DeepBlue, AlphaGo, and AI?
Announcements:

• Check website
• Next week: check website for updated office hours
• Final:
  – One side of handwritten notes
  – comprehensive, more emphasis on the second half
  – List of topics posted on Canvas
  – Understand the HWs

• Today:
  – DeepBlue, AlphaGo, and AI?
  – Monte Carlo Tree Search (MCTS)
Q5 histograms

Misclassification Error rate on 5.2
DeepBlue vs Kasparov:

• First match: 1996
  – Kasparov-DeepBlue: 4-2

• Second Match: 1997
  – Karasparov-DeepBlue: 2.5-3.5

• Logic-based, AI approach:
  – look ahead: alpha-beta search
  – Human board evaluations:
    • knight/bishop = 3 pawns, queen = 8 pawns, king = \( \infty \)
    • thousands of such rules
A fascination on games for “Humans vs. AI”...

- DeepBlue success didn’t amount to much for “AI”, societal impact, etc...
- Underlying techniques seem limited?
  - Hand coded rules...
  - Brute force look-ahead...
Thought to be a difficult task...

2016

AlphaGo deep RL defeats Lee Sedol (4-1)
Chess vs. Alpha Go

• Will the technical advances (underlying AlphaGo) have broader implications?


Search space: $b^d : b = 35, d = 80$

Search space: $b^d : b = 250, d = 150$
What is different today?

• Is it AI? **NO**
  – It might be better to ignore that question...
• Different from DeepBlue? **YES**
• Viewpoint: AlphaZero is at the forefront of progress in ML.
  – Pattern recognition (think of: supervised learning)
    • object recognition and ImageNet
    • the “universal” translator
  – exciting: the approach integrates “planning/search” with “pattern recognition
AlphaZero: the basic idea

• This is the “newer” method (over AlphaGo)
• A certain “lookahead” approach, using both a policy $p$ and value $v$
  – $p(a|s)$ is a distribution over move probabilities (for every state)
  – $v(s)$ is the ‘value’ of the state (e.g. the estimated probability of winning)
• Learns: $(p, v) = f(s, w)$
  – $s$ is the game state; $w$ are the model parameters (some neural net)
Look Ahead...

- The search space is too big (even for “alpha-beta pruning” to handle)
- Suppose you had a perfect estimate of the “value” of any state.
- Then one step look ahead suffices:
Suppose we have approximate values?

• We would hope that some ‘look-ahead/search’ would lessen the errors in our value estimates?
  – How do we decide which paths?

• Idea: should try to visit ’un-explored’ states
Monte Carlo Tree Search (MCTS)

- A “heuristic” which tries to balance exploration/exploitation

- AlphaZero:
  - Key idea: utilizes (+learns) a heuristic that both:
    1) estimates the values
    2) estimates a policy

\[ a_t = \operatorname{argmax}_a (Q(s_t, a) + u(s_t, a)) \]
Monte Carlo Tree Search (MCTS)

- A popular heuristic search algorithm for game play
  - By lots of simulations and select the most visited action.

Key point: how to calculate this value

Node value: Wining rate

One playout to simulate the game
Thanks!

• **Machine learning:**
  – many different methods/tools/challenges in the wild...
  – many research questions...
  – Participate in the ML community.

• *(one more week to go....)*
Have a great spring break!
AlphaZero

• AlphaGo: (the earlier system)
  – Was (sorta) specific to Go (in that it used ConvNets)
  – Use previous world championship games for SL.

• AlphaZero:
  – this is the system that “learns from scratch”...
    • At a massive computational expense...
  – works for Go and Chess (and other games)
  – gets above human level performance