When to Use RL?

台灣清華大學
孫民教授

VSLab

RL itself is limited

- “Pure” Reinforcement Learning (cherry)
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- Supervised Learning (icing)
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- Unsupervised/Predictive Learning (cake)
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

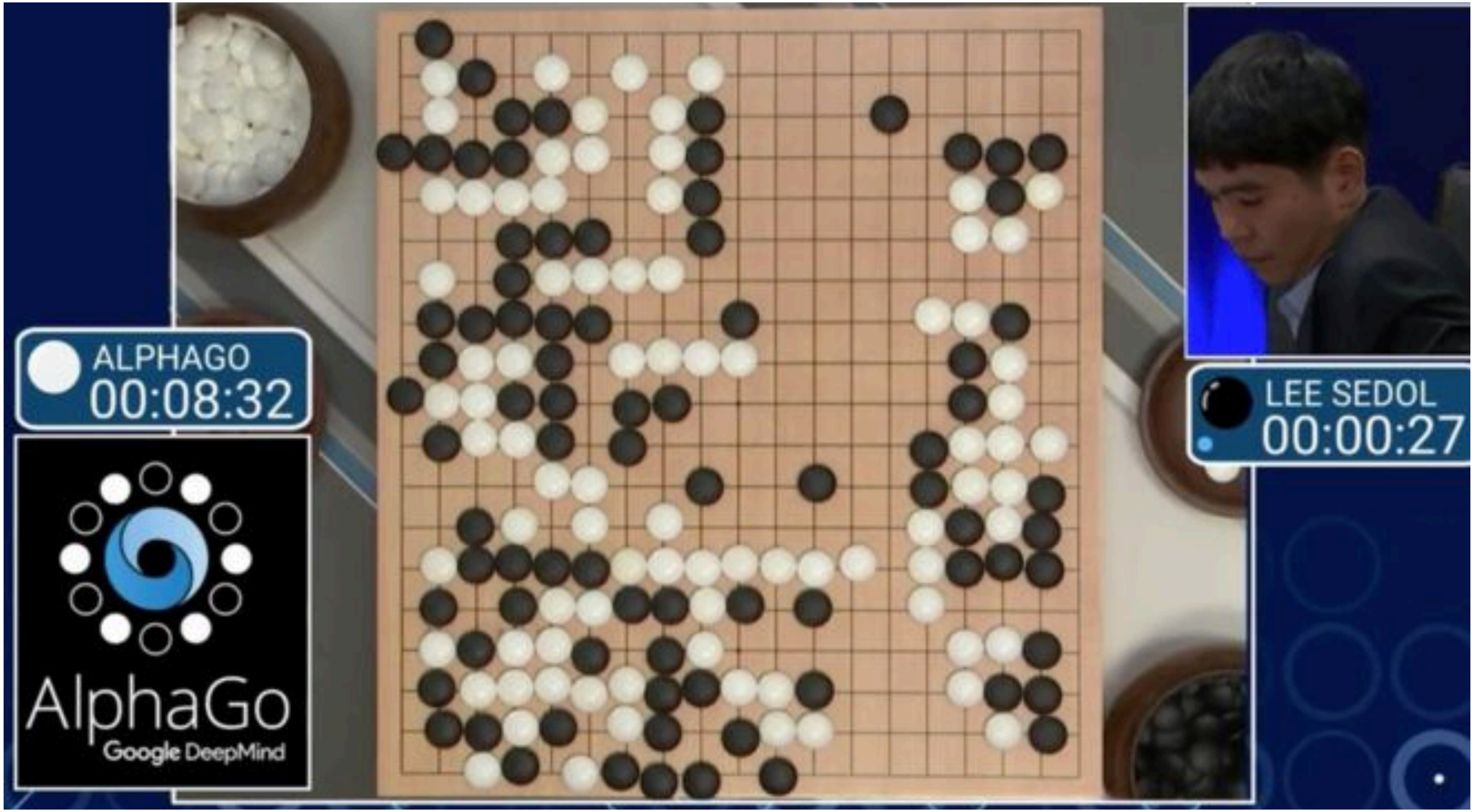
- (Yes, I know, this picture is slightly offensive to RL folks. But I’ll make it up)

https://www.youtube.com/watch?v=Ount2Y4qxQo

Motivation

- "The brain has about 10^14 synapses and we only live for about 10^9 seconds. So we have a lot more parameters than data. This motivates the idea that we must do a lot of **unsupervised (predictive) learning** since the perceptual input (including proprioception) is the only place we can get 10^5 dimensions of constraint per second."
  - Geoffrey Hinton (in his 2014 AMA on Reddit)
    (but he has been saying that since the late 1970s)

Can We Apply AlphaGo Everywhere?



2016 by Google DeepMind

AlphaGo, in context

- **Fully deterministic**. There is no noise in the rules of the game; if the two players take the same sequence of actions, the states along the way will always be the same.
- **Fully observed**. Each player has complete information and there are no hidden variables. Texas hold'em does not satisfy this property because you cannot see the cards of the other player.
- **The action space is discrete**. A number of unique moves are available. In contrast, in robotics you might want to instead emit continuous-valued torques at each joint.
- **We have access to a perfect simulator.** The effects of any action are known exactly. This is a strong assumption that AlphaGo relies on quite strongly, but is also quite rare in other real-world problems.
- **Each episode/game is relatively short**, of approximately 200 actions. This is a relatively short time horizon compared to other RL settings which may involve thousands (or more) of actions per episode.
- **The evaluation is clear,** fast and allows a lot of trial-and-error experience. In other words, the agent can experience winning/losing millions of times, which allows is to learn, slowly but surely, as is common with deep neural network optimization.

https://medium.com/@karpathy/alphago-in-context-c47718cb95a5

Suitable Problems for RL

- Can train in super real-time (e.g., typical in simulation)
  - Games
  - Data Center
  - Autonomous Vehicle
- Simulation is close to real-world cases
  - Games
  - Data Center*

RL is Powerful when Suitable

- Alpha Zero
  - Go
  - Chess
  - Japanese game Shoju

  without human knowledge

https://www.technologyreview.com/s/609736/alpha-zeros-alien-chess-shows-the-power-and-the-peculiarity-of-ai/

Examples

- Neuron Architectural Search
- Intention Anticipation through Trigger-based Sensing
- Language Style Adaptation