CASE STUDY

Chess: Deep Blue's Victory

1

AI as Sport

In 1965 the Russian mathematician Alexander Kronrod said, "Chess is the Drosophila of artificial intelligence." However, computer chess has developed as genetics might have if the geneticists had concentrated their efforts starting in 1910 on breeding racing Drosophilia. We would have some science, but mainly we would have very fast fruit flies."

- John McCarthy

2

How Intelligent is Deep Blue?

Saying Deep Blue doesn't really think about chess is like saying an airplane doesn't really fly because it doesn't flap its wings.

- Drew McDermott

3

On Game 2

(Game 2 - Deep Blue took an early lead. Kasparov resigned, but it turned out he could have forced a draw by perpetual check.)

This was real chess. This was a game any human grandmaster wouldhave been proud of.

Joel Benjamin
grandmaster, member Deep Blue team

4

Kasparov on Deep Blue

1996: Kasparov Beats Deep Blue

“I could feel --- I could smell --- a new kind of intelligence acrossthe table.”

1997: Deep Blue Beats Kasparov

“Deep Blue hasn't proven anything.”

5

Combinatorics of Chess

Opening book

Endgame

- database of all 5 piece endgames exists; database of all 6 piece games being built

Middle game

- branching factor of 30 to 40
- $1000^{(d/2)}$ positions
  - 1 move by each player = 1,000
  - 2 moves by each player = 1,000,000
  - 3 moves by each player = 1,000,000,000

6

Positions with Alpha-Beta Pruning

Search Depth		Positions
2		60
4		2,000
6		60,000
8		2,000,000
10	(<1 second DB)	60,000,000
12		2,000,000,000
14	(5 minutes DB)	60,000,000,000
16		2,000,000,000,000

7

History of Search Innovations

Shannon, Turing	Minimax search	1950
Kotok/McCarthy	Alpha-beta pruning	1966
MacHack	Transposition tables	1967
Chess 3.0+	Iterative-deepening	1975
Belle	Special hardware	1978
Cray Blitz	Parallel search	1983
Hitech	Parallel evaluation	1985
Deep Blue	ALL OF THE ABOVE	1987

8

Minimax Game Tree Search

Independently invented by Shannon (1950) and Turing (1951)

Evaluation function combines material & position

Shannon type A: all branches to same depth

Shannon type B: different branches to different depths

- Pruning "bad" nodes: doesn't work in practice
- Extend "unstable" nodes (e.g. after captures): works well in practice

9

A Note on Minimax

Minimax obviously correct -- but

- Nau (1982) discovered pathological game trees

Games where

- evaluation function grows more accurate as it nears the leaves
- but performance is worse the deeper you search!

10

Clustering

Monte Carlo simulations showed clustering is important

- if winning or loosing terminal leaves tend to be clustered, pathologies do not occur
- in chess: a position is “strong” or “weak”, rarely completely ambiguous!

But still no completely satisfactory theoretical understanding of why minimax is good!

11

Evaluation Functions

Primary way knowledge of chess is encoded

- material
- position
  - doubled pawns
  - how constrained position is

Must execute quickly - constant time

- parallel evaluation: allows more complex functions
  - tactics: patterns to recognize weak positions
  - arbitrarily complicated domain knowledge

12

Learning better evaluation functions

- Deep Blue: learn by tuning weights on subterms of evaluation function

$$f(p) = w_1f_1(p) + w_2f_2(p) + ... + w_nf_n(p)$$

- Tune weights to find best least-squares fit with respect to moves actually choosen by grandmasters in 1000+ games

13

Transposition Tables

Introduced by Greenblat's Mac Hack (1966)

Basic idea: cacheing

- once a position is evaluated, save in a hash table, avoid re-evaluating
- called “transposition” tables, because different orderings (transpositions) of the same set of moves can lead to the same position

14

Transposition Tables as Learning

Can be viewed as a simple kind of learning

- positions generalize sequences of moves
- learning on-the-fly
- don't repeat blunders: can't beat the computer twice in a row using same moves!

Deep Blue - huge transposition tables (100,000,000+), must be carefully managed

15

Time vs Space

Iterative Deepening

- a good idea in chess, as well as almost everywhere else!
- Chess 4.x, first to play at Master's level
- trades a little time for a huge reduction in space
  - lets you do breadth-first search with (more space efficient) depth-first search
- anytime: good for response-time critical applications

16

Special-Purpose and Parallel Hardware

Belle (Thompson 1978)

Cray Blitz (1993)

Hitech (1985)

Deep Blue (1987-1996)

- Parallel evaluation: allows more complicated evaluation functions
- Hardest part: coordinating parallel search

17

Deep Blue

Hardware

- 32 general processors
- 220 VSLI chess chips

Overall: 200,000,000 positions per second

- 5 minutes = depth 14

Selective extensions - search deeper at unstable positions

- down to depth 25 !

18

Evolution of Deep Blue

From 1987 to 1996

- faster chess processors
- port to IBM base machine from Sun
  - Deep Blue's non-Chess hardware is actually quite slow, in integer performance!
- bigger opening and endgame books
- 1996 differed little from 1997 - fixed bugs and tuned evaluation function!
  - After its loss in 1996, people underestimated its strength!

19

Power Comparison

[[Fig. 5.12 R&N]]

20

Tactics into Strategy

As Deep Blue goes deeper and deeper into a position, it displays elements of strategic understanding. Somewhere out there mere tactics translate into strategy. This is the closet thing I've ever seen to computer intelligence. It's a very weird form of intelligence, but you can feel it. You can smell it... It feels like thinking.

- Frederick Friedel (grandmaster), Newsday, May 9, 1997

21

Studying General AI in the Context of Games

The questions behind the compute intensive hypothesis:

- When can/must you use search in place of knowledge?
  - the compute intensive approach
- When can/must you use knowledge in place of search?
  - the knowledge intensive (expert systems) approach

22

The Role of Knowledge in Compute Intensive Programs

Deep Blue's strength lies in brute force

But - the improved evaluation function from 1996 to 1997 pushed it from impressing the world champion to beating the champion

- Traditional expert systems: a little search on top of a lot of knowledge
- Compute intensive programs: a little knowledge on top of a lot of search

23

Things that Don't Work

The applicability and limits of many AI techniques can be studied by understanding why they do NOT work for chess:

- Forward pruning
- Pattern recognition (Michalski & Negri 1977)
- Analogical reasoning (de Groot 1965, Levinson 1989)
- Partitioning

Is it time to revisit these techniques in the context of game playing?

24