

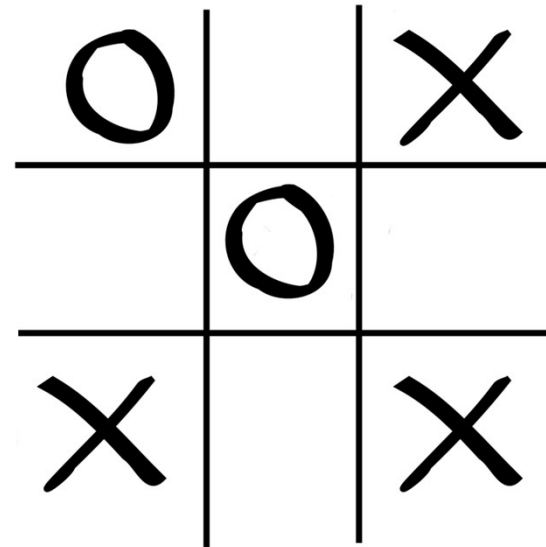
Reinforcement Learning: The Future

CS786

9th September 2024

The state space problem in model-free RL

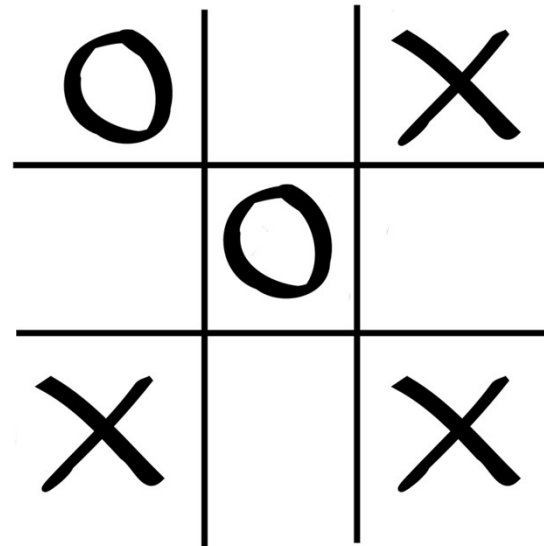
- Number of states quickly becomes too large
 - Even for trivial applications
 - Learning becomes too dependent on right choice of exploration parameters
 - Explore-exploit tradeoffs become harder to solve



State space = 765 unique states

Solution approach

- Cluster states
- Design features to stand in for important situation elements
 - Close to win
 - Close to loss
 - Fork opp
 - Block fork
 - Center
 - Corner
 - Empty side



What's the **basis** for your evaluation?

- Use domain knowledge to spell out what is better
- $\phi_1(s) \rightarrow$ self center, opponent corner
- $\phi_2(s) \rightarrow$ opponent corner, self center
- $\phi_3(s) \rightarrow$ self fork, opponent center
- $\phi_4(s) \rightarrow$ opponent fork, self center
- ... as many as you can think of
- These are *basis* functions

Value function approximation

- RL methods have traditionally approximated the state value function using linear basis functions

$$V(s) \approx V_{\mathbf{w}}(s) = \mathbf{w}^T \phi(s)$$

- \mathbf{w} is a k valued parameter vector, where k is the number of features that are part of the function ϕ
- Implicit assumption: all features contribute independently to evaluation

Function approximation in Q-learning

- Approximate the Q table with linear basis functions

$$Q(s, a) = \sum_i^k \phi_i(s, a) w_i$$

- Update the weights

$$w_i \leftarrow w_i + \alpha \delta \phi_i(s, a)$$

– Where δ is the TD term

Non-linear approximations

- Universal approximation theorem – a neural network with even one hidden layer can approximately represent any continuous-valued function
- Neural nets were always attractive for their representation generality
 - But were hard to train
 - That changed with the GPU revolution ten years ago

The big idea

- Approximate Q values using non-linear function approximation

$$Q(s, a) \approx Q_{\theta}(s, a) = f(s, a, \theta)$$

- Where θ are the parameters of the neural network and $f(x)$ is the output of the network for input x
- Combines both association and reinforcement principles
 - Association buys us state inference
 - Reinforcement buys us action policy learning

Conv nets basics

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |

Image patch

| | | |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |

Filter

| | | | | |
|-----------------|-----------------|-----------------|---|---|
| 1 _{x1} | 1 _{x0} | 1 _{x1} | 0 | 0 |
| 0 _{x0} | 1 _{x1} | 1 _{x0} | 1 | 0 |
| 0 _{x1} | 0 _{x0} | 1 _{x1} | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |

Image








Convolution

| | | |
|---|--|--|
| 4 | | |
| | | |
| | | |

Convolved
Feature

<https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>

Discriminability from diverse filtering

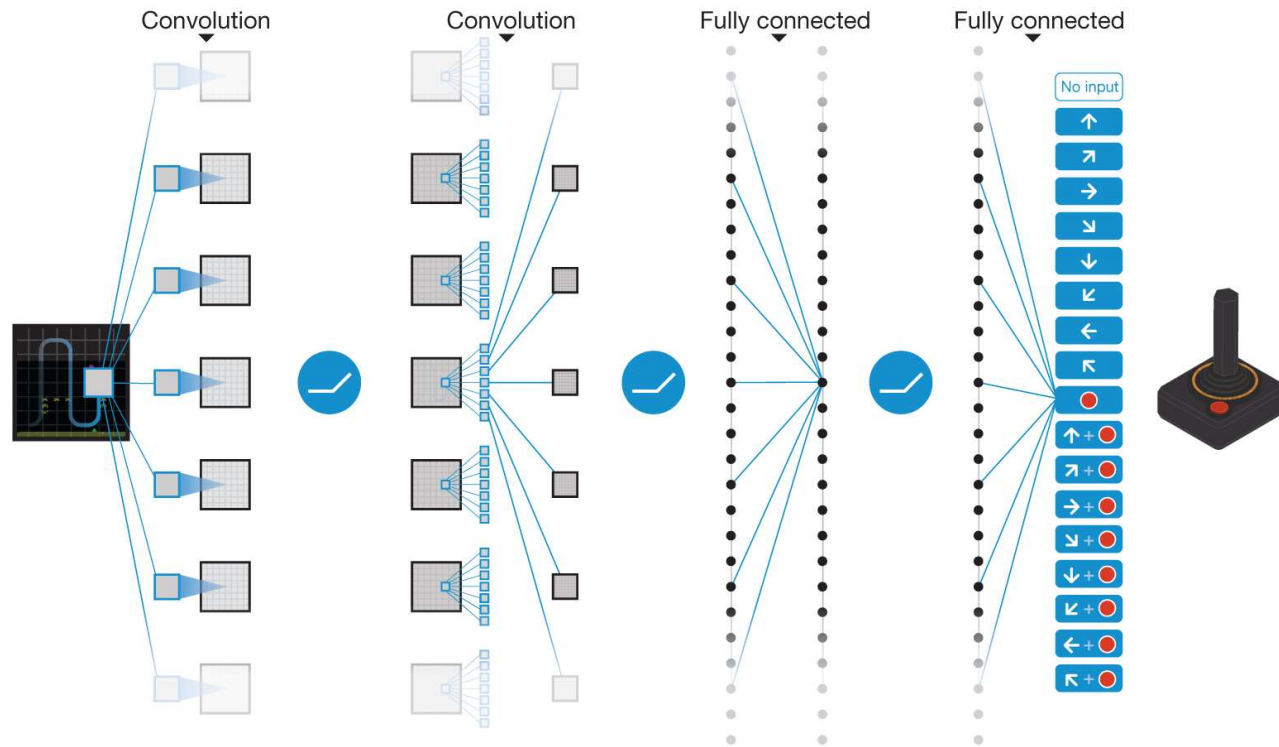
| Operation | Filter | Convolved Image |
|----------------------------------|--|---|
| Identity | $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ |  |
| Edge detection | $\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$ |  |
| | $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ |  |
| | $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$ |  |
| Sharpen | $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ |  |
| Box blur (normalized) | $\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ |  |
| Gaussian blur (approximation) | $\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ |  |

The Atari test bench

- A very popular RL test bench
- Limited space of actions
- Non-stop reward feedback
- Free to use
- Earlier methods used features handcrafted for each game



Schematic illustration of the convolutional neural network.



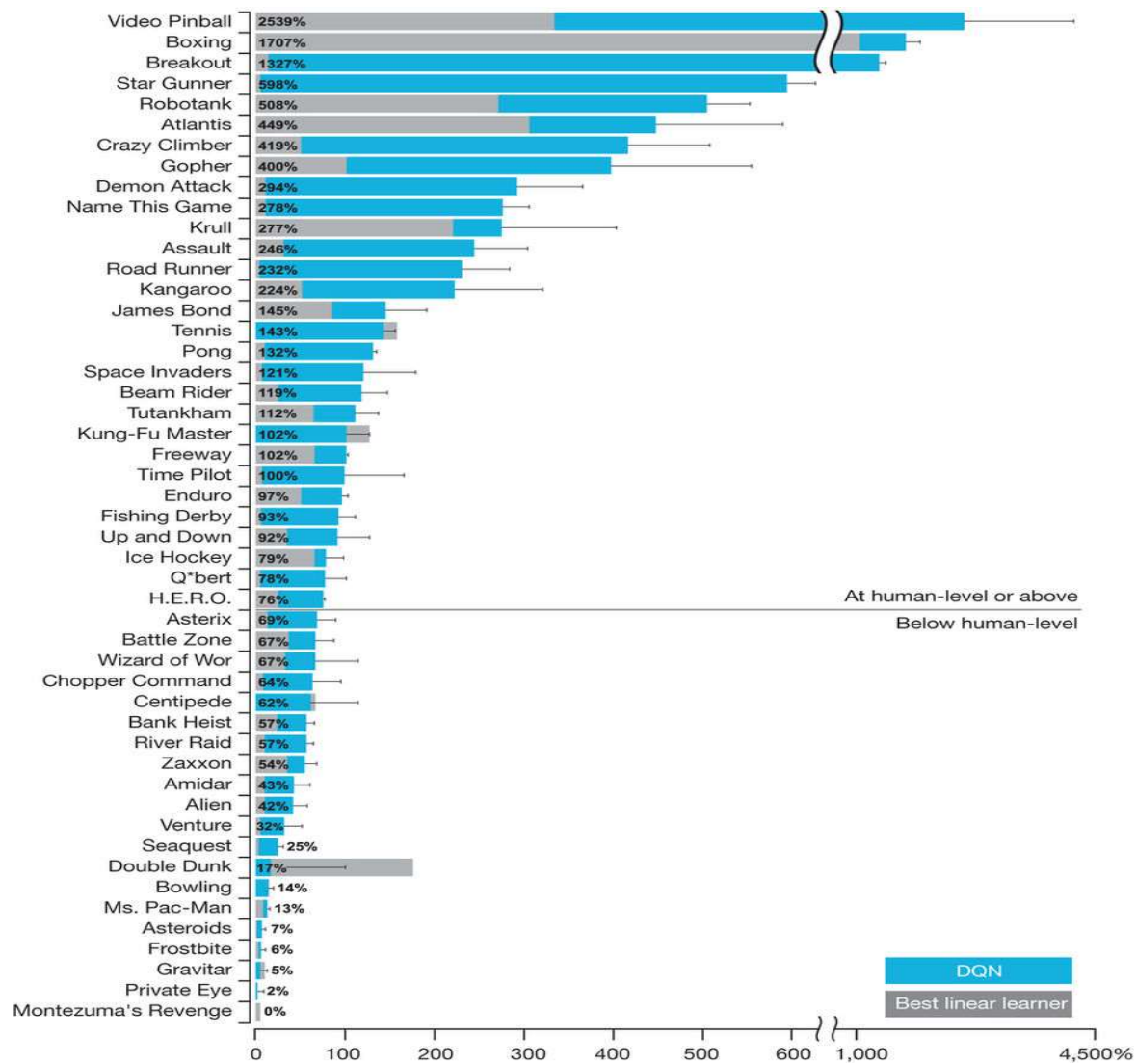
V Mnih *et al.* *Nature* **518**, 529-533 (2015) doi:10.1038/nature14236

nature

Deep Q network

- Basic Q learning algorithm augmented a bunch of different ways
 - Use of experience replay
 - Use of batch learning
 - Use of non-linear function approximation

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s') \sim U(D)} \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i) \right)^2 \right]$$



AlphaZero

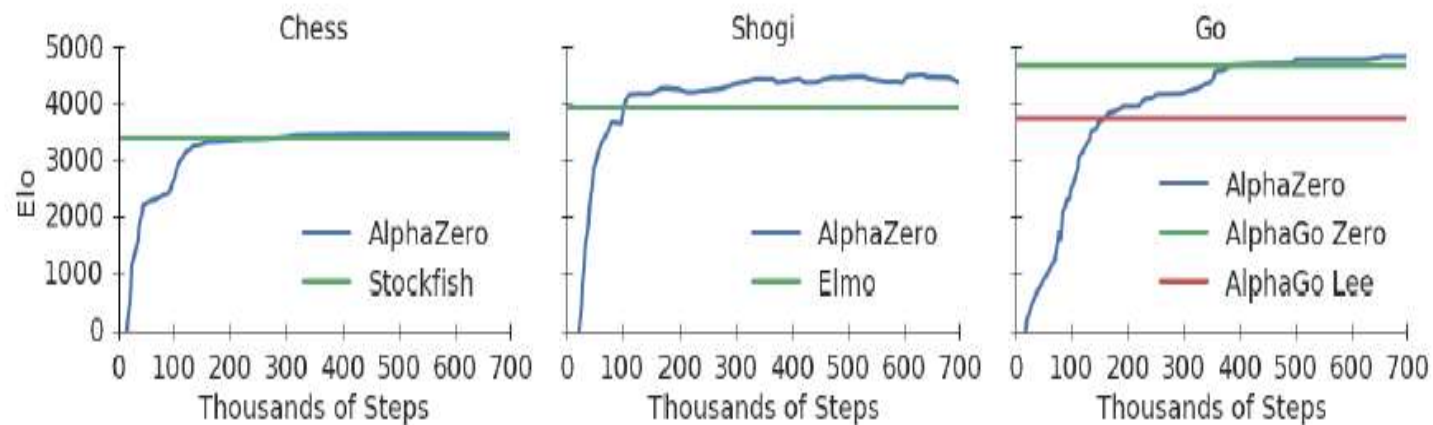


Figure 1: Training *AlphaZero* for 700,000 steps. Elo ratings were computed from evaluation games between different players when given one second per move. *a* Performance of *AlphaZero* in chess, compared to 2016 TCEC world-champion program *Stockfish*. *b* Performance of *AlphaZero* in shogi, compared to 2017 CSA world-champion program *Elmo*. *c* Performance of *AlphaZero* in Go, compared to *AlphaGo Lee* and *AlphaGo Zero* (20 block / 3 day) (29).

Secret ingredient

- Some algorithmic innovations
 - MCTS
- Mostly, just lots and lots of computation
- 5000 TPUs to generate game-play
- 64 TPUs to train the DQN
- This work closes a long chapter in game-based AI research
 - And brings research in RL to a dead end!

<https://www.quora.com/Is-reinforcement-learning-a-dead-end>

Summary

- Deep reinforcement learning is the cognitive architecture of the moment
 - Perhaps of the future also
 - Beautifully combines the cognitive concepts of association and reinforcement
 - Excellent generalizability across toy domains
 - Limitations exist: timing, higher-order structure, computational complexity etc.

RL is as intelligent as a railway engine

- You tell it what to do
 - Shape behavior using reward signals
- It does what you tell it to do
 - After tons of cost-free simulations
- Can work in specific toy domains
- Does not work as a model of real real-time learning



<https://www.sciencedirect.com/science/article/pii/S0921889005800259>

Elephants don't play chess

- The world as its own model
 - Subsumption architecture
 - Don't try to model the world with states and rewards
 - Give individual robot components their own (simple, maybe hardwired) goals
 - Tweak components until you get behavior that looks reasonable
 - Big success
 - Roomba!

https://en.wikipedia.org/wiki/BEAM_robotics

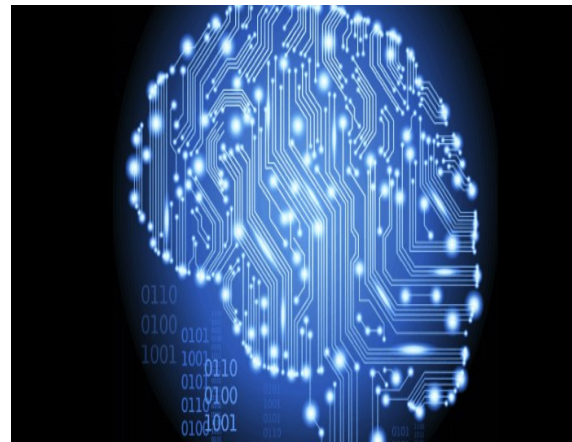
<http://cid.nada.kth.se/en/HeideggerianAI.pdf>

What is the mind?

- Classical Antiquity
 - Vata, pitta and kapha
 - Explain dispositions and traits
 - Greeks had the same concept
 - Five humors for the five elements
- Middle Ages
 - Humans are machines (Descartes)
 - The brain is a telegraph (von Helmholtz)
- Modern Age
 - The human nervous system is basically digital (von Neumann)
 - The brain is a complex computer (a 21st century truism)

The map-territory illusion

- Metaphors are useful to guide thinking about natural phenomena
- Have to be philosophically careful not to mistake metaphors for reality



The mind constructs information

- Current neuroscience and AI fashions argue for the brain as a passive store of information
- The mind is not a passive receptacle
- It exists in a body with a long history and a complex present
- It constructs information based on being in the world (Heidegger)

Neural RL

Positives

- The Schultz, Montague & Dayan result shows a clear role for RL in human learning systems
- Supported by evidence from neurophysiology ([link](#), [link](#))
- We are beginning to understand how model-free learning could support model-based behavior ([link](#))

Negatives

- Representation of time and timing remains a gaping hole in the RL paradigm ([link](#))
- Assumption of reward availability remains problematic
- Simulations are too sample inefficient to match human behavior ([link](#))

Coda: the successor representation

- Remember the value iteration equation?

$$V^\pi(s) = \sum_a \pi(a|s) [R(s, a) + \sum_{s'} P(s'|s, a) \gamma V^\pi(s')]$$

- Peter Dayan showed long ago that it could be rewritten as

$$V^\pi(s) = \sum_{s'} M^\pi(s, s') \sum_a \pi(a|s') R(s', a)$$

- Where

$$M^\pi(s, s') = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \mathbb{I}(s_t = s') \mid S_0 = s],$$

- The successor representation offers one [explanation](#) for how the TD signal could yield model-based policies

Summary

- Principles of association and reinforcement are being used prominently in both ML and cognitive science
- They work well for specific applications and as partial explanations of human learning
- But not as general models of learning to be in the world
- Much remains to be learned about
 - Internal representations
 - Processes controlling internal representations
 - Embodied priors
 - How embodied priors interact with processes controlling internal representations
- We will start talking about how this is currently being done computationally beginning after mid sem week