AlphaGo

Andrei Behel AC-43И

History

- The game of Go originated in China more than 2,500 years ago. The rules of the game 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.

Introduction

- Go and chess are very popular board games, which are similar in some respects: both are played by two players taking turns, and there is no random element involved.

In 1997, Garry Kasparov was defeated by Deep Blue, a computer program written by IBM, running on a supercomputer.

In March 2016, Professional Go player Lee Sedol, one of the best players at Go, was beaten by AlphaGo computer program developed by Google DeepMind.

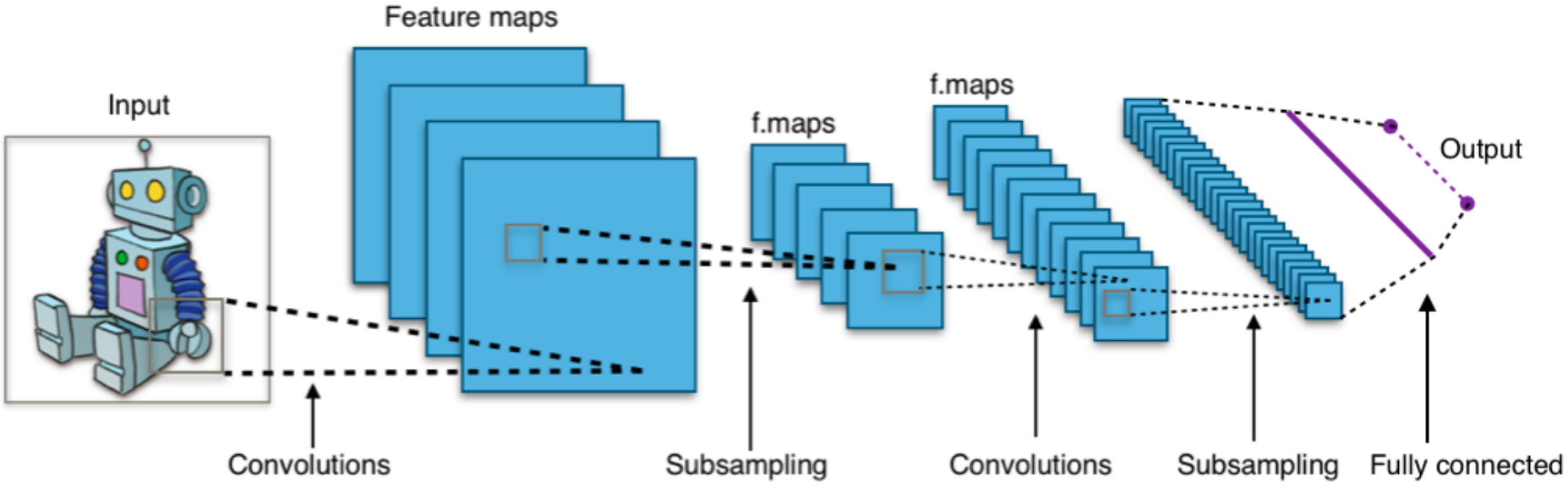
Chess vs. Go

- In chess, each player begins with 16 pieces of six different types. Each piece type moves differently. The goal of the game is to capture the opponent's king.
- Go starts with an empty board. At each turn, a player places a stone on the board. Stones all obey the same rules. The goal of the game is to capture as much territory as possible. It can therefore be argued that Go has *simpler* rules than chess.
- The *complexity* of Go is higher than chess. At each game state, a player is faced with a choice of a greater number of possible moves compared to chess (about 250 vs. 35).
- A typical game in Go might last for 150 moves vs. 80 in chess.

Structure

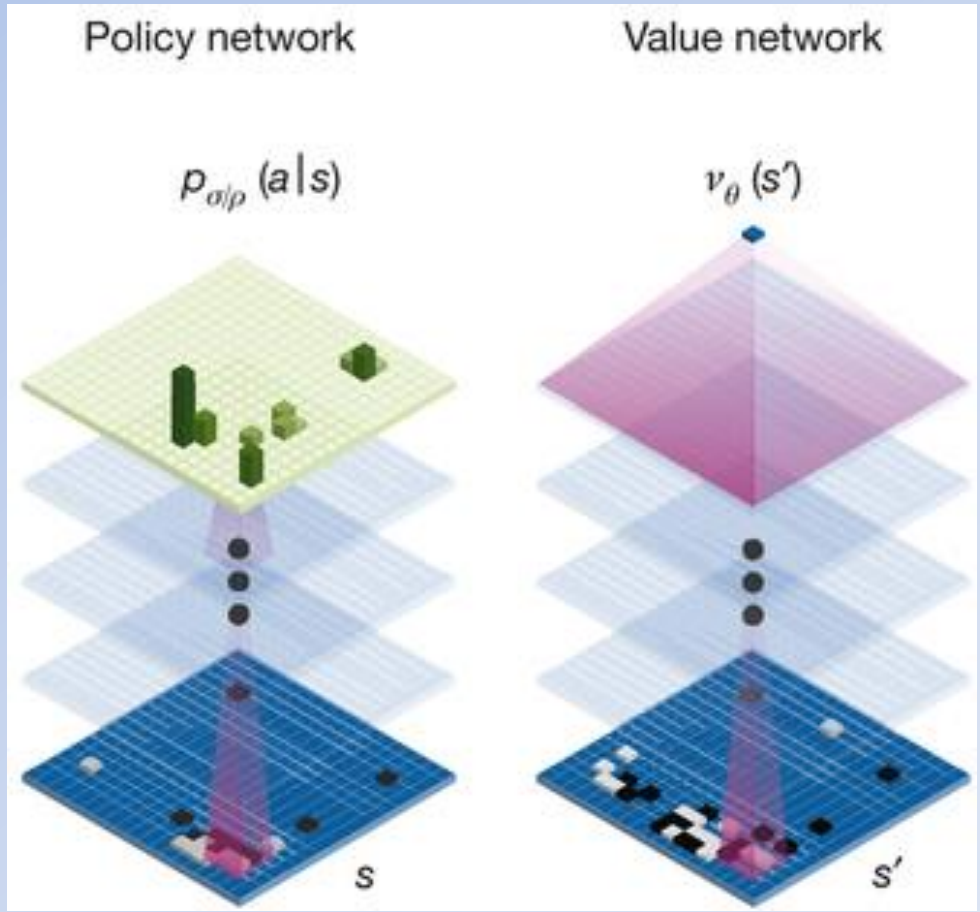
- AlphaGo relies on two different components:

1) A tree search procedure
2) Convolutional networks that *guide* the tree search procedure.



Network Structure

- In total, three convolutional networks are trained, of two different kinds: two *policy networks* and one *value network*.
- Both types of networks take as input the current game state, represented as an image.



The value network

- The value network provides an estimate of the *value* of the current state of the game: what is the probability of the black player to ultimately win the game, given the current state.
- The input to the value network is the whole game board, and the output is a single number, representing the probability of a win.

The policy network

- The policy networks provide guidance regarding which action to choose, given the current state of the game.
- The output is a probability value for each possible legal move (i.e. the output of the network is as large as the board).
- Actions (moves) with higher probability values correspond to actions that have a higher chance of leading to a win.

Results of trainings

- A policy network was trained on 30 million positions from games played by human exports, available at the KGS GO server.
- An accuracy on a withheld test-set of 57% was achieved.

- A smaller policy network is trained as well. Its accuracy is much lower (24.2%), but is much faster (2 microseconds instead of 3 milliseconds).

The policy network

- The goal should not be to be as good as possible at predicting human moves, the goal should be to have networks that are optimized to win the game.
- The policy networks were improved by letting them play against each other, using the outcome of these games as a training signal.
- This is called *reinforcement* learning, or even *deep* reinforcement learning (the networks being trained are deep).

Performance

- The AlphaGo team then tested the performance of the policy networks. They tested their best-performing policy network against Pachi, the strongest open-source Go program.
- AlphaGo's policy network won 85% of the games against Pachi.
- A convolutional network was able to outperform a system that relies extensively on search.


Monte Carlo Tree Search

MCTS

- Monte Carlo Tree Search (MCTS) is used to search the game tree.
- The idea is to run many game simulations. Each simulation starts at the current game state and stops when the game is won by one of the two players.
- At first, the simulations are completely random: actions are chosen randomly at each state, for both players. At each simulation, some values are stored, such as how often each node has been visited, and how often this has led to a win. These numbers guide the later simulations in selecting actions.

Algorithm

- AlphaGo's tree search procedure is somewhat similar to MCTS, but is guided by all three types of networks in an innovative manner.
- AlphaGo uses a mixture of the output of the value network and the result of a self-play simulation of the fast policy network:

  value of a state = value network output + simulation result.

- This method suggests a mixture of intuition and reflection.

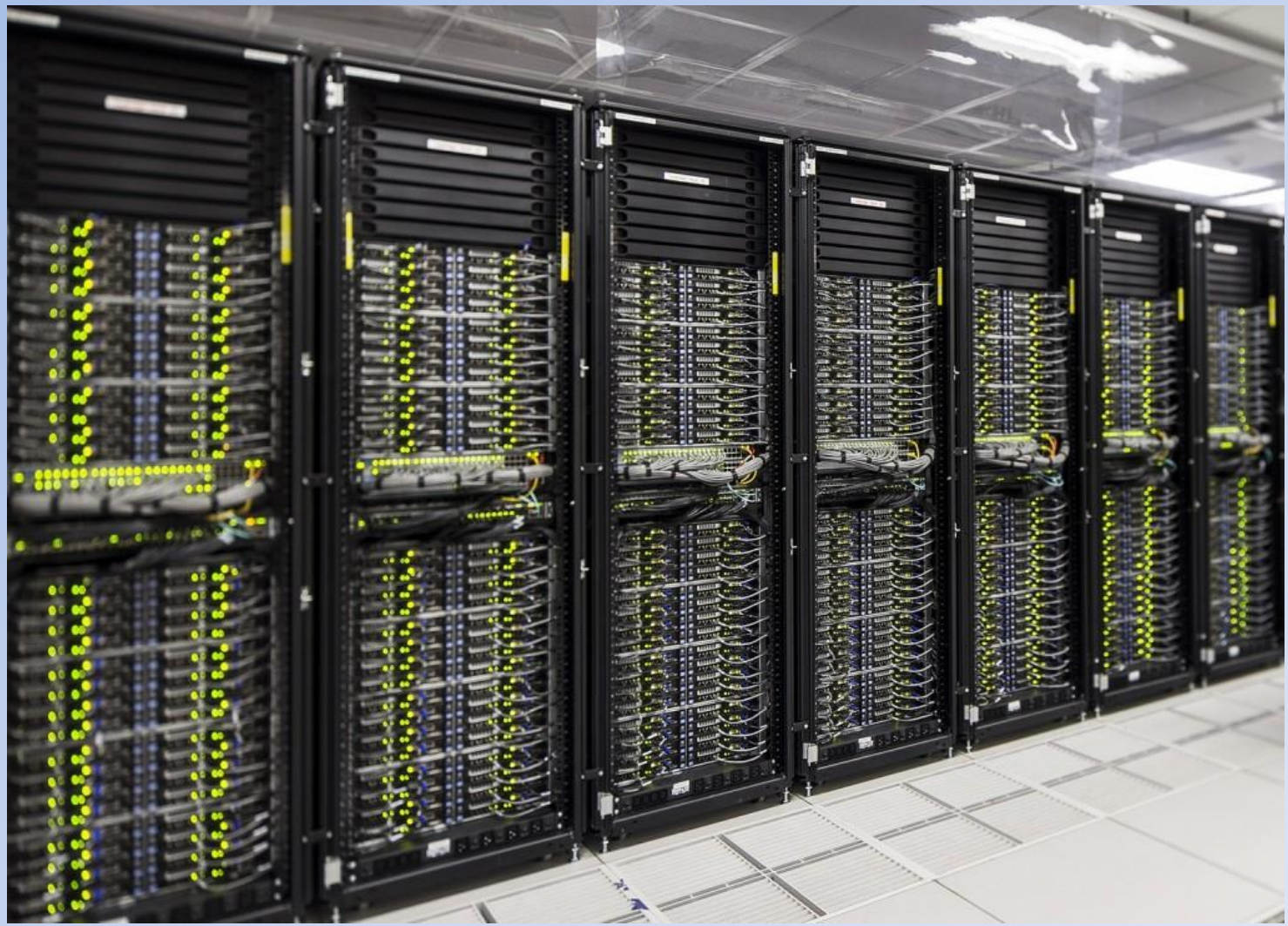
Performance Analysis

AI name	Elo rating
Distributed AlphaGo (2015)	3140
AlphaGo (2015)	2890
CrazyStone	1929
Zen	1888
Pachi	1298
Fuego	1148
GnuGo	431

Elo rating system is used for comparing the strength of players. The difference in the ratings between two players serves as a predictor of the outcome of a match, where higher ratings indicate a higher chance of winning.

Performance Analysis

- AlphaGo ran on 48 CPUs and 8 GPUs and the distributed version of AlphaGo ran on 1202 CPUs and 176 GPUs.



Performance Analysis

- On March 15, 2016, the distributed version of AlphaGo won 4-1 against Lee Sedol, whose Elo rating is now estimated at 3520.
- The distributed version of AlphaGo is now estimated at 3586.

Conclusion

- The importance of AlphaGo is enormous. The same techniques could be applied not only to robotics and scientific research, but so many other tasks, from Siri-like mobile digital assistants to financial investments.

Thank you
for your attention