CS 188: Artificial Intelligence

Advanced Topics: AI for Games

Instructors: Angela Liu and Yanlai Yang

University of California, Berkeley

(Slides adapted from Pieter Abbeel, Anca Dragan, and Stuart Russell)
Outline

- History of AI for Games
- Components of AlphaGo
  - Value Network
  - Policy Network
  - MCTS
- AlphaZero
- Games Beyond Go
Outline

- History of AI for Games
- Components of AlphaGo
  - Value Network
  - Policy Network
  - MCTS
- AlphaZero
- Games Beyond Go
Why Games?

- Clear objectives
- Can run very fast by parallelizing
- No safety or ethical concerns
1959: Arthur Samuel published checkers program that learned to play better checkers than himself!
- Disproved the belief that the capability of a computer program cannot exceed that of the programmer
- Defeated US #4 player in 1961; one draw with world champion

1992: Gerald Tesauro developed TD-Gammon, which uses a neural network to represent the value function
- Relied on very few handcrafted expert features

1997: IBM’s Deep Blue beat Garry Kasparov in chess

2014: Deepmind started their project on Go
"It may be a hundred years before a computer beats humans at Go -- maybe even longer," said Dr. Piet Hut, an astrophysicist at the Institute for Advanced Study in Princeton, N.J., and a fan of the
Go as a Target Problem

- Not much progress was made in the 2000s
- Why is hard?
  - In particular, why is it harder than chess?
- How would you start thinking about making an AI for Go?
  - What about Minimax?
Exhaustive Search is Hopeless

- Number of board configurations is greater than the number of atoms in the universe!
  - What did we learn to deal with this?
  - Evaluation functions and depth-limited search
Outline

- History of AI for Games
- **Components of AlphaGo**
  - Value Network
  - Policy Network
  - MCTS
- AlphaZero
- Games Beyond Go
Reducing depth with value network
The Value Network

Evaluation

Position

\( v(s) \)

\( \theta \)

\( s \)
Reducing breadth with policy network
The Policy Network

Move probabilities

\[ p(a|s) \]

Position
Neural Network Training Pipeline

Human expert positions → Supervised Learning policy network → Reinforcement Learning policy network → Self-play data → Value network

- Classification
- Self Play
- Regression
Supervised Learning Phase

- Supervised learning from expert databases to initialize the policy
  - 30 million boards from human experts
  - Take only one move from each board to build the dataset
  - 13-layer convolutional neural network
  - Some Go-specific human-designed input features
  - Test-set accuracy 57% (non-NN method 44%)
Reinforcement Learning Phase

- Repeat 1.28 million times
  - Play current policy with a random previous version of itself
  - Use the policy gradient method to improve the policy
Training the Value Function

- Use self-play to generate a dataset of \((s, z)\) pairs
  - \(s\) represent current board state, \(z\) is the results (how many wins and how many loses) starting from this board state
- Use supervised learning to train value function
  - And use as the evaluation function in search!
AlphaGo Results

- Expert-Level performance with only pattern matching (no rollouts)
- But best results are achieved by incorporating MCTS
- Even better results with distributed search
AlphaGo Weakness

- Can train adversarial agents to specifically find and attack AlphaGo’s weaknesses
- With an adversarial opponent, AlphaZero can completely mis-estimates the value of the positions
  - And keeps filling in its own territory with pieces!
  - No human expert player will make this kind of mistake.
Outline

- History of AI for Games
- Components of AlphaGo
  - Value Network
  - Policy Network
  - MCTS
- AlphaZero
- Games Beyond Go
AlphaZero

- **Learn from first principles**
  - gets rid of all Go-specific knowledge
  - uses neither expert databases nor any Go-specific features
  - uses only the board positions as the input

- **Can generalize to other board games**
  - Evaluated on Chess and Shogi in the paper

- **Many other improvements in the implementation**
  - combines the policy and value networks into a single network with a shared backbone and with two separate heads
  - purely uses the trained value function to evaluate positions in the tree, instead of using rollouts
  - ...
Outline

- History of AI for Games
- Components of AlphaGo
  - Value Network
  - Policy Network
  - MCTS
- AlphaZero
- Games Beyond Go
Next level games?

- Dota2 – OpenAI Five
  https://openai.com/five/

- Starcraft – Deepmind’s AlphaStar
Why is Starcraft Hard?

- The game of Starcraft is:
  - Adversarial
  - Long Horizon
  - Partially Observable
  - Realtime
  - Huge branching factor
  - Concurrent
  - Resource-rich
  - ...
AlphaStar

- Large NN trained:
  - Phase 1: supervised learning to imitate (strong) human players (why?)
  - Phase 2: reinforcement learning

- How strong is AlphaStar?
  - Won 5-0 over the world’s strongest StarCraft II players
RL agent defeats in-house OpenAI team at fairly restricted 5v5.

Mirror match of 5 fixed heroes utilizing 5 invulnerable couriers. No neutrals, runes, shrines, wards, invisibility, summons, illusions, or Scan. No Divine Rapier, Bottle, Quelling Blade, Boots of Travel, Tome of Knowledge, or Infused Raindrop.

READ “OPENAI FIVE”

WATCH VIDEO

Bill Gates
@BillGates

#AI bots just beat humans at the video game Dota 2. That’s a big deal, because their victory required teamwork and collaboration – a huge milestone in advancing artificial intelligence.

via Twitter
Summary

- The AlphaGo series demonstrate the benefits of
  - Large-scale pattern recognition
  - MCTS guided by an accurate policy
  - Lots of computation!

- More generally, recent success in AI for games show that:
  - Scaling up existing Deep RL algorithms + getting the details right got the job done!
  - This is also demonstrated in other fields, such as GPT-3 for NLP
Games that are still Unsolved

- **Contract Bridge**
  - Requires explainable policies (in the bidding phase)

- **Hanabi**
  - Purely cooperative gameplay
  - Need to reason about the beliefs and intentions of other agents