Reinforcement Learning

AIMS

- Discuss formal models of classical and instrumental conditioning in animals
- Describe how reinforcement learning (e.g. using the temporal difference learning rule) solves the 'temporal credit assignment' problem in learning to act from infrequent reward.
- Describe how the involvement of neuromodulators, such as dopamine, in reward and punishment learning is included in these models.

READING

- For modeling: Chapter 9, Dayan & Abbott, "Theoretical Neuroscience" (but v mathematical);
- For dopamine: Schultz W. 2002 Getting formal with dopamine and reward. Neuron 36: 241-63.
- For algorithms: Sutton RS & Barto AG "Reinforcement learning: An Introduction"

Reinforcement Learning

How to learn to make decisions in sequential problems
(like: chess, a maze)

Why is this difficult?
Temporal credit assignment

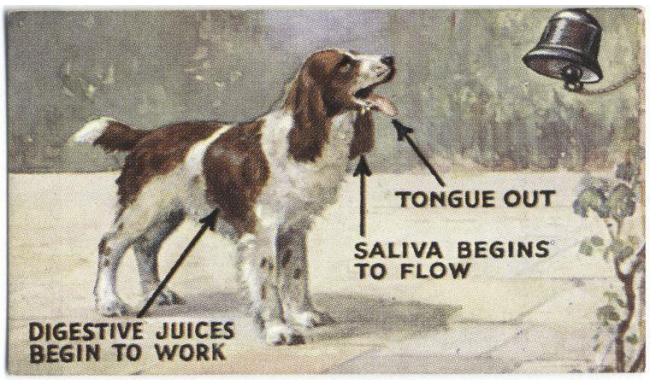
Prediction can help

Classical conditioning

learning

CS: Conditioned Stimulus (S)

Pair stimulus (bell, light)



US: Unconditioned stimulus (reinforcement r)

…with significant event (food, shock)

CR: Conditioned response

Measure anticipatory behavior (salivation, freezing)

Rescorla-Wagner rule (1972)

Experimental terms

	Phase 1:	**Phase 2:**	**Test:**
Acquisition:	S → r		S? response
Extinction:	S → r	S → -	S? –
Partial reinforcement:	S → r or -		S? weak resp

Simple “delta-rule” model,
if stimulus present S=1 (S=0 if not present):

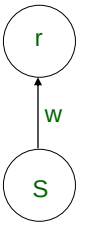
$w \rightarrow w + \varepsilon S\delta; \quad \delta = r - wS,$

Learning reduces “reward prediction error”

Equivalently, if S=1: $w \rightarrow (1-\varepsilon)\, w + \varepsilon r$

w comes to estimate the reinforcement r associated with S.

Minimizes ‘error’ $\langle (r-wS)^2 \rangle$ whenever S is present.



Reinforcement (aka reward, or punishment)

Stimulus

Rescorla-Wagner rule (1972)

Experimental terms

	Phase 1:	**Phase 2:**	**Test:**
Acquisition:	S → r		S? response
Extinction:	S → r	S → -	S? –
Partial reinforcement:	S → r or -		S? weak resp

Simple "delta-rule" model,
if stimulus present S=1 (S=0 if not present):

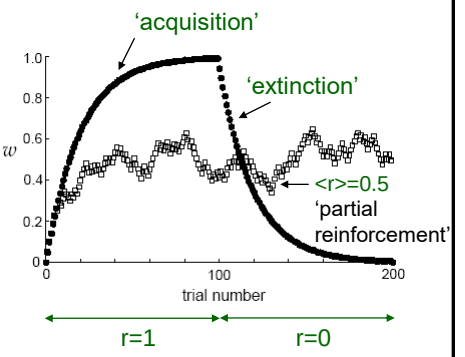
$w \rightarrow w + \varepsilon S\delta$; $\delta = r - wS$,

Learning reduces "reward prediction error"

Equivalently, if S=1: $w \rightarrow (1-\varepsilon)\, w + \varepsilon r$

w comes to estimate the reward r

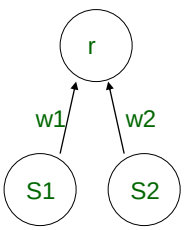
associated with S.



Multiple stimuli

What about when multiple stimuli are present? e.g. S1, S2→r

How would animals respond to S1 or S2?
How should the model be modified?



$w_i \rightarrow w_i + \varepsilon\, S_i\, \delta_i$

(a) $\delta_i = r - w_i S_i$

i.e. separate error terms for each S_i

(b) $\delta_i = \delta = r - V$; $V = \Sigma_i\, w_i S_i$

V is expected reinforcement r given all stimuli

i.e. single error term for all stimuli S_i:

δ is the difference between actual r and V (expected r)

(aka "reward prediction error").

Experiments with multiple stimuli

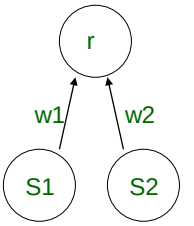
Experimental terms	Phase 1:	Phase 2:	Test:
Overshadowing:	S1, S2 → r		S1? weak resp
Blocking:	S1 → r	S1, S2 → r	S2? –

Which model is favoured?

$w_i \rightarrow w_i + \varepsilon S_i \delta_i$

Blocking (Kamin, 1969) and *overshadowing* (Kamin, 1969; Pavlov, 1927) imply:

(b) $\delta_i = \delta = r - V$; $V = \Sigma_i w_i S_i$



i.e. single error term for all stimuli (the Rescorla-Wagner rule)
= difference between reinforcement and expected reinforcement given all stimuli (aka "reward prediction error")

Second-order conditioning

Phase 1:	Phase 2:	Test:
S1→ r	S2 → S1	S2 ?

What happens? S2 → r

R-W rule does not work – no r in Phase 2

One problem: time (important that S2 precede S1 for effect)
=> the temporal credit assignment problem.

Temporal-difference learning (Sutton)

We need V(t) to predict the sum of future rewards, not just r(t), so we can learn S2 → S1 → R,

i.e. we want $V(t) = < \Sigma_{\tau \geq t} r(\tau) >$

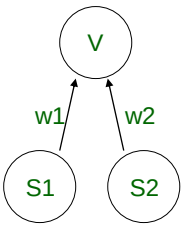
If $V(t) = \Sigma_i w_i(t) S_i$.

How can we learn the right w_i?

If $V(t) = < \Sigma_{\tau \geq t} r(\tau) >$, then $V(t) = r(t) + V(t+1)$

current reward

estimate of subsequent reward



So use delta rule to ensure that this happens, i.e. modify connection weights to make V(t) closer to r(t) + V(t+1), i.e. use:

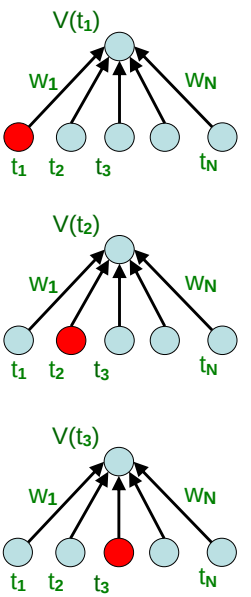
$w_i(t+1) = w_i(t) + \varepsilon s_i \delta(t)$; $\delta(t) = [r(t) + V(t+1)] - V(t)$

so that δ is the difference between V(t) and estimate of all future reward.

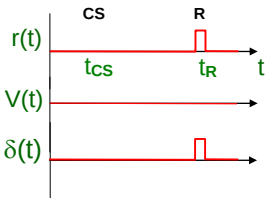
Compare with R-W rule: $d(t) = r(t) - V(t)$ (i.e. δ is the difference between V(t) and current reward only).

Time: expectation of reward

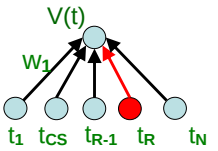
Need representation of time as an input.
Code time as sequential activity of set of input neurons.
Then output can represent V(t)

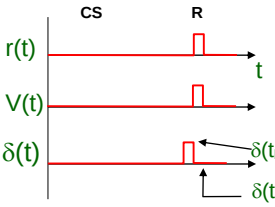


Temporal difference learning rule: $w_i \rightarrow w_i + \varepsilon s_i \delta(t_i)$; $\delta(t) = r(t)+V(t+1)-V(t)$

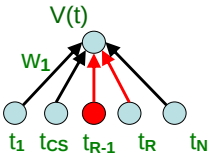


trial 1
unexpected reward

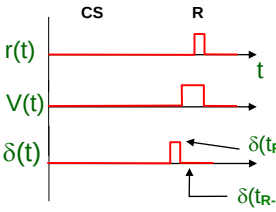




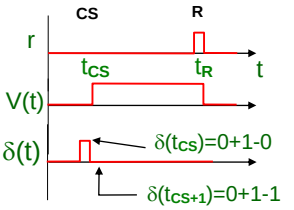
trial 2



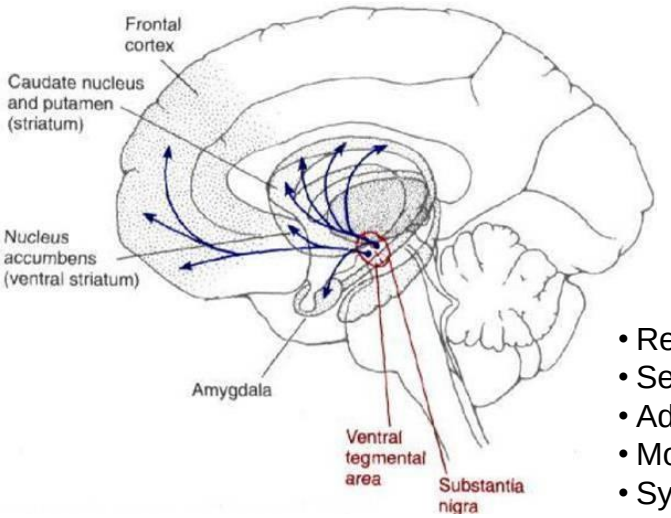
Many trials later, expected future reward V increases as soon as the CS occurs..



trial 3



Does dopamine signal $\delta(t)$?



- Reward
- Self-stimulation
- Addiction
- Motor control
- Synaptic plasticity

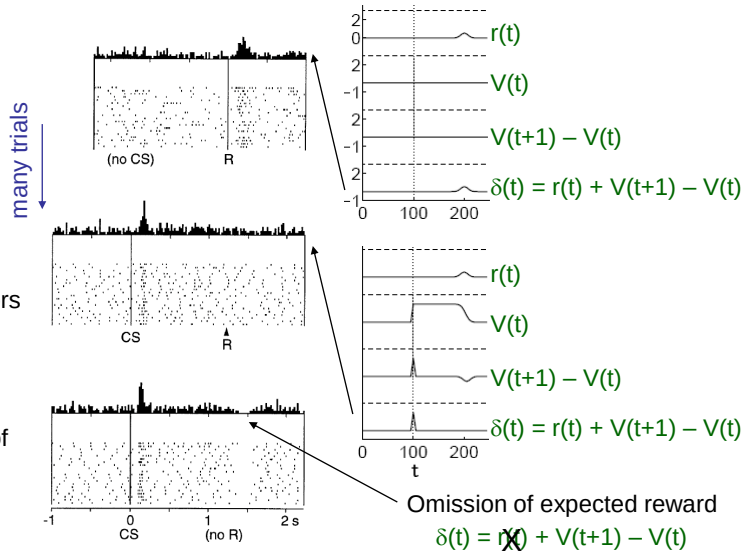
Dopamine responses interpreted as $\delta(t)$

(Schultz, Montague & Dayan, Science, 1997)

Burst to unexpected reward

Response transfers to reward predictors

Pause at time of omitted reward

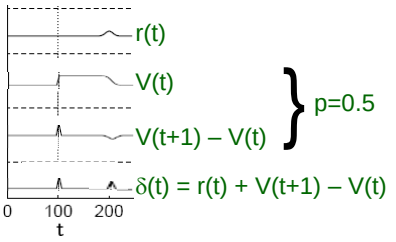


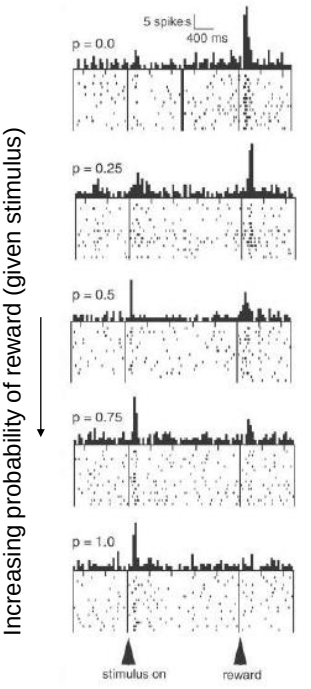
Omission of expected reward

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

More dopamine responses

- Partial reinforcement task (Fiorillo, Tobler & Schultz)
- Accords with TD models





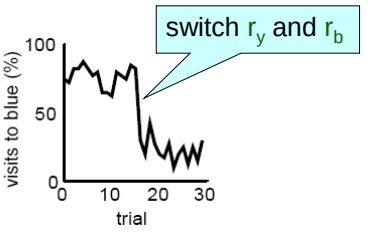
Action choice

In operant (aka instrumental) conditioning, rewards are contingent on **actions** (e.g., lever press)

– cf. classical conditioning (rewards are just dependent on stimuli, we just model expectation of reward)

Consider simple bee foraging problem:

- Choose between yellow and blue flowers
- Each pays off probabilistically, with different amounts of nectar r_y versus r_b
- Bees rapidly learn to choose richer colour

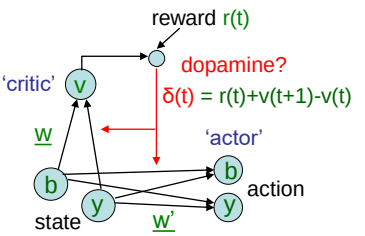


Modeling action choice

"Actor-critic" architecture: use value function V to decide on actions to maximize expected reward.

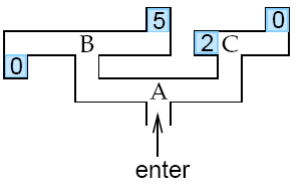
If V is correct, it allows future reward to be taken into account in reinforcing actions: solving the temporal credit assignment

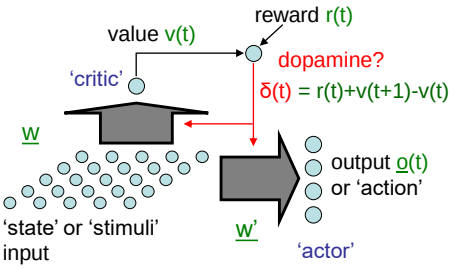
(we can also evaluate whether actions are good or bad using δ(t) even if r(t) = 0)



Sequential choice

- Suppose we know how to learn what to do at B and C (e.g. choose action associated with maximum r)
- How do we solve the temporal credit assignment problem at A?
- Requires second order conditioning: A-B-r; A-C-r
- Use Temporal Difference learning to learn values (expected future reward) at B and C; use these to learn best action at A ('direct actor')
- "actor/critic" architecture.





State evaluation

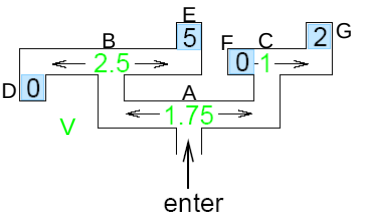
The critic learns to estimate value states $S_i(t)$:

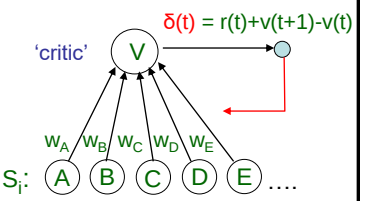
$V(t) = \langle \Sigma_{\tau>t}\, r(\tau) \rangle$

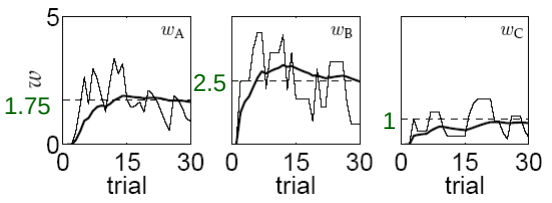
Given initially random choice of actions L/R ('policy'), values V are learned by changing w_i with Temporal Difference rule:

$w_i \rightarrow w_i + \varepsilon s_i \delta(t)$;

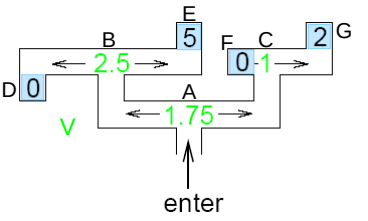
$\delta(t) = r(t)+V(t+1) - V(t)$







Policy improvement

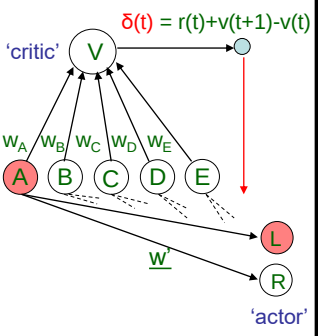


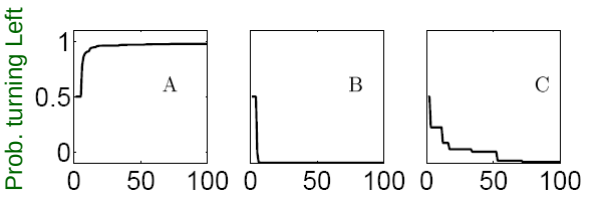
The actor learns to act, using V(t) to calculate δ and δ to assess actions: $\delta(t) = r(t)+V(t+1)-V(t)$

- If left at A, $\delta(t) = 0 + 2.5 - 1.75 = 0.75$
- If right at A, $\delta(t) = 0 + 1 - 1.75 = -0.75$

Thus should choose left more frequently from A

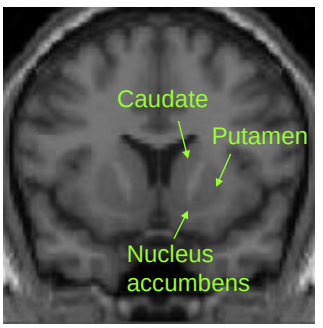
$$w'_i \rightarrow w_i' + \varepsilon s_i \delta(t);$$

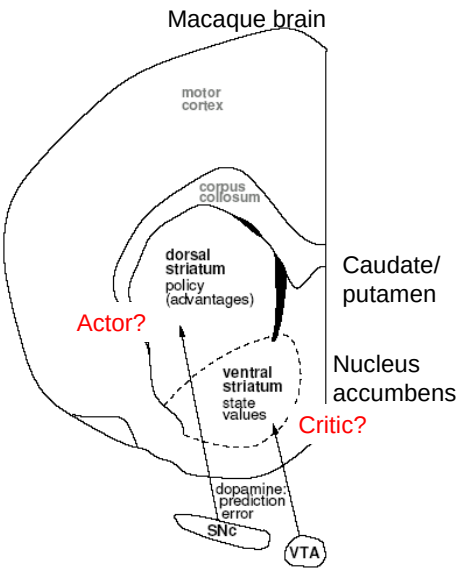




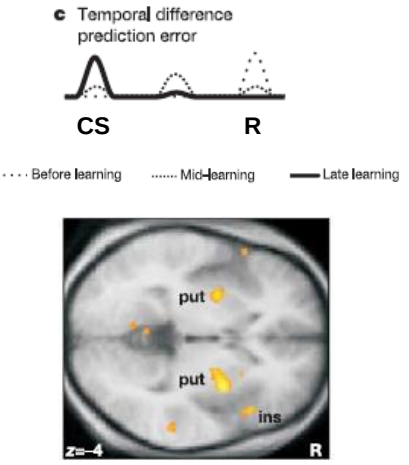
Back to dopamine

- Dual dopamine systems project same signal to motivational (ventral striatum) & motor (dorsal striatum) areas
- For state evaluation & policy improvement respectively?

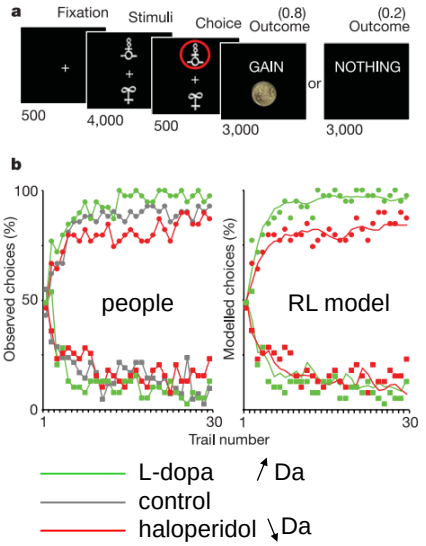




Striatal targets of DA ~ “prediction-error” signal for reinforcement learning



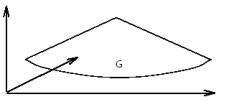
(Seymour et al., 2004)



(Pessiglione et al., 2006)

Summary of ‘temporal difference’ or ‘sequential reinforcement’ solution to the temporal credit assignment problem

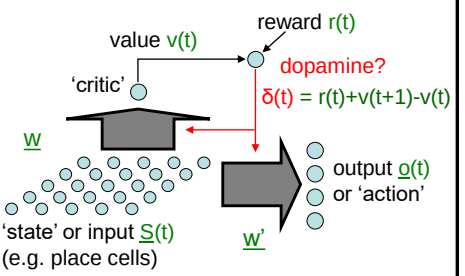
In A_{rp} the strength of the smell of the tree gives reinforcement r(t) to evaluate any action (i.e. output o(t)): does it lead closer to the goal or not?



If reinforcement r(t) is intermittent (e.g. only when goal is reached), a ‘critic’ learns an ‘*evaluation function*’: the value v of each state (or input) S is *the expected future reward from that state* (given how actions are usually made).

The change in value v(t+1)-v(t) + any reward r(t) is used to evaluate any action o(t) so output weights w’ can be modified as in A_{rp}, using $\delta(t) = v(t+1)-v(t)+r(t)$ (if $\delta(t)>0$, action o(t) good):
$\underline{w} \rightarrow \underline{w} + \varepsilon \underline{S}(t)\delta(t)$

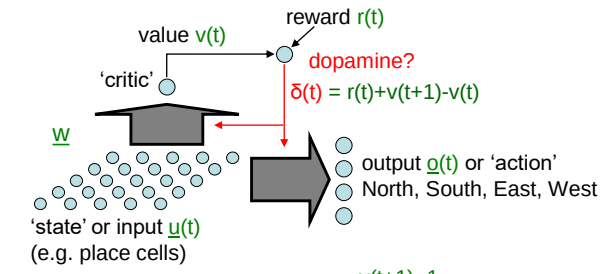
But how to learn v? A simple learning rule creates connection weights w so that $v(t) = \underline{w}.\underline{S}(t)$.
This is: $\underline{w} \rightarrow \underline{w} + \varepsilon \underline{S}(t)\delta(t)$, i.e. you can also use δ to learn weights for the critic!

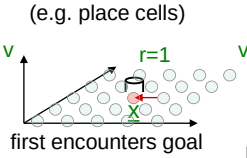


See Intro. to temporal difference learning (spatial example).

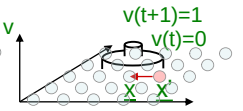
Use $\delta(t) = v(t+1)-v(t)+r(t)$ to evaluate actions

Use $v(t) = \underline{w}.\underline{S}(t)$ where $\underline{w} \rightarrow \underline{w} + \varepsilon \underline{S}(t)\delta(t)$ to learn value function.

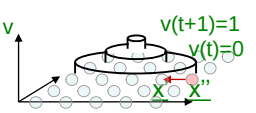




first encounters goal at location x: r=1 so increase weight to critic from place cell at x. Now v(x)=1

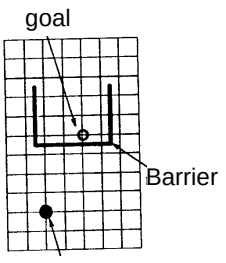


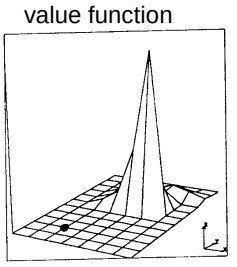
next time go to x from x', v(t+1)-v(t)=1, so increase weight to critic from place cell at x'. Now v(x')=1



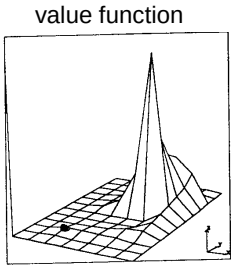
next time go to x' from x'', v(t+1)-v(t)=1, so increase weight to critic from place cell at x'', and so on..

Temporal difference learning is a slow 'trial and error' based method. Development of own value function overcomes problem of temporal credit assignment





After initial exploration (e.g. 100 runs to goal)



After 2000 runs to goal

Initial actions will be random (takes a long time to get to goal), actions improve as a result of learning, as in the Arp action network (but a random element is required to enable ongoing improvement: i.e. finding new shorter routes).

Dayan (1991) Neural Information Processing Systems 3. p464-70. Morgan Kaufmann

Temporal discounting

Often, we value more (temporally) distant rewards less than immediate reward of the same size. We can do this with a small change to the Reinforcement Learning rule:

We want $V(t)$ to predict the sum of future rewards, discounted by how long you have to wait for them,

i.e. we want $V(t) = < \Sigma_{\tau \geq t} \gamma^{(\tau - t)} r(\tau) >$ where $\gamma < 1$

If $V(t) = < \Sigma_{\tau \geq t} \gamma^{(\tau - t)} r(\tau) >$, then $V(t) = r(t) + \gamma V(t+1)$

So use delta rule to ensure that this happens, i.e. modify connection weights to make $V(t)$ closer to $r(t) + \gamma V(t+1)$, i.e. use:

$w_i(t+1) = w_i(t) + \varepsilon s_i \delta(t)$; where $\delta(t) = [r(t) + \gamma V(t+1)] - V(t)$

so that δ is the difference between $V(t)$ and the estimate of all (temporally discounted) future reward.

Undergraduate BSc and 4th year MSci students: There is a course essay and a 3 hour exam.

The course essay consists of analysing a research paper, max. 2,000 words.

Papers for essay available on: https://www.ucl.ac.uk/icn/neur0016-neural-computation-models-brain-function. The essay constitutes 10% of the final mark for the course. The exam constitutes the remaining 90% of the final mark for the course.

MSc students, and affiliate students (leaving before May): One 3,000 word essay, chosen from these titles:

1. Can a mechanistic neuron-level understanding of some aspects of cognition be attained?
2. Discuss the approximations made in computational approaches to understanding the functional properties of networks of neurons, including when and how they have proved to be useful.
3. Describe examples where understanding of the electrophysiological behaviour of neurons allows increased understanding of the behaviour of the organism.

The deadline for essays is 2.00pm Tuesday January 14th 2020

(except where an MSc administrator has agreed otherwise)