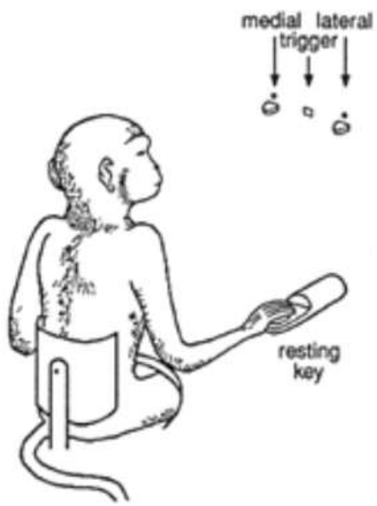
RL and the brain

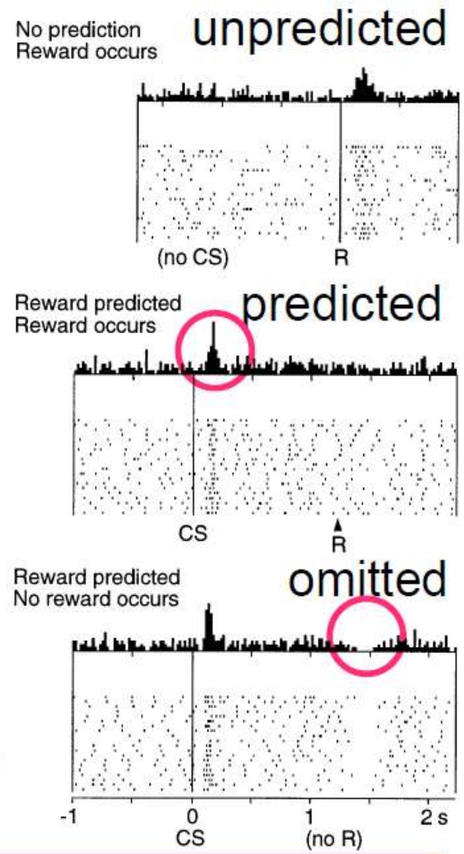
CS 786

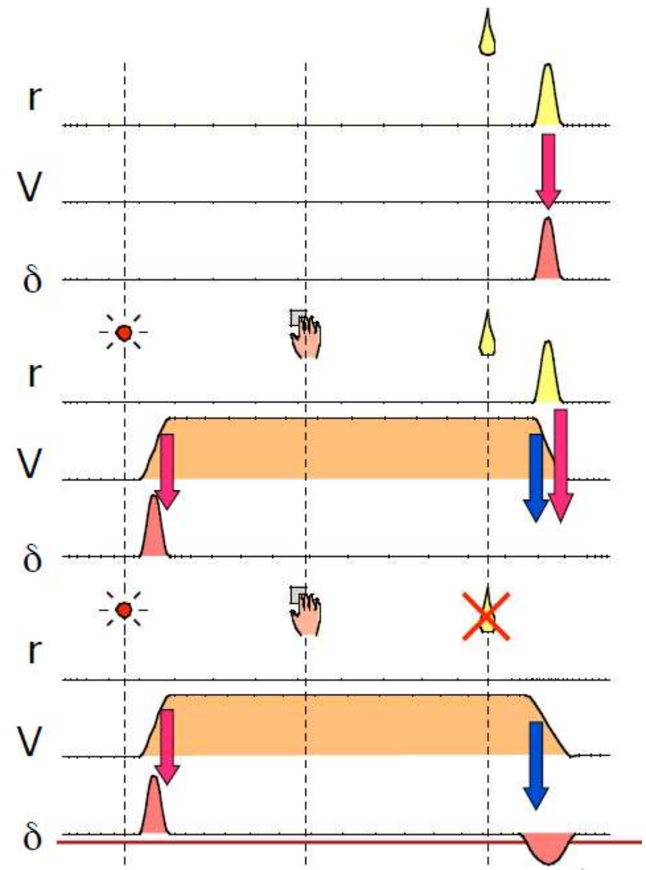
September 2nd 2024

Dopamine Neurons Code TD Error

$$\delta(t) = r(t) + \gamma V(s(t+1)) - V(s(t))$$



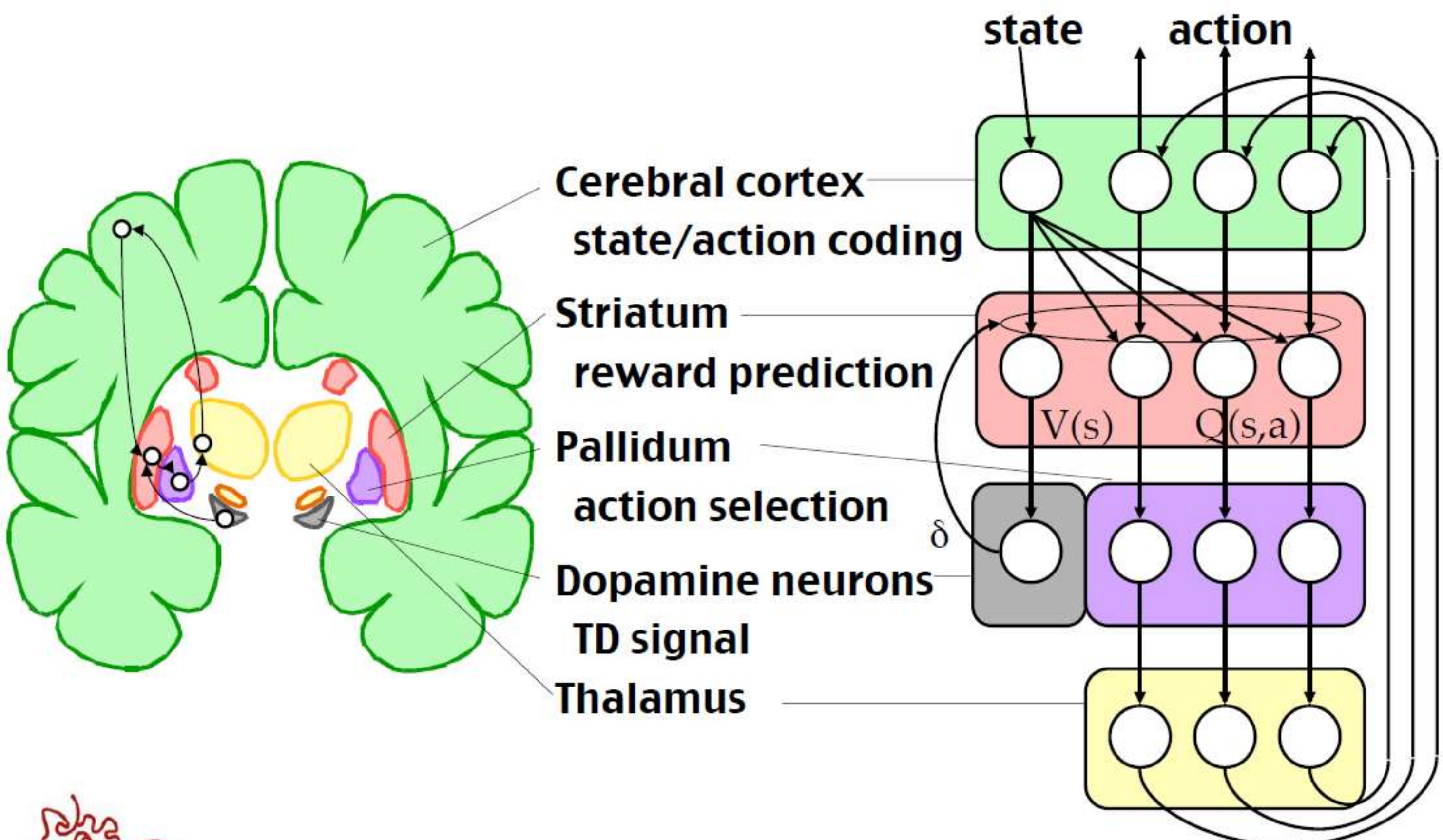




(Schultz et al. 1997)

Basal Ganglia for Reinforcement Learning?

(Doya 2000, 2007)



Cocaine addiction (a success story)

- Cocaine pharmacodynamics
  - Is a dopamine reuptake inhibitor
- Under normal circumstances the TD signal is

$$\delta_t = r_{t+1} + \gamma V\left(s_{t+1}\right) - V\left(s_t\right)$$

- When you take cocaine

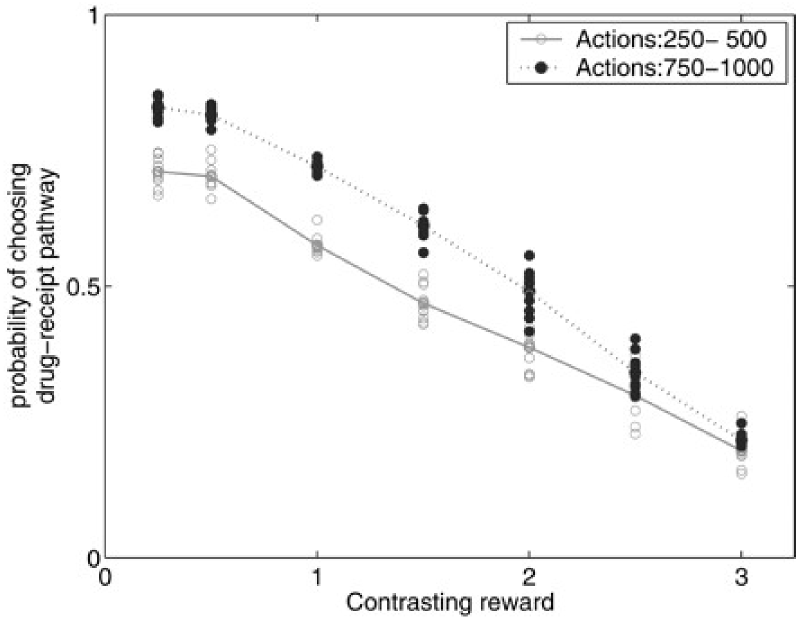
$$\delta_t = \max\left\{r_{t+1} + \gamma V\left(s_{t+1}\right) - V\left(s_t\right) + D_t, D_t\right\}$$

The mechanics of physical addiction

- In the beginning, taking cocaine is associated with positive TD signal
  - So taking cocaine is learned
- But presence of cocaine in the system prevents the TD signal from becoming negative
  - No matter what you do
  - Behavior cannot be unlearned!

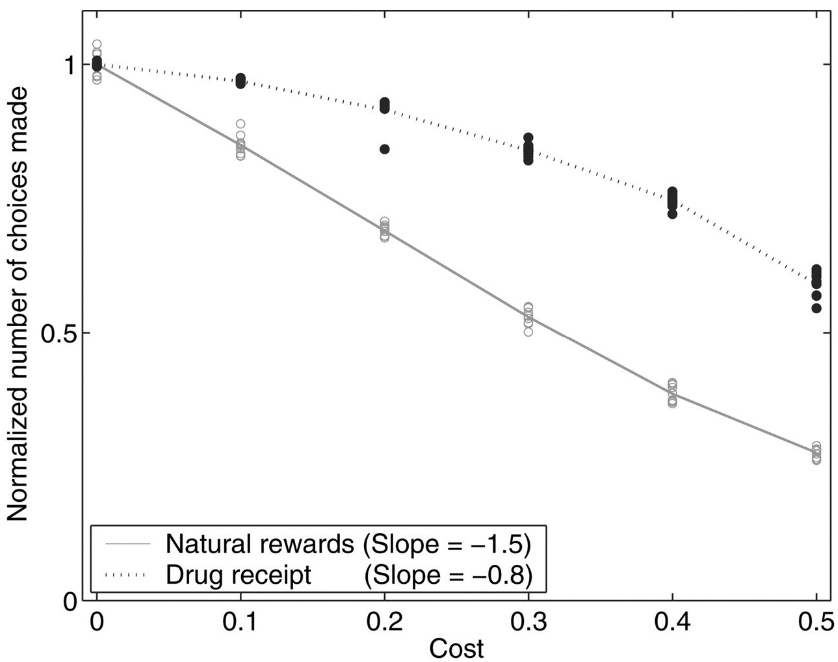
Reward insensitivity

- Observer will become unable to tradeoff drug consumption with other rewards



Cost insensitivity

- Observe is unable to reduce preference with increasing cost



Cocaine addiction (a success story)

- Cocaine pharmacodynamics
  - Is a dopamine reuptake inhibitor
- Under normal circumstances the TD signal is

$$\delta_t = r_{t+1} + \gamma V\left(s_{t+1}\right) - V\left(s_t\right)$$

- When you take cocaine

$$\delta_t = \max\left\{r_{t+1} + \gamma V\left(s_{t+1}\right) - V\left(s_t\right) + D_t, D_t\right\}$$

Addiction: a computational process gone awry *(Redish, 2004)*

The model free vs model-based debate

- Model free learning → actions that lead to rewards become more preferable
- What about goal-based decision-making?
  - Do animals not learn the physics of the world in making decisions?
- Model-based learning
- People have argued for two systems
  - Thinking fast and slow (Balleine & O'Doherty, 2010)

Levels of analysis

Level	Description
Computational	What is the problem?
Algorithmic	How is the problem solved?
Implementation	How this is done by networks of neurons?

RL in the brain

- What is the problem?
  - Reinforcement → learning preferences for actions that lead to desirable outcomes
- How is it solved?
  - MDPs provide a general mathematical structure for solving decision problems under uncertainty
  - RL was developed as a set of online learning algorithms to solve MDPs
  - A critical component of model-free RL algorithms is the temporal difference signal
  - Hypothesis: brain is implementing model-free RL?
- Implementation
  - Spiking rates of dopaminergic neurons in the basal ganglia and ventral striatum behave as if they are encoding this TD signal

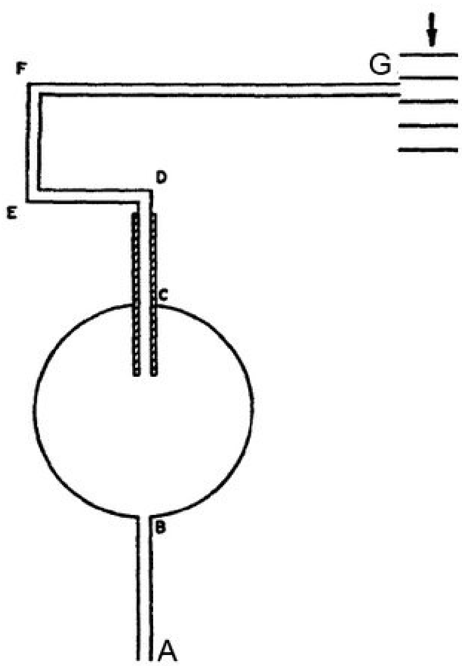
Implication?

- Model-free learning
  - Learn the mapping from action sequences to rewarding outcomes
  - Don’t care about the physics of the world that lead to different outcomes
  - Is this a realistic model of how human and non-human animals learn?

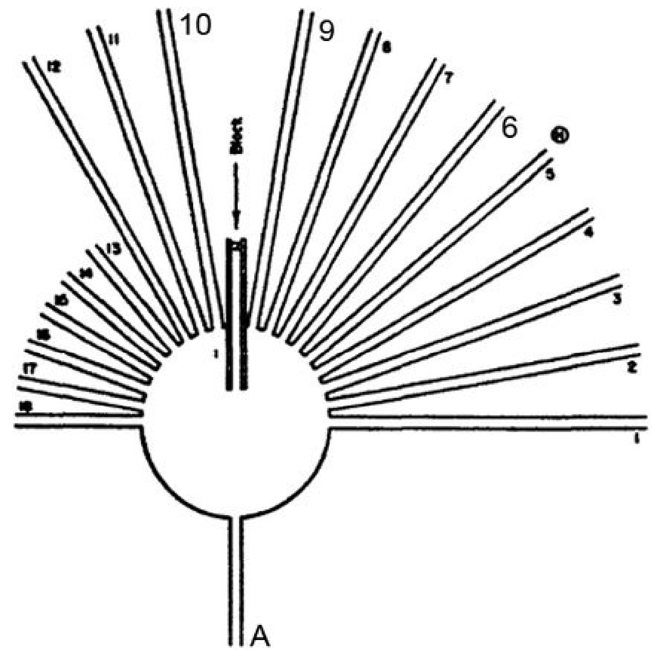
Learning maps of the world

Of mice and men

Cognitive maps in rats and men



Apparatus used in preliminary training

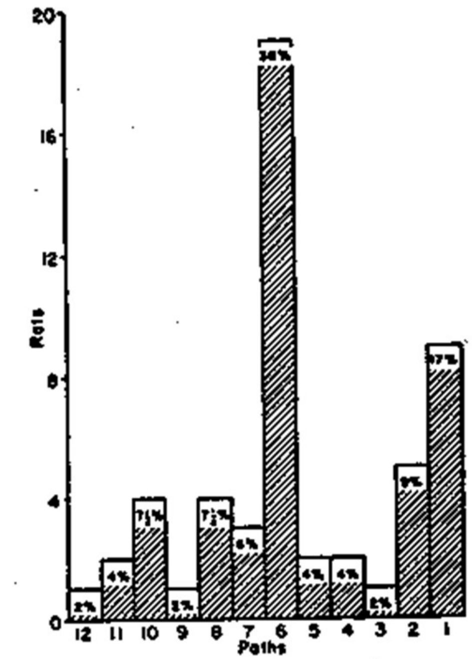


Apparatus used in the test trial

(From E. C. Tolman, B. F. Ritchie and D. Kalish, Studies in spatial learning. I. Orientation and short-cut. *J. exp. Psychol.*, 1946, **36**, p. 17.)

Rats learned a spatial model

- Rats behave as if they had some sense of p(s'|s,a)
- This was not explicitly trained
- Generalized from previous experience
- Corresponding paper is recommended reading
- So is Tolman's biography



Numbers of rats which chose each of the paths

FIG. 17

(From E. C. Tolman, B. F. Ritchie and D. Kalish, Studies in spatial learning. I. Orientation and the short-cut. *J. exp. Psychol.*, 1946, 36, p. 19.)

http://psychclassics.yorku.ca/Tolman/Maps/maps.htm

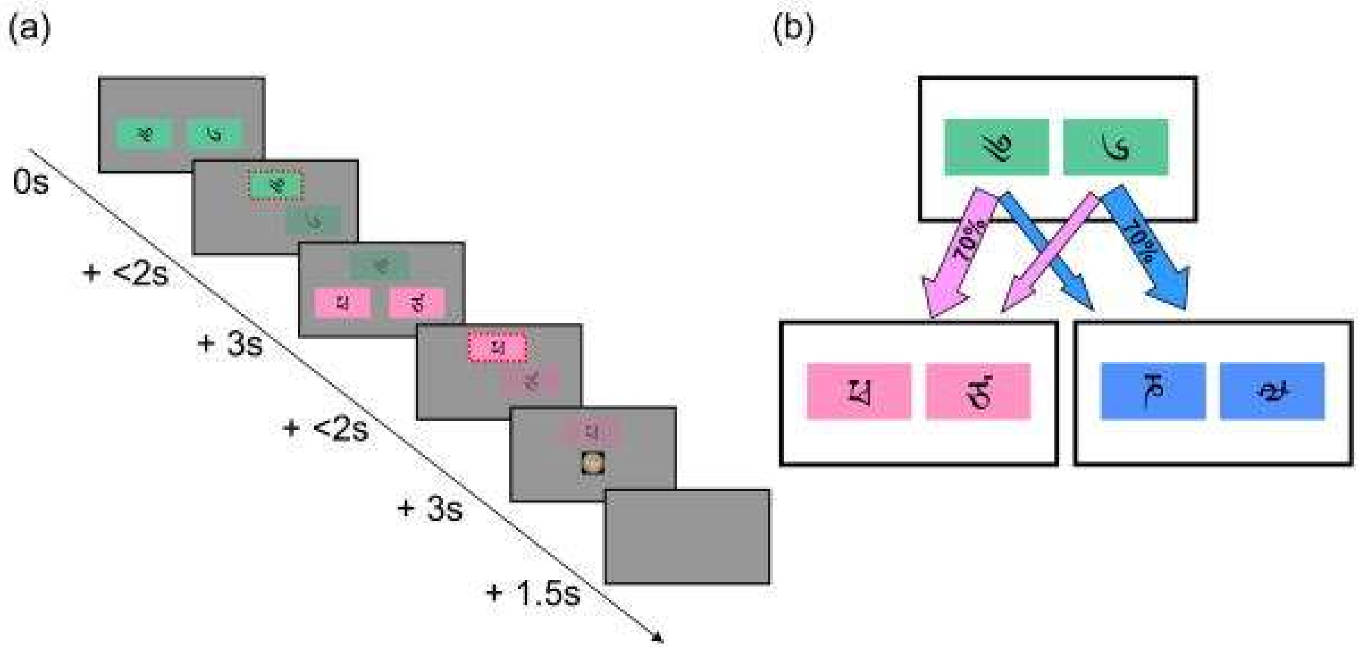
Multiple modes of learning

Computational Approaches to Reward Learning

	Model-Based	Model-Free
Instrumental	**Goal-Directed Plans** *Computation*: Tree Searches & Act-Outcome Cognition *Example*: Act chosen based on declarative memory of previous hedonic values embedded in modeled world relationships *Feature*: Adjusting action after outcome devaluation or contingency degradation needs retasting to update goal value or reduce uncertainty [1,2]	**Habits** *Computation*: Temporal Difference Prediction Error Mechanism *Example*: Incremental trial-by-trial learning of a cached habit strength *Feature*: Habitual responding persists unchanged after outcome revaluation as an automatic movement procedure [2]
Pavlovian	**UCS Identity Representations** *Computation*: Mesolimbic UCS identity transform into CS incentive salience *Examples*: Novel body-brain state makes Dead Sea salt CS suddenly attractive; Dopamine drug stimulations enhance 'wanting' for CS before new CS UCS learning. *Feature*: Immediate CS transform without need of learning about UCS new value [3]	**Cached UCS Value Predictions** *Computation*: Incremental teaching signals form cached value prediction *Example*: Temporal difference hypotheses of phasic dopamine signals as prediction error learning mechanisms. *Feature*: Requires incremental retraining of CS-UCS pair after UCS revaluation to alter predicted CS future value [4]

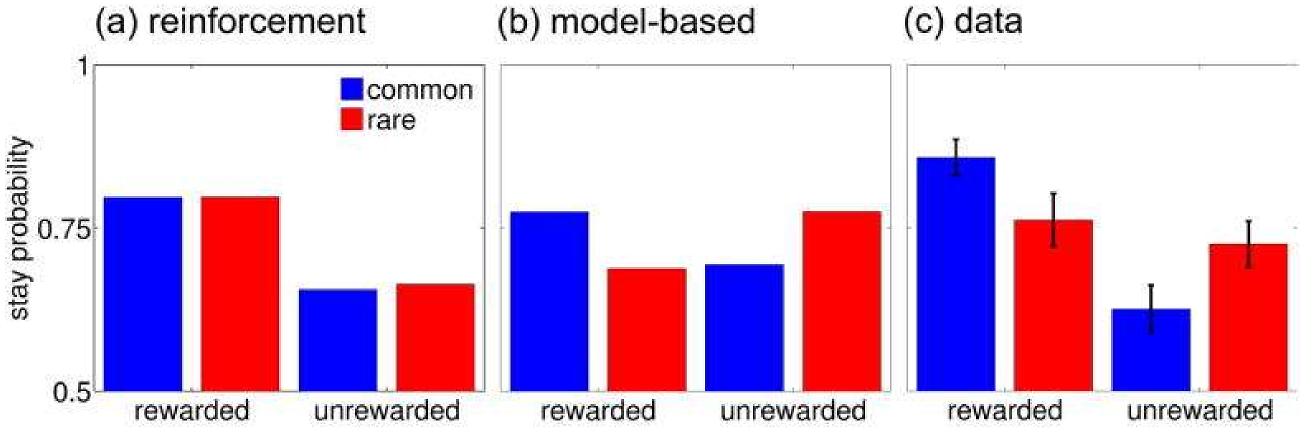
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4074442/

A contemporary experiment



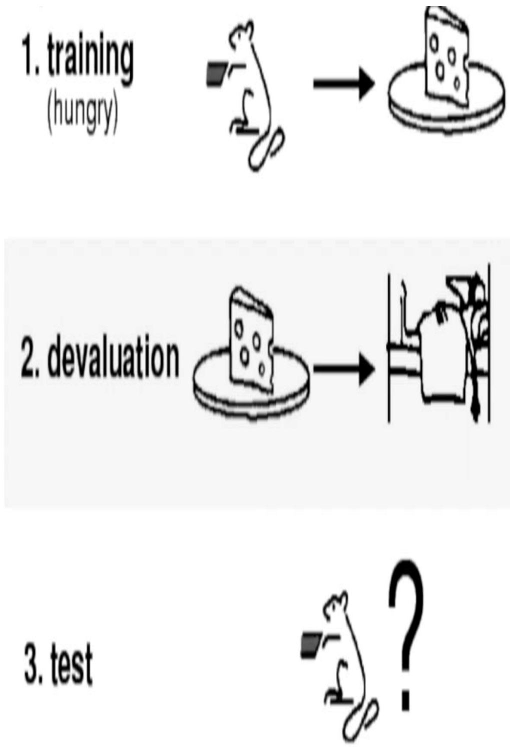
- The Daw task (Daw et al, 2011) is a two-stage Markov decision task
- Differentiates model-based and model-free RL accounts empirically

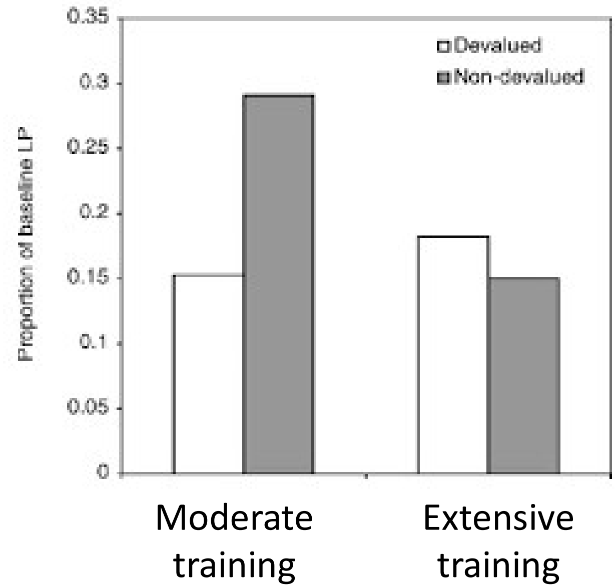
Predictions meet data



- Behavior appears to be a mix of both strategies
- What does this mean?
- Active area of research

Some hunches





(Holland, 2004; Kilcross & Coutureau, 2003)

Current consensus

- In moderately trained tasks, people behave as if they are using model-based RL
- In highly trained tasks, people behave as if they are using model-free RL
- Nuance:
  - Repetitive training on a small set of examples favors model-free strategies
  - Limited training on a larger set of examples favors model-based strategies

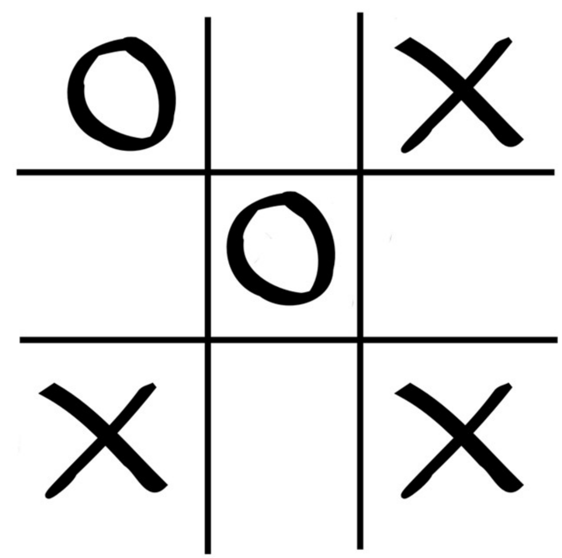
(Fulvio, Green & Schrater, 2014)

Open RL problems

DESIGNING BETTER STATE SPACES

The state space problem in model-free RL

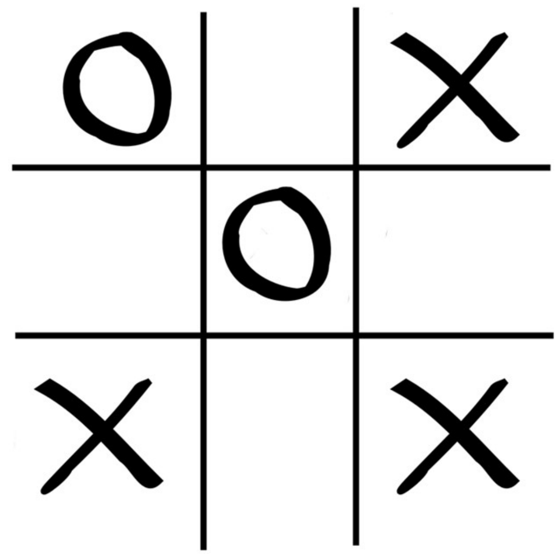
- Number of states quickly becomes too large
  - Even for trivial applications
  - Learning becomes too dependent on right choice of exploration parameters
  - Explore-exploit tradeoffs become harder to solve



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
- $\phi_1(s)$ → self center, opponent corner
- $\phi_2(s)$ → opponent corner, self center
- $\phi_3(s)$ → self fork, opponent center
- $\phi_4(s)$ → 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

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

  - **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 as action policy learning

https://www.nature.com/articles/nature14236

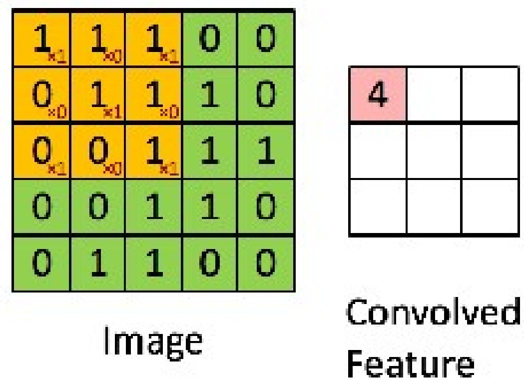
Conv nets basics

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

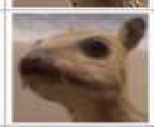
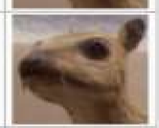
Filter



Convolution

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

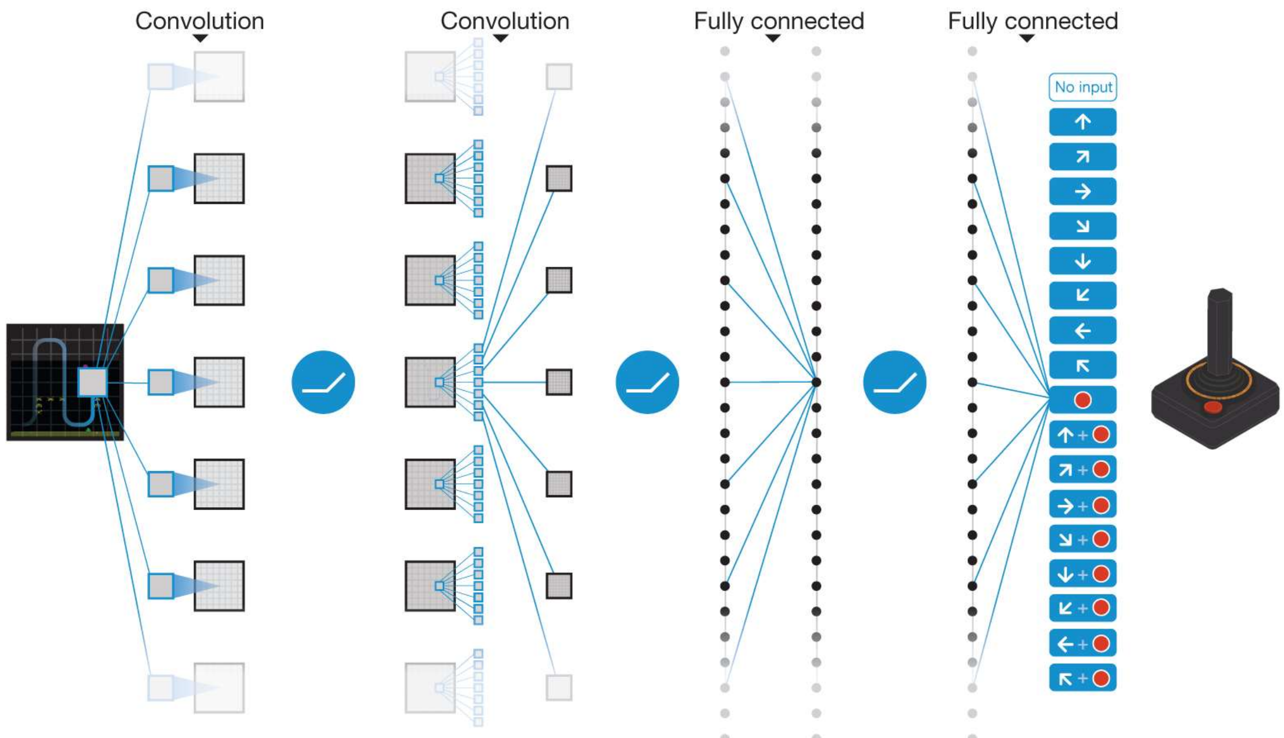
Discriminability from diverse filtering

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.



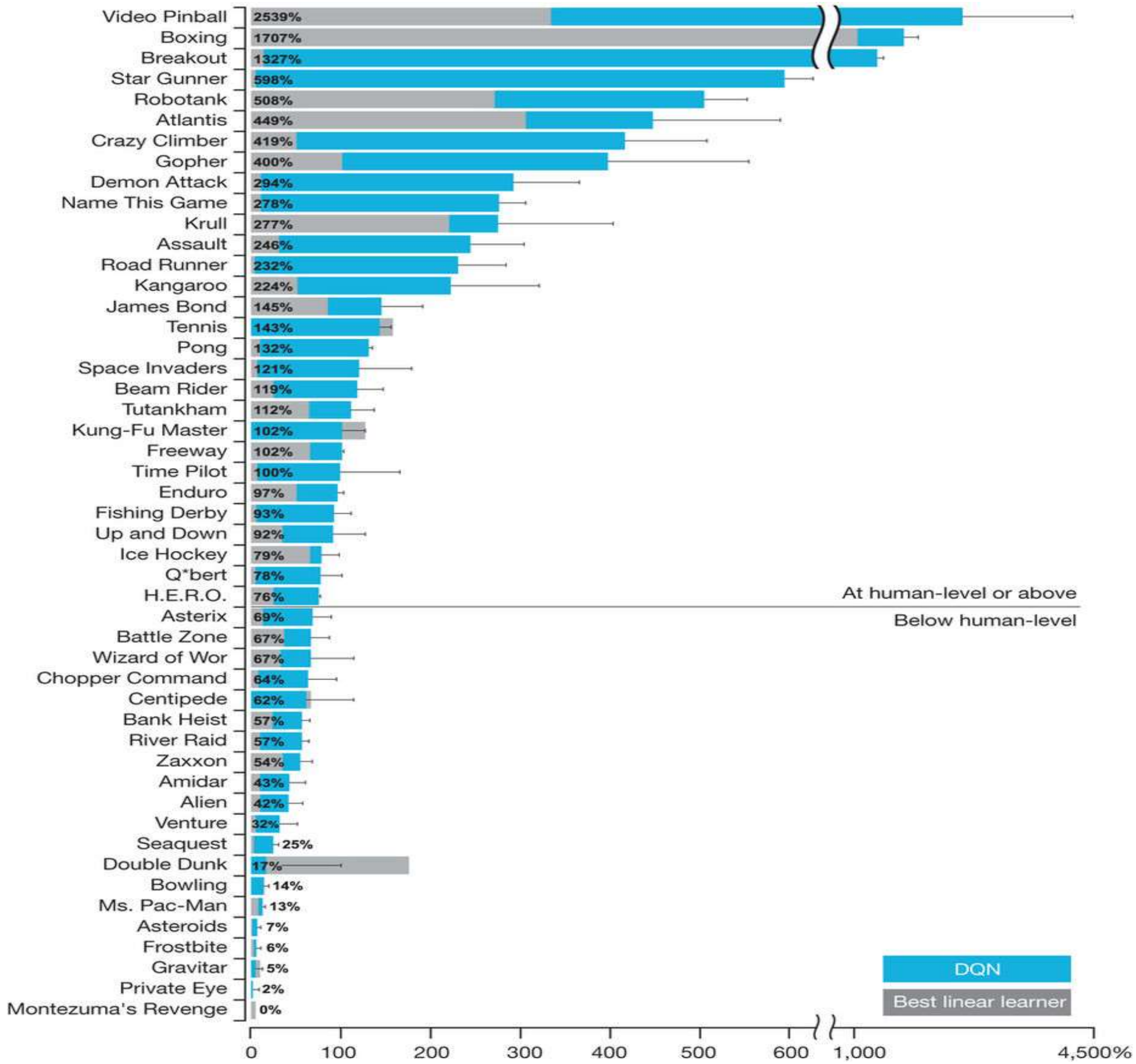
V Mnih *et al. Nature* **518**, 529-533 (2015) doi:10.1038/nature14236

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

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



Video Pinball
Boxing
Breakout
Star Gunner
Robotank
Atlantis
Crazy Climber
Gopher
Demon Attack
Name This Game
Krull
Assault
Road Runner
Kangaroo
James Bond
Tennis
Pong
Space Invaders
Beam Rider
Tutankham
Kung-Fu Master
Freeway
Time Pilot
Enduro
Fishing Derby
Up and Down
Ice Hockey
Q*bert
H.E.R.O.
Asterix
Battle Zone
Wizard of Wor
Chopper Command
Centipede
Bank Heist
River Raid
Zaxxon
Amidar
Alien
Venture
Seaquest
Double Dunk
Bowling
Ms. Pac-Man
Asteroids
Frostbite
Gravitar
Private Eye
Montezuma's Revenge
2539%
1707%
1327%
598%
508%
449%
419%
400%
294%
278%
277%
246%
232%
224%
145%
143%
132%
121%
119%
112%
102%
102%
100%
97%
93%
92%
79%
78%
76%
69%
67%
67%
64%
62%
57%
57%
54%
43%
42%
32%
25%
17%
14%
13%
7%
6%
5%
2%
0%
At human-level or above
Below human-level
DQN
Best linear learner
0
100
200
300
400
500
600
1,000
4,500%

AlphaZero

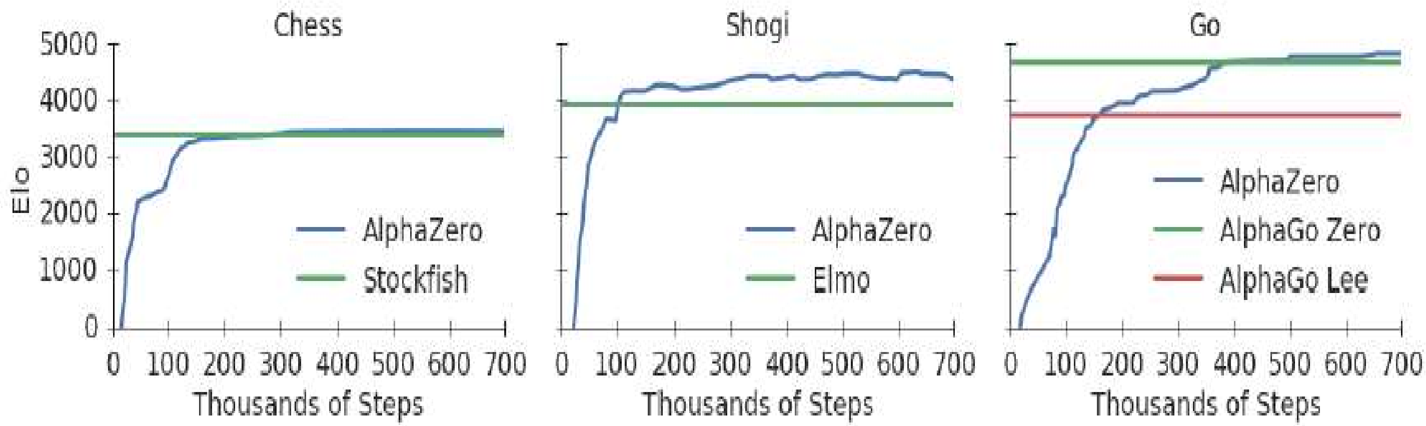


Figure 1: Training *AlphaZero for 700,000 steps. Elo ratings were computed from evaluation games between different players when given one second per move. a Performance of AlphaZero in chess, compared to 2016 TCEC world-champion program Stockfish. b Performance of AlphaZero in shogi, compared to 2017 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.