**Computer Go**

- “Task Par Excellence for AI” (Hans Berliner)
- “New Drosophila of AI” (John McCarthy)
- “Grand Challenge Task” (David Mechner)

9x9 (smallest board)  
19x19 (standard board)
A Brief History of Computer Go

- 1997: Super human Chess w/ Alpha-Beta + Fast Computer
- 2005: Computer Go is impossible!

Why?

- Branching Factor
  - Chess ≈ 35
  - Go ≈ 250

- Required search depth
  - Chess ≈ 14
  - Go ≈ much larger

- Leaf Evaluation Function
  - Chess – good hand-coded function
  - Go – no good hand-coded function
A Brief History of Computer Go

- **1997**: Super human Chess w/ Alpha-Beta + Fast Computer
- **2005**: Computer Go is impossible!
- **2006**: Monte-Carlo Tree Search applied to 9x9 Go (bit of learning)
- **2007**: Human master level achieved at 9x9 Go (bit more learning)
- **2008**: Human grandmaster level achieved at 9x9 Go (even more)

Computer GO Server rating over this period:
1800 ELO → 2600 ELO

- **2012**: Zen program beats former international champion Takemiya Masaki with only 4 stone handicap in 19x19
- **2015**: DeepMind’s AlphaGo Defeats European Champion 5-0 (lots of learning)
- **2016**: AlphaGo Defeats Go Legend Lee Sedol 4-1 (lots more learning)

AlphaGo

- Deep Learning + Monte Carlo Tree Search + HPC
- Learn from 30 million expert moves and self play
- Highly parallel search implementation
- 48 CPUs, 8 GPUs (scaling to 1,202 CPUs, 176 GPUs)

March 2016:
AlphaGo beats Lee Sedol 4-1
Mastering the game of Go with deep neural networks and tree search
Idea #1: board evaluation function via random rollouts

Evaluation Function:
- play many random games
- evaluation is fraction of games won by current player
- surprisingly effective

Even better if use rollouts that select better than random moves

Idea #2: selective tree expansion

Figure from: Chaslot (2006)
Monte Carlo Tree Search

Idea #2: selective tree expansion

Non-uniform tree growth

Arsenal of AlphaGo

Monte Carlo Tree Search
Deep Neural Networks
Supervised Learning
Huge Data Set
Distributed High-Performance Computing
Reinforcement Learning

Mastering the game of Go with deep neural networks and tree search
Deep Neural Networks

How can you write a program to distinguish cats from dogs in images?

**Machine Learning:** show computer example cats and dogs and let it decide how to distinguish them

Deep Neural Network

State-of-the-Art Performance: very fast GPU implementations allow training giant networks (millions of parameters) on massive data sets
Deep Neural Networks

**State-of-the-Art Performance:** very fast GPU implementations allow training giant networks (millions of parameters) on massive data sets.

Could a Deep NN learn to predict expert Go moves by looking at board position? Yes!

---

Arsenal of AlphaGo

- Monte Carlo Tree Search
- Deep Neural Networks
- Supervised Learning

---

**Distributed High-Performance Computing**

**Reinforcement Learning**

**Huge Data Set**

Mastering the game of Go with deep neural networks and tree search

**Supervised Learning for Go**

**Input:** Board Position

**Output:** probability of each move

Deep NN Internal Layers

Trained for 3 weeks on 30 million expert moves

- 57% prediction accuracy!

---

**Arsenal of AlphaGo**

- Monte Carlo Tree Search
- Distributed High-Performance Computing
- Deep Neural Networks
- Supervised Learning
- Reinforcement Learning
- Huge Data Set

*Mastering the game of Go with deep neural networks and tree search*

Reinforcement Learning: learn to act well in an environment via trial-and-error that results in positive and negative rewards

TD-Gammon (1992)

- Neural network with 80 hidden units (1 layer)
- Used Reinforcement Learning for 1.5 Million games of self-play
- One of the top (2 or 3) players in the world!
Reinforcement Learning for Go

**Output:** probability of each move

- Start with Deep NN from supervised learning.
- Continue to train network via self play.
- AlphaGo did this for months.
- 80% win rate against the original supervised Deep NN
- 85% win rate against best prior tree search method!
- Still not close to professional level

**Input:** Board Position

Monte Carlo Tree Search

**Idea:** use deep NN for rollout evaluation
Monte Carlo Tree Search

Idea: use deep NN for rollout evaluation

Monte Carlo Tree Search

Idea: use Deep NN for rollouts in Monte Carlo Tree Search

Problem: deep NN takes too long (msec) to evaluate
Monte Carlo Tree Search

Repeated X times

Selection → Expansion → Simulation → Backpropagation

Solution: use deep NN for selection phase
- Evaluate once per tree node
- Use probabilities to bias search toward actions that look good to deep NN

Monte Carlo Tree Search

Repeated X times

Selection → Expansion → Simulation → Backpropagation

Solution: train smaller network for rollout
- Less accurate but much faster
AlphaGo

- Deep Learning + Monte Carlo Tree Search + HPC
- Learn from 30 million expert moves and self play
- Highly parallel search implementation
- 48 CPUs, 8 GPUs (scaling to 1,202 CPUs, 176 GPUs)

2015:
AlphaGo beats European Champ (5-0)

lots of self play

March 2016:
AlphaGo beats Lee Sedol (4-1)

Computers are good at Go now – So What?

- The idea of combining search with learning is very general and widely applicable

- Deep Networks are leading to advances in many areas of AI now
  - Computer Vision
  - Speech Processing
  - Natural Language Processing
  - Bioinformatics
  - Robotics

- It is a very exciting time to be working in AI