Introduction to reinforcement learning in the brain

Jesse Geerts
Sainsbury Wellcome Centre neuroinformatics course 2024

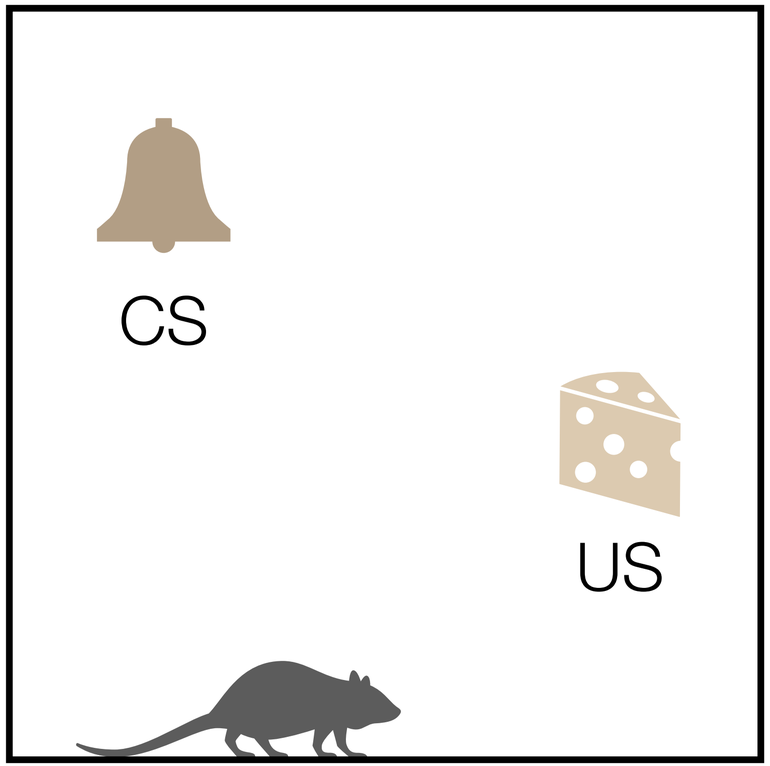
March 15, 2024

The RL problem: given task, how to find the right actions to maximise reward?

Plan for today

- Classical conditioning and associative learning theory
  - (Learning under uncertainty)
- Model-free reinforcement learning
  - TD learning in the brain
- Model-based learning and beyond

Pavlovian conditioning

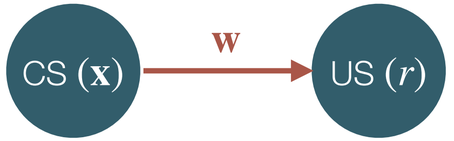


Rescorla-Wagner model of associative learning

Rescorla & Wagner (1972)

1. Reinforcement learning is about learning associations between states, actions and rewards

2. Learning these associations is driven by prediction errors



$$\mathbf{w}_{n+1} = \mathbf{w}_n + \alpha \mathbf{x}_n \delta_n$$
$$v_n = \mathbf{w}_n^T \mathbf{x}_n, \delta_n = r_n - v_n$$

Examples: what does the Rescorla-Wagner model predict?

A —> +
AB —> +
B —> ?

Forward blocking (Kamin, 1969)

AB—> +
B —> ?

Overshadowing (Mackintosh, 1971)

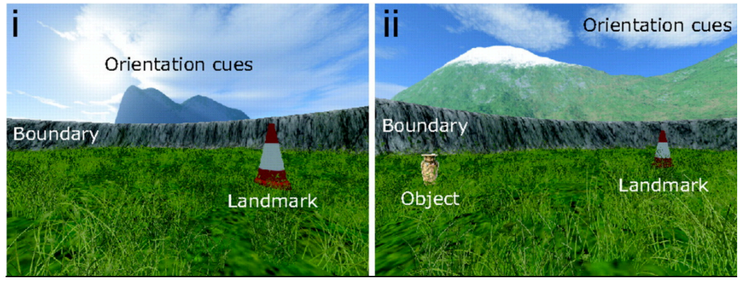
$$\mathbf{w}_{n+1} = \mathbf{w}_n + \alpha \mathbf{x}_n \delta_n$$
$$v_n = \mathbf{w}_n^T \mathbf{x}_n, \delta_n = r_n - v_n$$

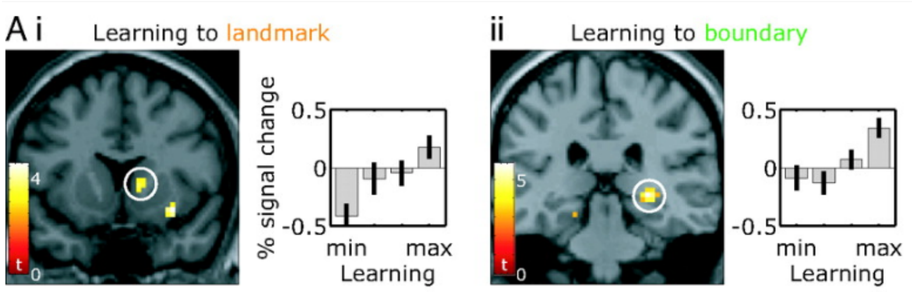
A—>+
B—>+
AB—>+
B—>?

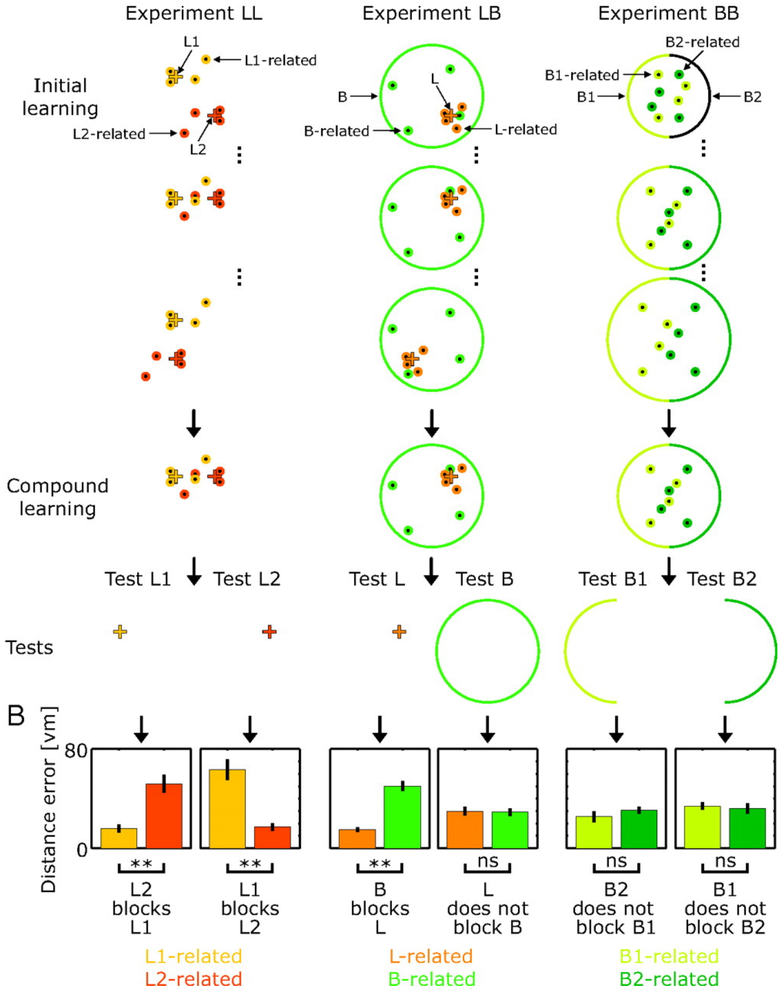
Overexpectation (Kamin & Gaioni, 1974)

Egocentric (landmark) navigation is prediction-error based but allocentric navigation is not

Doeller & Burgess (2008)







Normative motivation of Rescorla-Wagner

Gershman (2015)

$$\mathbf{w}_0 \sim \mathcal{N}\left(0, \sigma^2 I\right)$$
$$\mathbf{w}_n \sim \mathcal{N}\left(\mathbf{w}_{n-1}, \tau^2 I\right)$$
$$r_n \sim \mathcal{N}\left(\mathbf{w}_n^T \mathbf{x}_n, \sigma_r^2\right)$$

Generative model of rewards

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \alpha \mathbf{x}_n \delta_n$$
$$v_n = \mathbf{w}_n^T \mathbf{x}_n, \delta_n = (r_n - v_n)$$

Rescorla-wagner maximises the log-likelihood!

Bayesian version of associative learning: the Kalman Filter

- Animals should be representing a probability distribution over $\mathbf{w}$
- The can use Bayes' rule to infer posterior distribution over weights:
  - $p\left(\mathbf{w}_n \mid \mathbf{x}_{1:n}\right) \propto p\left(\mathbf{x}_{1:n} \mid \mathbf{w}_n\right) p\left(\mathbf{w}_n\right)$
- Model characteristics:
  - Uncertainty grows over time
  - Higher uncertainty = faster learning
  - Learn about absent features in the case of nonzero covariance

Kalman Filter equations

$$\hat{\mathbf{w}}_{n+1} = \hat{\mathbf{w}}_n + \mathbf{k}_n(r_n - v_n)$$

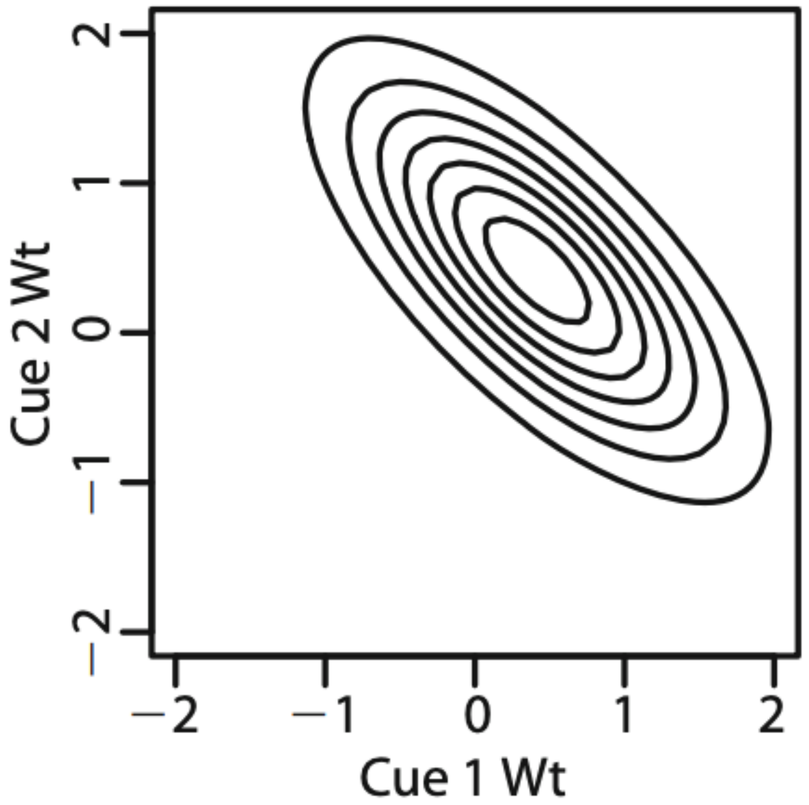
$$\Sigma_{n+1} = (I - \mathbf{k}_n \mathbf{x}_n^T)\Sigma_{n|n-1}$$

$$\mathbf{k}_n = \Sigma_{n|n-1}\mathbf{x}\left(\mathbf{x}^T\Sigma_{n|n-1}\mathbf{x} + \sigma^2\right)$$

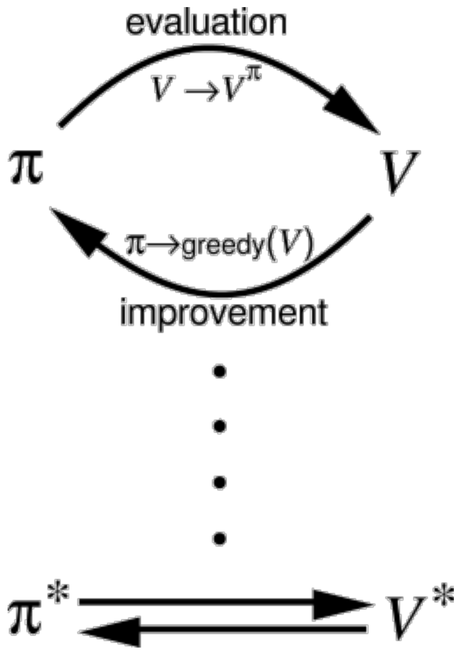
Backward blocking

Kruschke (2008)

AB —> +
A —> +
B —> ?



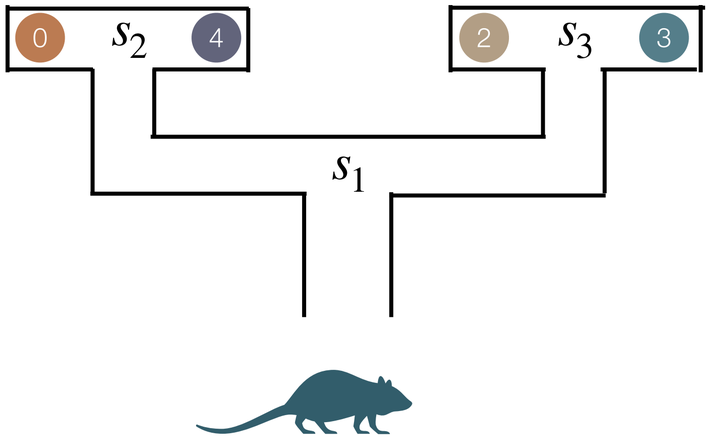
The RL problem: given task, how to find the right actions to maximise reward?



Value = cumulative future reward:

$$V(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,|\, s_0 = s\right]$$

Habitual versus goal-directed behaviour

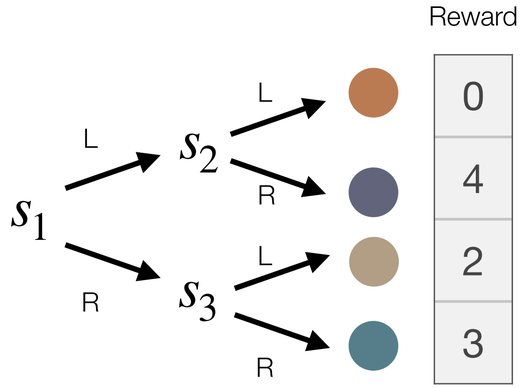


Fast
Inflexible

$Q(s, a)$

	L	R
S_1	4	3
S_2	0	4
S_3	2	3

Model-free



Model-based

Slow
Flexible

Model-free reward prediction

- Value function: $V_t = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k}\right]$
- Bellman equation: $V_t = r_t + \gamma \mathbb{E}\left[V_{t+1}\right]$
- Prediction error: $\delta_t = r_t + \gamma \mathbb{E}\left[V_{t+1}\right] - V_t$

Temporal difference learning

Sutton & Barto (1981)

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha \mathbf{x}_t \delta_t$$
$$V_t = \mathbf{w}_t^T \mathbf{x}_t, \delta_t = r_t + \gamma V_{t+1} - V_t$$

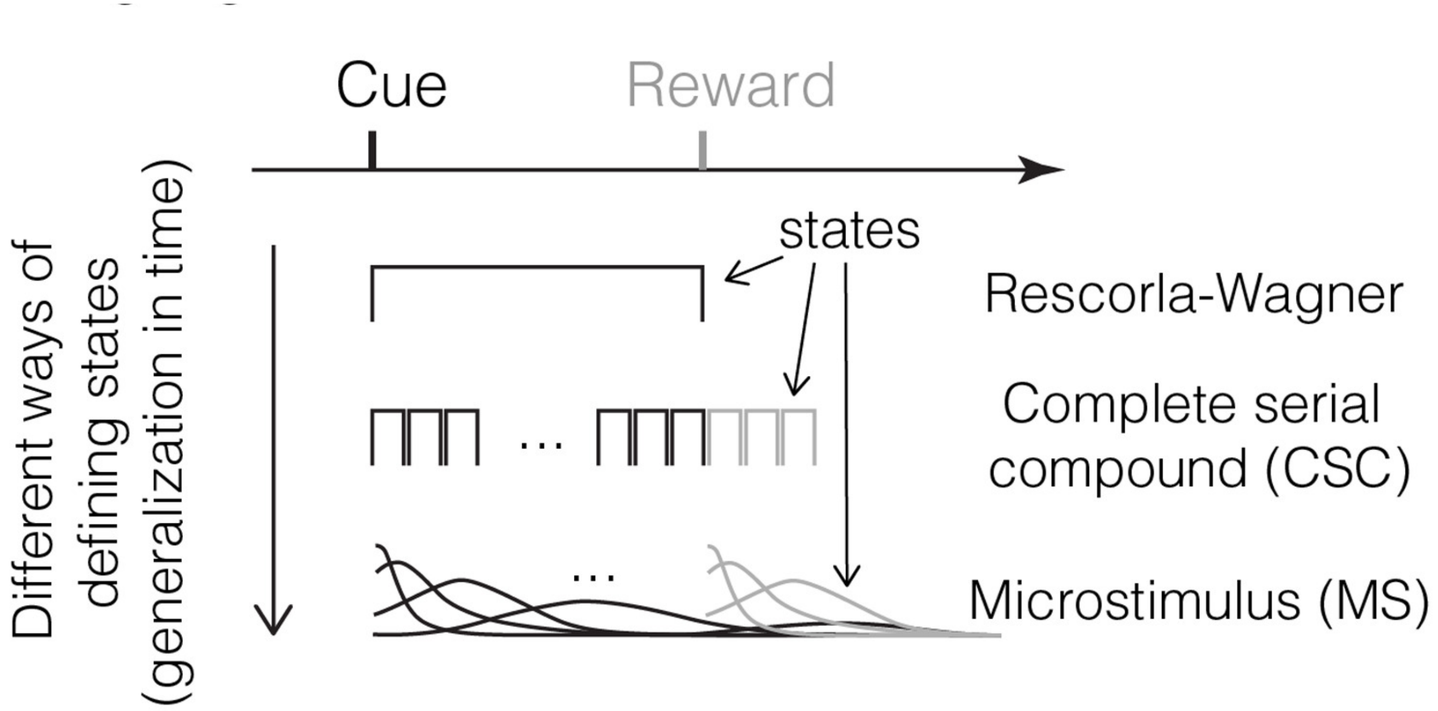
TD learning

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \alpha \mathbf{x}_n \delta_n$$
$$v_n = \mathbf{w}_n^T \mathbf{x}_n, \delta_n = r_n - v_n$$

Rescorla-Wagner

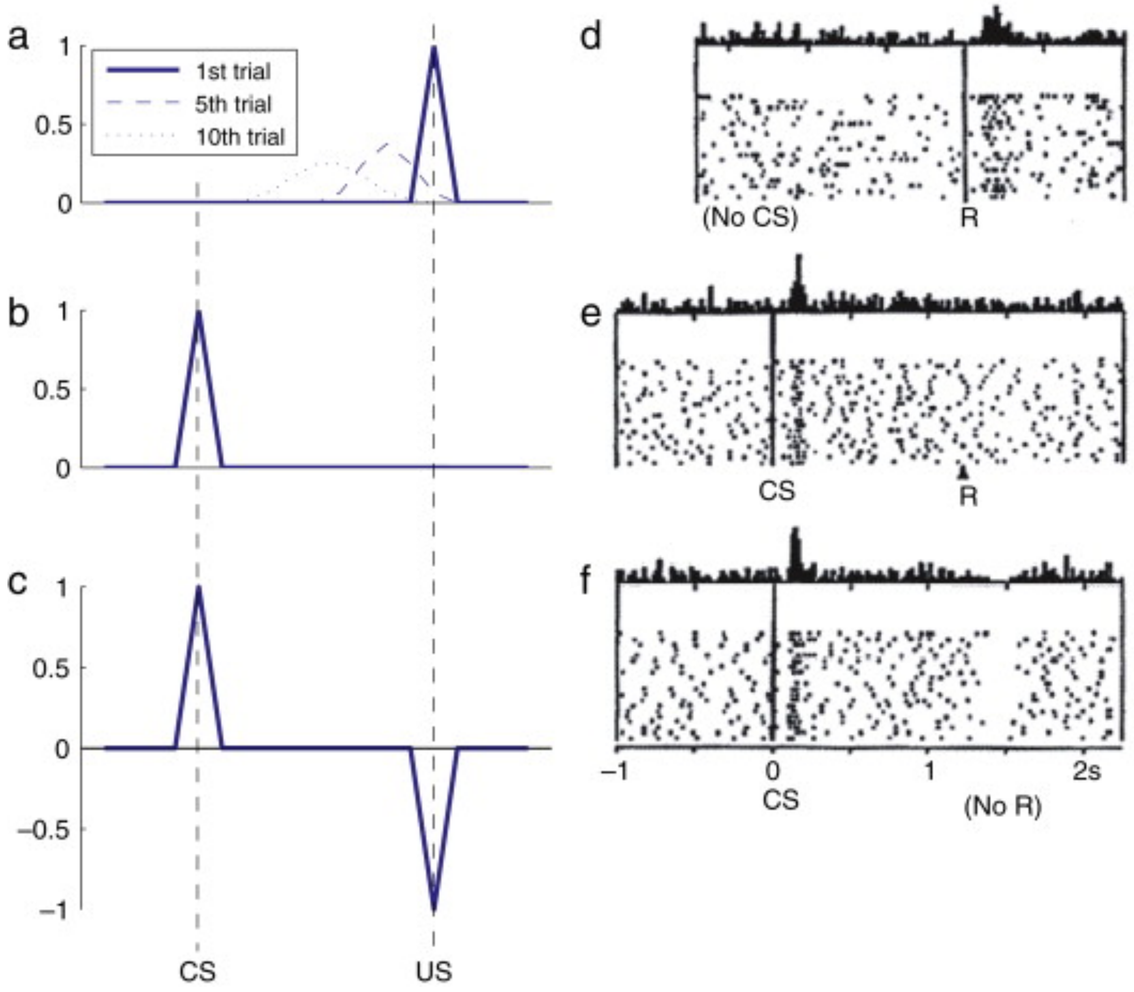
How are stimuli represented?

Jeong et al. (2022)



Dopamine as TD reward prediction error

Niv (2009); Schultz, Dayan & Montague (1998)



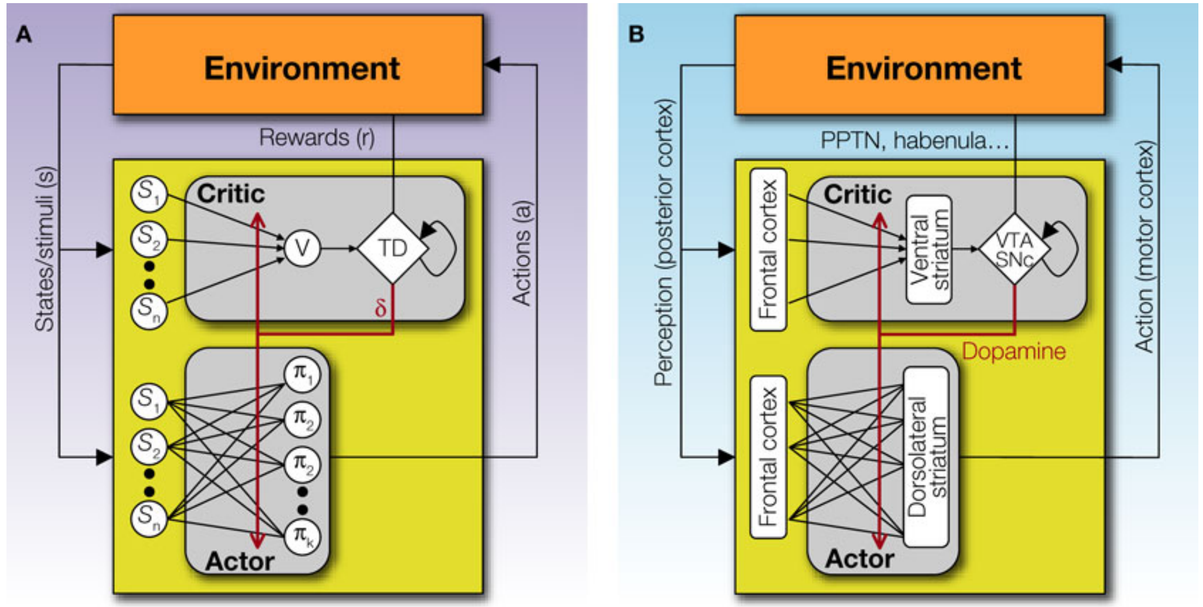
But how to select actions? Actor-critic model of basal ganglia

- Prediction error
  $$\delta_t = r_t + \gamma \hat{V}(s_{t+1}) - \hat{V}(s_t)$$

- Can be used to update “critic” weights:
  $$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha \mathbf{x}_t \delta_t$$

- And also the “actor” weights that parameterise the policy:
  $$\boldsymbol{\theta}^{a'}_{t+1} = \theta^{a'}_t + \alpha \delta_t \left(\mathbb{I}(a_t = a') - \pi(a' \mid s) \right)$$

- Select actions with softmax



Takahashi et al. (2008)

Model-free RL and Marr's level of analysis

Marr (1982)

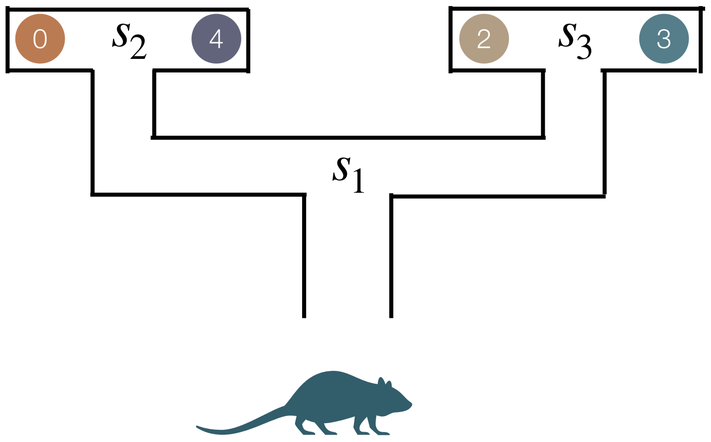
Computational theory	Representation and algorithm	Hardware implementation
What is the goal of the computation, why is it appropriate, and what is the logic of the strategy by which it can be carried out?	How can this computational theory be implemented? In particular, what is the representation for the input and output, and what is the algorithm for the transformation?	How can the representation and algorithm be realized physically?

Animals maximise reward

Using TD learning

Actor-critic architecture in the striatum with dopamine prediction errors

Habitual versus goal-directed behaviour

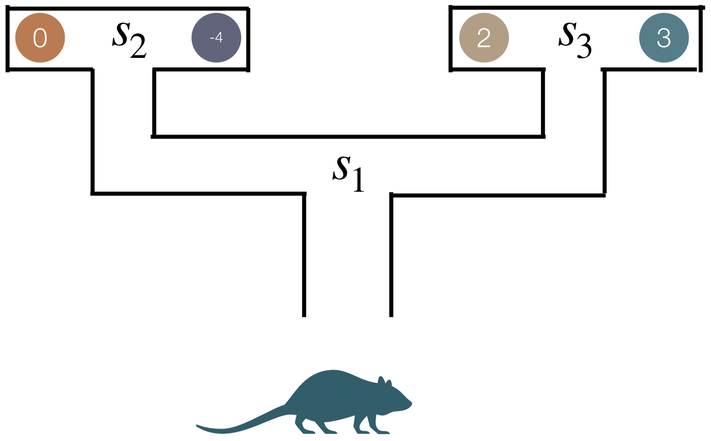


$Q(s, a)$

	L	R
S_1	4	3
S_2	0	4
S_3	2	3

Model-free

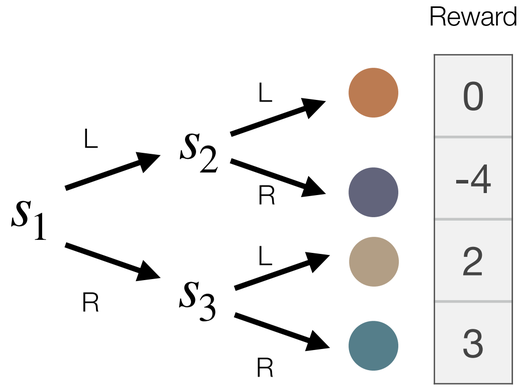
Habitual versus goal-directed behaviour



$Q(s, a)$

	L	R
S_1	4	3
S_2	0	-4
S_3	2	3

Model-free



Model-based

What is a “model” and how do we learn it

- Model-based methods learn a transition function $T(s' \mid s, a)$ and a reward expectation $\hat{R}(s, a)$

. $\hat{R}(s, a)$ can be updated with a simple delta rule: $\hat{R}(s) \rightarrow \hat{R}(s) + \alpha \left(r_t - \hat{R}(s_t) \right)$

- Similarly we can update the transition matrix:
  $T(s' \mid s, a) \leftarrow T(s' \mid s, a) + \alpha \left(\mathbb{I}(s_{t+1} = s') - T(s' \mid s, a) \right)$

- Planning is just repeated application of the transition matrix

A middle ground: the successor representation

Dayan (1993)

- The infinite discounted sum of n-step transition matrices gives a steady state distribution over states we'll visit: $M^{\pi} = \sum_{t=0}^{\infty} \gamma^t T_{\pi}^t = (I - \gamma T_{\pi})^{-1}$

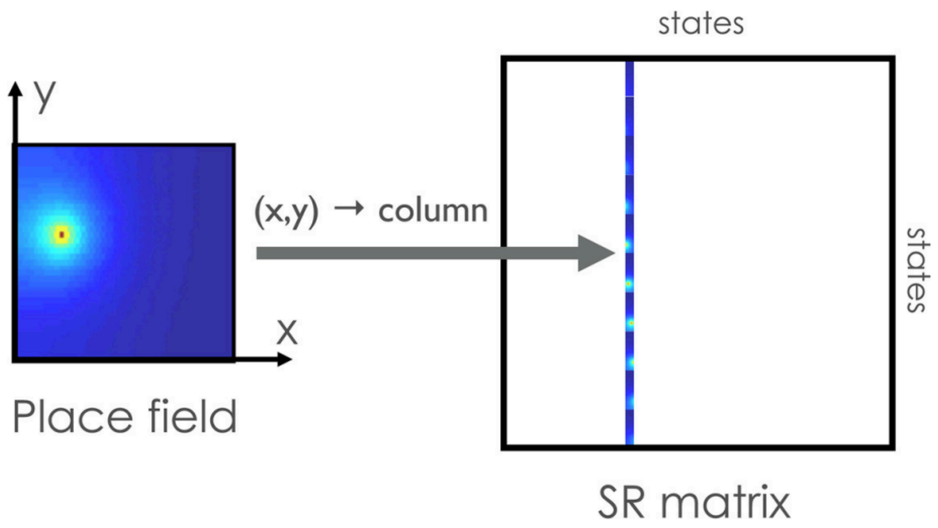
- This quantity is known as the *successor representation*

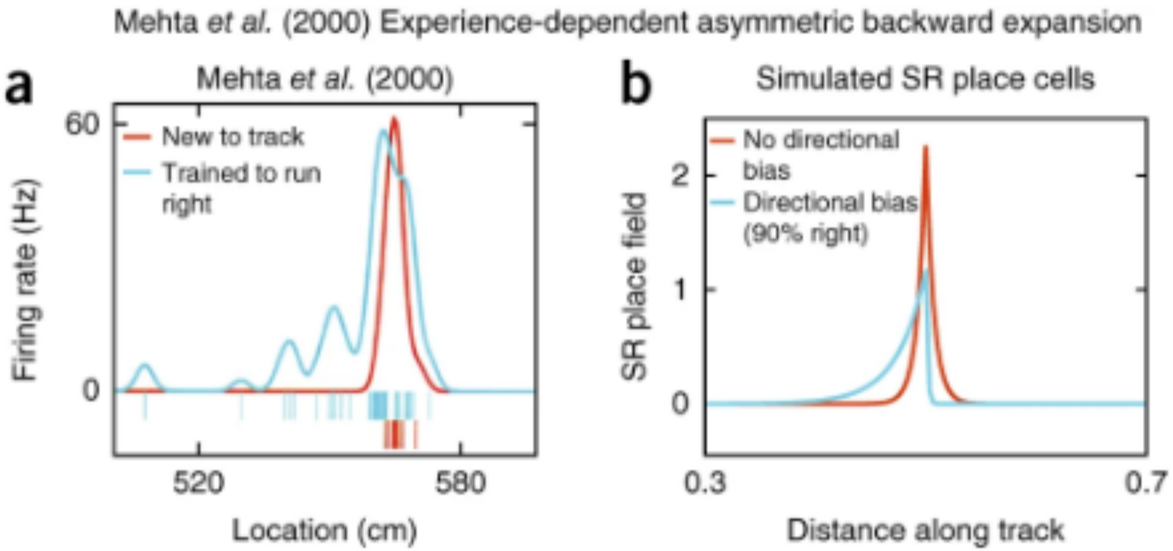
- When we have this and a reward estimate for each state, we can easily compute value: $V^{\pi}(s) = \sum_{s'} M^{\pi}(s, s') R(s')$

- SR can be estimated using TD learning:
  $\Delta \hat{M}(s_t, s') = \alpha \left[\mathbb{I}(s_t = s') + \gamma \hat{M}(s_{t+1}, s') - \hat{M}(s_t, s') \right]$

SR as a model of place cells

Stachenfeld et al. (2017)





Summary

- Rescorla-Wagner predicts animal behaviour at the trial level
- Model-free TD learning is a good model for sequential decision making
    - Actor-critic models of the basal ganglia span Marr's levels
- Model-based learning is more flexible but harder and less well understood
    - There are alternative strategies that lie in between these two extremes
- For neuroscientists, RL gives a formal (normative) framework to model autonomous learning from neurons to behaviour