led to a win, so it was a good move."

Sutton knows the process is not as simple as that. Even in the neat, tightly controlled world of a game, deducing which moves lead to a win is a notoriously difficult problem because of the delay between an action and its reward, a key feature of reinforcement learning. In many games, you receive no feedback at all until the end of the game, such as a 1 for a win or a −1 for a loss.

"Typically, you have to go through hundreds of actions before your score increases," explains Pieter Abbeel, an associate professor at the University of California, Berkeley, who applies deep reinforcement learning to robotics. (For a robot, a reward comes for completing a task, such as correctly assembling two Lego pieces.) "Before you understand how the game works, and are learning through your own trial and error," Abbeel says, "you just kind of do things, and every now and then your score goes up, and every now and then it goes down, or doesn't go up. How do you tease apart which subset of things that you did contributed to your score going up, and which subset was just kind of a waste of time?"

This thorny question—known as the credit assignment problem—remains a major challenge in reinforcement learning. "Reinforcement learning is unique in that it's the only machine learning field that's focused on solving the credit assignment problem, which I think is at the core of intelligence," says computer scientist Itamar Arel, a professor of electrical engineering and computer sciences at the University of Tennessee and CEO of Osaro, a San Francisco-based AI startup. "If something good happens now, can I think back to the last *n* steps and figure out what I did that led to the positive or negative outcome?"

Despite the buzz around DeepMind, combining reinforcement learning with neural networks is not new.

Just as for human players, figuring out smart moves enables the program to repeat that move the next time it faces the same situation—or to try a new, possibly better move in hope of stumbling into an even higher reward. In extremely small worlds (think of a simple game like Tic-Tac-Toe, also known as Noughts and Crosses), the same exact situations come up again and again, so the learning agent can store the best action for every possible situation in a lookup table. In complex games like chess and Go, however, it is impossible to enumerate all possible situations. Even checkers has so much branching and depth that the game yields a mind-boggling number of different positions. So imagine what happens when you move from games to real-world interactions.

"You're never going to see the same situation a second time in the real world," says Abbeel, "so instead of a table, you need something that understands when situations are similar to situations you've seen before." This is where deep learning comes in, because understanding similarity—being able to extract general features from many specific examples—is the great strength of deep neural networks. These networks, whose multiple layers learn relevant features at increasingly higher levels of abstraction, are currently the best available way to measure similarities between situations, Abbeel explains.

The two types of learning—reinforcement learning and deep learning through deep neural networks—complement each other beautifully, says Sutton. "Deep learning is the greatest thing since sliced bread, but it quickly becomes limited by the data," he explains. "If we can use reinforcement learning to automatically generate data, even if the data is more weakly labeled than having humans go in and label everything, there can be much more of it because we can generate it automatically, so these two together really fit well."

Despite the buzz around DeepMind, combining reinforcement learning with neural networks is not new. TD-Gammon, a backgammon-playing program developed by IBM's Gerald Tesauro in 1992, was a neural network that learned to play backgammon through reinforcement learning (the TD in the name stands for Temporal-Difference learning, still a dominant algorithm in reinforcement learning). "Back then, computers were 10,000 times slower per dollar, which meant you couldn't have very deep networks because those are harder to train," says Jürgen Schmidhuber, a professor of artificial intelligence at the University of Lugano in Switzerland who is known for seminal contributions to both neural networks and reinforcement learning. "Deep reinforcement learning is just a buzzword for traditional reinforcement learning combined with deeper neural networks," he says.

Schmidhuber also notes the technique's successes, though impressive, have so far been in narrow domains, in which the current input (such as the board position in Go or the current screen in Atari) tells you everything you need to know to guide your next move. However, this "Markov property" does not normally hold outside of the world of games. "In the real world, you see just a tiny fraction of the world through your sensors," Schmidhuber points out, speaking of both robots and humans. As humans, we complement our limited perceptions through selective memories of past observations; we also draw on decades of experience to combine existing knowledge and skills to solve new problems. "Our current reinforcement learners can do this in principle, but humans still do it much better," he says.

Researchers continue to push reinforcement learners' capabilities, and are already finding practical applications.

"Part of intelligence is knowing what to remember," says University of Michigan reinforcement learning expert Satinder Singh, who has used the world-building game Minecraft to test how machines can choose which details in their environment to look at, and how they can use those stored memories to behave better.

Singh and two colleagues recently co-founded Cogitai, a software company that aims to use deep reinforcement learning to "build machines that can learn from experience the way humans can learn from experience," Singh says. For example, devices like thermostats and refrigerators that are connected through the Internet of Things could continually get smarter and smarter by learning not only from their own past experiences, but also from the aggregate experiences of other connected devices, and as a result become increasingly better at taking the right action at the right time.

Osaro, Arel's software startup, also uses deep reinforcement learning, but promises to eliminate the costly initial part of the learning curve. For example, a computer learning to play the Atari game Pong from scratch starts out completely clueless, and therefore requires tens of thousands of plays to become proficient—whereas humans' experience with the physics of balls bouncing off walls and paddles makes Pong intuitive even to children.

"Deep reinforcement learning is a promising framework, but applying it from scratch is a bit problematic for real-world problems," Arel says. A factory assembling smartphones, for example, requires its robotic manufacturing equipment to get up to speed on a new design within days, not months. Osaro's solution is to show the learning agent what good performance looks like "so it gets a starting point far better than cluelessness," enabling the agent to rapidly improve its performance.

Even modest amounts of demonstration, Arel says, "give the agent a head start to make it practical to apply these ideas to robotics and other domains where acquiring experience is prohibitively expensive."

Further Reading

Mnih, V., et al.
Human-level control through deep reinforcement learning. *Nature* (2015), vol. 518, pp. 529–533.

Silver, D., et al.
Mastering the game Go with deep neural networks and tree search. *Nature* (2016), vol. 529, pp. 484–489.

Tesauro, G.
Temporal difference learning and TD-Gammon. *Communications of the ACM* (1995), vol. 38, issue 3, pp. 58–68.

Sutton, R.S., and Barto, A.G.
Reinforcement Learning: An Introduction. MIT Press, 1998, https://mitpress.mit.edu/books/reinforcement-learning

Based in San Francisco, **Marina Krakovsky** is the author of *The Middleman Economy: How Brokers, Agents, Dealers, and Everyday Matchmakers Create Value and Profit* (Palgrave Macmillan, 2015).

Milestones

Computer Science Awards, Appointments

NEWEST AMERICAN ACADEMY MEMBERS INCLUDE SIX COMPUTER SCIENTISTS

Among the 176 new Fellows recently elected to the American Academy of Arts and Scientists are six in the computer science arena:

▸ Jeffrey A. Dean, Google. A Google Senior Fellow, Dean is a Fellow of ACM, and shared the ACM-Infosys Foundation Award for 2012 with Sanjay Ghemawat.

▸ Sanjay Ghemawat, Google. A research scientist who works with MapReduce and other large distributed systems, Ghemawat is also the author of the popular calendar application iCal, and with Dean, wrote the 2004 paper "MapReduce: Simplified Data Processing on Large Clusters."

▸ Anna R. Karlin, University of Washington. The Microsoft Professor of Computer Science & Engineering at the University of Washington, Karlin, an ACM Fellow since 2012, writes about the use of randomized packet markings to perform IP traceback, competitive analysis of multiprocessor cache coherence algorithms, unified algorithms for simultaneously managing all levels of the memory hierarchy, Web proxy servers, and hash tables with constant worst-case lookup time.

▸ Tom M. Mitchell, Carnegie Mellon University (CMU). Chair of the Machine Learning Department at CMU and E. Fredkin University Professor, Mitchell has contributed to the advancement of machine learning, artificial intelligence, and cognitive neuroscience.

▸ Tal D. Rabin, IBM T.J. Watson Research Center. Head of the cryptography research group at theWatson Research Center, Rabin's research focuses on the design of efficient, secure encryption algorithms, as well as secure distributed algorithms, the theoretical foundations of cryptography, number theory, and the theory of algorithms and distributed systems.

▸ Scott J. Shenker, University of California, Berkeley. Leader of the Initiatives Group and Chief Scientist of the International Computer Institute, Shenker received the 2002 SIGCOMM for lifetime contribution to the field of communication networks "For contributions towards an understanding of resource sharing on the Internet." He has been an ACM Fellow since 2003.

SCHNEIDER RECEIVES CRA SERVICE AWARD

The Computing Research Association (CRA) has named Fred Schneider, Samuel B. Eckert Professor and Chair of Computer Science at Cornell University, recipient of the Service to CRA Award for his ongoing work with the organization.

A member of the CRA Board from 2007 to 2016, Schneider served as chair of the organization's Government Affairs Committee for seven years, helping to drive CRA's policy agenda, and developing the Leadership in Science Policy Institute to educate computing researchers on how science policy in the U.S. is formulated.

Schneider led the organization's Committee on Best Practices for Hiring, Promotion, and Scholarship in conducting interviews with more than 75 academic and industry computing and information unit heads to understand issues and gain insights from practice. Preliminary recommendations were vetted with department chairs and CRA Deans at the CRA Conference at Snowbird in 2014, and were published in a CRA Best Practices memo, "Incentivizing Quality and Impact: Evaluating Scholarship in Hiring, Tenure, and Promotion."