CSE/ECE 848 Introduction to Evolutionary Computation

Module 1 - Lecture 3 - Part 3

Search and Machine Learning: Some Background

Wolfgang Banzhaf, CSE
John R. Koza Chair in Genetic Programming

Why are we here?

- We are here to study a particular approach to learning based on principles of evolution and/or genetics
- To understand it, we need a little context, including
  - Other natural analogies for computing
  - Other approaches to search and AI

Some Background: Computing Analogies from Nature

History is full of computing analogs!

- Boole's ideas on thought -> Boolean logic
- Babbage mechanizing calculations -> Analytical Engine
- Von Neumann, cells -> Cellular Automata
- Brain cells -> Neural Networks
- Glass/metal annealing -> Optimization
- Evolution -> Evolutionary Computation
- DNA -> DNA Computing
- Quantum Physics -> Quantum Computing
- Collective animal behaviour: Ants and flocks of birds -> Particle Swarm Optimization

Some Background: Early ideas

Many of these ideas came very early in the history of computing, really almost before true computers were even available

- Boole's and Boolean logic: 1850
- McCulloch and Pitts: Neural Networks: 1943
- Von Neumann: Cellular Automata: 1953
- Friedberg: Evolution 1958

Some Background: Attempts at using Evolution

- Turing, 1950, Computing Machinery and Intelligence

- "Intelligent behavior presumably consists of a departure from the completely disciplined behavior involved in computation."
- "Now the learning process may be regarded as a search … Since there is probably a large number of satisfactory solutions, the random method seems to be better than the systematic. It should be noticed that it is used in the analogous process of evolution."

Some Background: Attempts at using Evolution

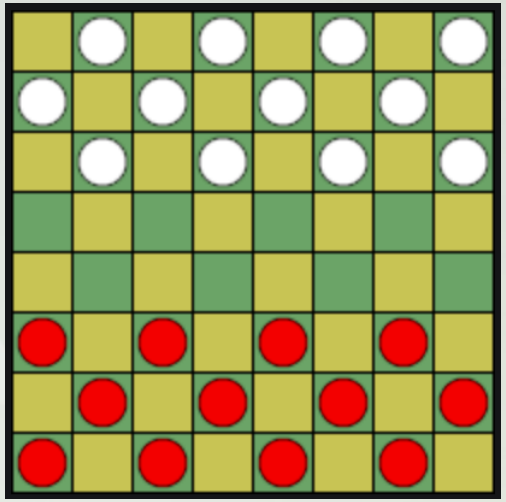
- Friedberg - attempted to solve fairly simple problems by teaching a computer how to write computer programs [Friedberg 1958/9]
- His programs consisted of instructions that would manipulate a data vector
- Friedberg's system learned by using what looks a lot like modern mutation operators - initially random individual solutions and random changes in the instructions

Some background: A brief history of Machine Learning

- A. Samuel coined the term Machine Learning in 1959
- Originally, he referred to “Machine Learning” for the attempt to allow computers to “program themselves”
- A more contemporary definition of ML (Mitchell, 1996): “ML is the study of computer algorithms that improve automatically through experience.”

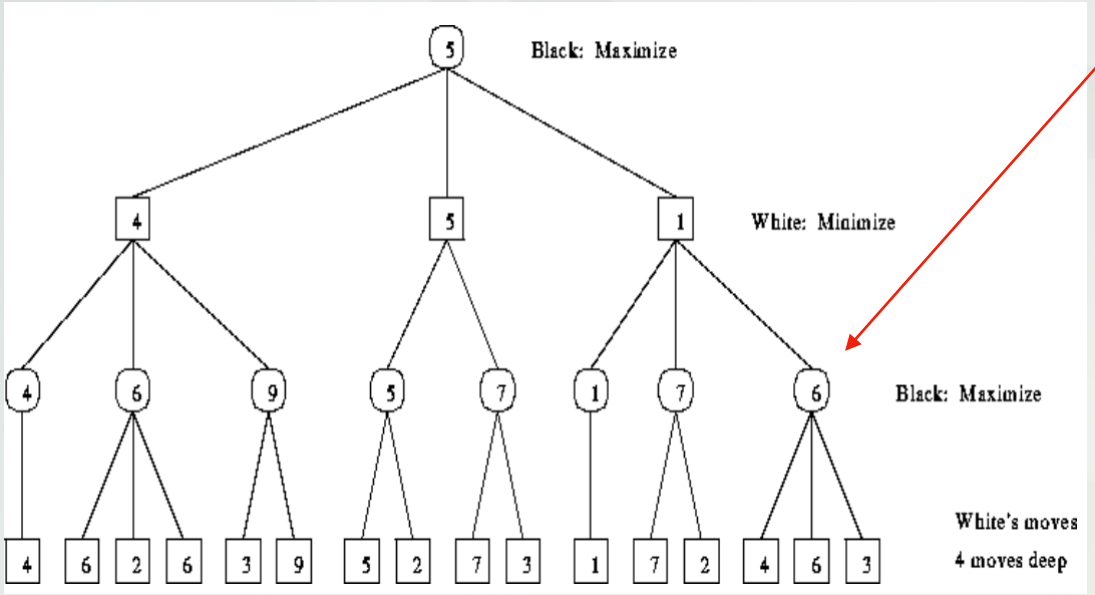
Some Background: A. Samuel and Checkers

- Samuel worked at IBM and studied checkers as a learning environment
- Because of very limited resources, he had to invent different ways to be efficient
  - alpha-beta pruning in min-max games
  - rote learning
- He also worked on how to represent human knowledge



Some Background: Minimax Strategy

- Common search process for finding the best move for a player assuming that the other player tries their best move
- Useful for "games of perfect information"



Utility value

Calculation starts with terminal nodes

Some Background: Are human players search machines?

- While something like minimax allowed a way to search a game space it was found by cognitive scientists that humans do not work like that
- Instead, humans tend to summarize combinatorial situations into a pattern
- In the case of board games, that means into a value for board positions

Some Background: An Evaluation Function

- Samuel created a linear "evaluation" function
  - Terms represent important "features" of the game (e.g. in Chess: Control of center, etc)
  - Terms had associated with them a weight for the relative importance of that term

- Need to learn those weights

Some Background: Repetitive Play

- Samuel snuck in game after game on a mainframe computer
- Examined the totality of the game using the present weights and had an algorithm tweaking the weights in a direction that would improve the results
- REPEAT - TUNE - REPEAT - TUNE
- Result: Very good checkers player

Some Background: Result

- First successful attempt at what could be done with Machine Learning
- Samuel came up with a representation and how to tune that representation based on expert knowledge and computer repetition
- Worked with limited resources, and still had great results

Some Background: Where do we stand today?

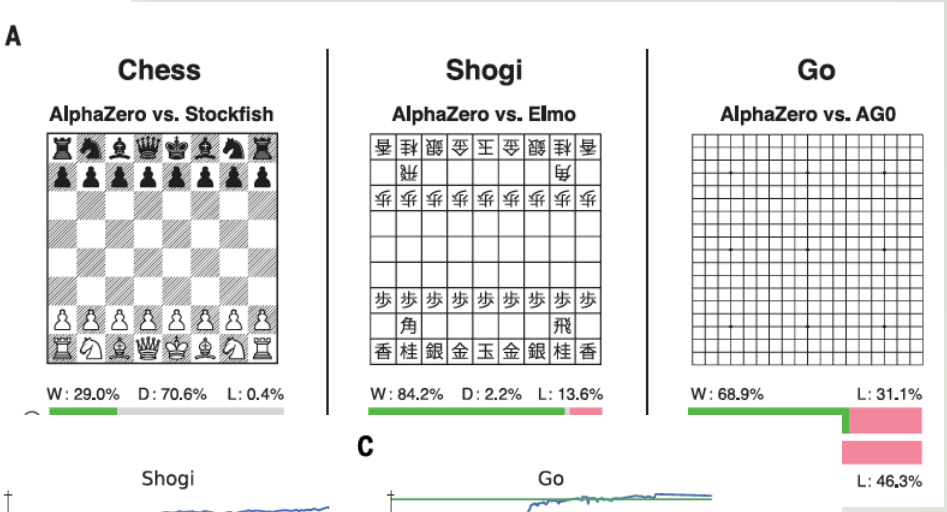
RESEARCH

COMPUTER SCIENCE

A general reinforcement learnir algorithm that masters chess, s and Go through self-play

David Silver[1,2*†], Thomas Hubert[1*], Julian Schrittwieser[1*], Ioannis Anto
Matthew Lai[1], Arthur Guez[1], Marc Lanctot[1], Laurent Sifre[1], Dharshan Ku
Thore Graepel[1], Timothy Lillicrap[1], Karen Simonyan[1], Demis Hassabis[1†]

Science **362**, 1140-1144



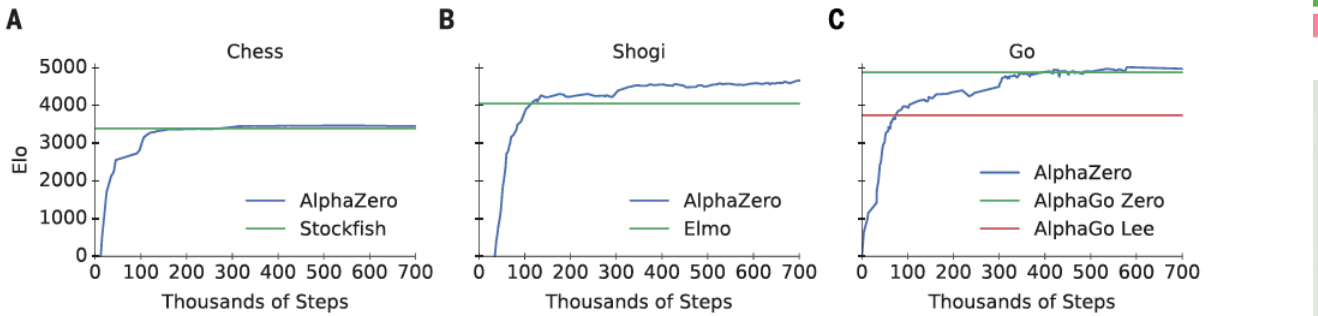


Fig. 1. Training AlphaZero for 700,000 steps. Elo ratings were computed from games between different players where each player was given 1 s per move. (**A**) Performance of AlphaZero in chess compared with the 2016 TCEC world champion program Stockfish. (**B**) Performance of AlphaZero in shogi compared with the 2017 CSA world champion program Elmo. (**C**) Performance of AlphaZero in Go compared with AlphaGo Lee and AlphaGo Zero (20 blocks over 3 days).