Alphago, Alphago Zero and the technology behind it

Alphago vs

Alphago beat Lee Sedol. Then Alphago zero beat Alphago. This event shows that AI is growing at an accelerating speed. Then Muzero was able to trump all of Alphago and Alphazero's skills.

The Technology and Artificial intelligence

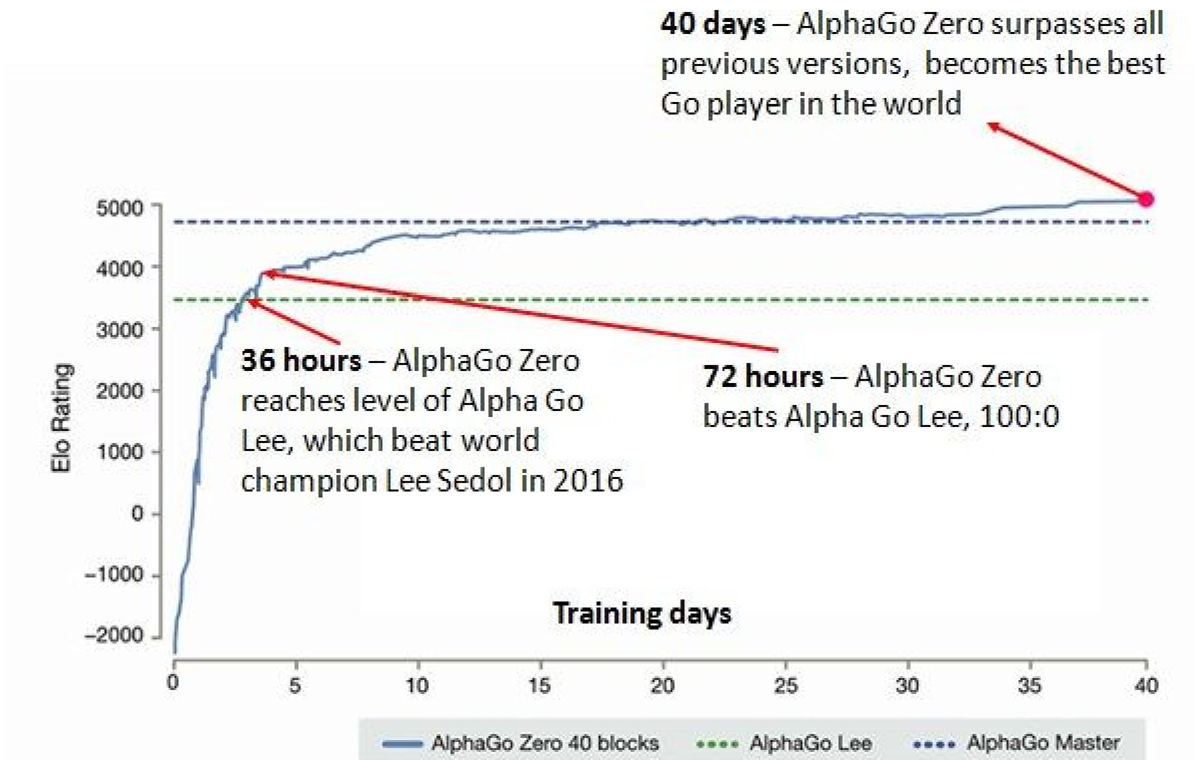
According to wiki **MuZero** is a computer program developed by artificial intelligence research company DeepMind to master games without knowing anything about their rules[1][2][3]. Its first release in 2019 included benchmarks of its performance in go, chess, shogi, and a standard suite of Atari games. The algorithm uses an approach similar to AlphaZero. It matched AlphaZero's performance in chess and shogi, improved on its performance in Go (setting a new world record), and improved on the state of the art in mastering a suite of 57 Atari games (the Arcade Learning Environment), a visually-complex domain.

MuZero was trained via self-play and play against AlphaZero, with no access to rules, opening books, or endgame tables. The trained algorithm used the same convolutional and residual algorithms as AlphaZero, but with 20% fewer computation steps per node in the search tree.[4]\

AlphaGo is a computer program that plays the board game Go.[1] It was developed by DeepMind Technologies[2] which was later acquired by Google. Subsequent versions of AlphaGo became increasingly powerful, including a version that competed under the name Master.[3]. After retiring from competitive play, AlphaGo Master was succeeded by an even more powerful version known as AlphaGo Zero, which was completely self-taught without learning from human games. AlphaGo Zero was then generalized into a program known as AlphaZero, which played additional games, including chess and shogi. AlphaZero has in turn been succeeded by a program known as MuZero which learns without being taught the rules.

AlphaGo and its successors use a Monte Carlo tree search algorithm to find its moves based on knowledge previously acquired by machine learning, specifically by an artificial neural network (a deep learning method) by extensive training, both from human and computer play.[4] A neural network is trained to identify the best moves and the winning percentages of these moves. This neural network improves the strength of the tree search, resulting in stronger move selection in the next iteration.

This statistic below shows how alphago learns



What google has to say

Games are a great testing ground for developing smarter, more flexible algorithms that have the ability to tackle problems in ways similar to humans. Creating programs that are able to play games better than the best humans has a long history - the first classic game mastered by a computer was noughts and crosses (also known as tic-tac-toe) in 1952 as a PhD candidate's project. Then fell checkers in 1994. Chess was tackled by Deep Blue in 1997. The success isn't limited to board games, either - IBM's Watson won first place on Jeopardy in 2011, and in 2014 our own algorithms learned to play dozens of Atari games just from the raw pixel inputs.

But one game has thwarted A.I. research thus far: the ancient game of Go. Invented in China over 2500 years ago, Go is played by more than 40 million people worldwide. The rules are simple: players take turns to place black or white stones on a board, trying to capture the opponent's stones or surround empty space to make points of territory. Confucius wrote about the game, and its aesthetic beauty elevated it to one of the four essential arts required of any true Chinese scholar. The game is played primarily through intuition and feel, and because of its subtlety and intellectual depth it has captured the human imagination for centuries.

But as simple as the rules are, Go is a game of profound complexity. The search space in Go is vast -- more than a googol times larger than chess (a number greater than there are atoms in the universe!). As a result, traditional "brute force" AI methods -- which construct a search tree over all possible sequences of moves -- don't have a chance in Go. To date, computers have played Go only as well as amateurs. Experts predicted it would be at least another 10 years until a computer could beat one of the world's elite group of Go professionals.

We saw this as an irresistible challenge! We started building a system, AlphaGo, described in a paper in Nature this week, that would overcome these barriers. The key to AlphaGo is reducing the enormous search space to something more manageable. To do this, it combines a state-of-the-art tree search with two deep neural networks, each of which contains many layers with millions of neuron-like connections. One neural network, the "policy network", predicts the next move, and is used to narrow the search to consider only the moves most likely to lead to a win. The other neural network, the "value network", is then used to reduce the depth of the search tree -- estimating the winner in each position in place of searching all the way to the end of the game.

AlphaGo's search algorithm is much more human-like than previous approaches. For example, when Deep Blue played chess, it searched by brute force over thousands of times more positions than AlphaGo. Instead, AlphaGo looks ahead by playing out the remainder of the game in its imagination, many times over - a technique known as Monte-Carlo tree search. But unlike previous Monte-Carlo programs, AlphaGo uses deep neural networks to guide its search. During each simulated game, the policy network suggests intelligent moves to play, while the value network astutely evaluates the position that is reached. Finally, AlphaGo chooses the move that is most successful in simulation.

We first trained the policy network on 30 million moves from games played by human experts, until it could predict the human move 57% of the time (the previous record before AlphaGo was 44%). But our goal is to beat the best human players, not just mimic them. To do this, AlphaGo learned to discover new strategies for itself, by playing thousands of games between its neural networks, and gradually improving them using a trial-and-error process known as reinforcement learning. This approach led to much better policy networks, so strong in fact that the raw neural network (immediately, without any tree search at all) can defeat state-of-the-art Go programs that build enormous search trees.

These policy networks were in turn used to train the value networks, again by reinforcement learning from games of self-play. These value networks can evaluate any Go position and estimate the eventual winner - a problem so hard it was believed to be impossible.

Below shows alphago rank development