Reinforcement Learning: Time for Action!

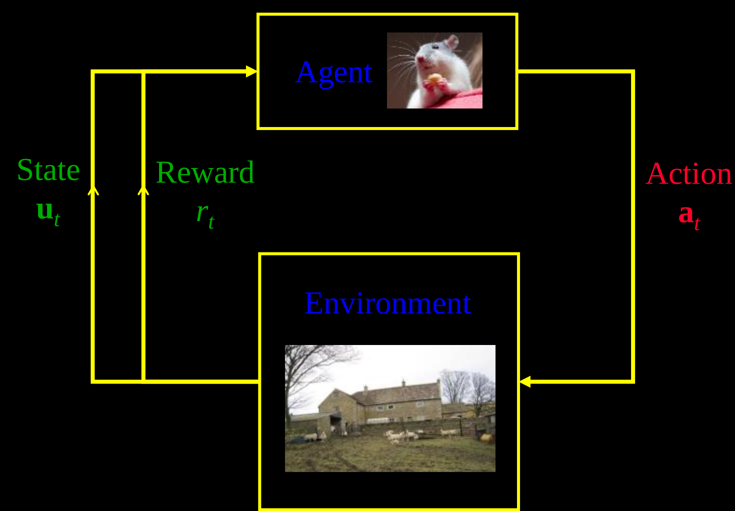
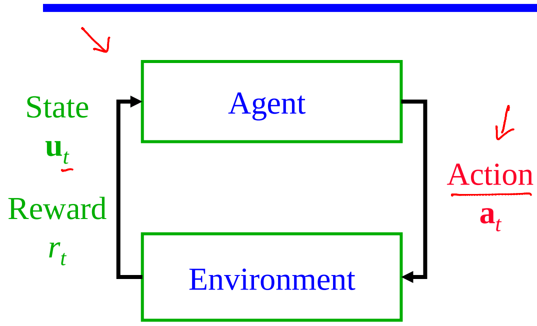


Image Source: Wikimedia Commons

The Problem



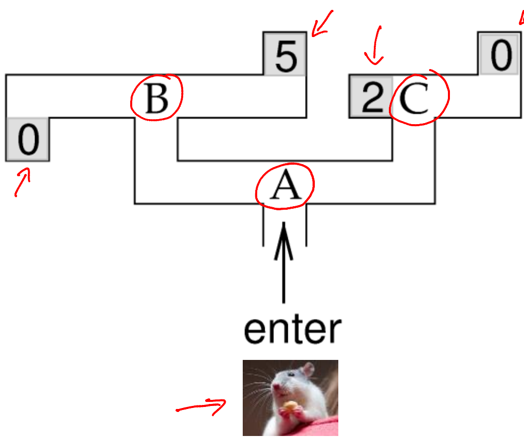
Learn a state-to-action mapping or "policy":

$$\pi(\mathbf{u}) = \mathbf{a}$$

which maximizes the expected total future reward:

$$\left\langle \sum_{\tau=0}^{T-t} r(t+\tau) \right\rangle_{trials}$$

Example: Rat in a barn



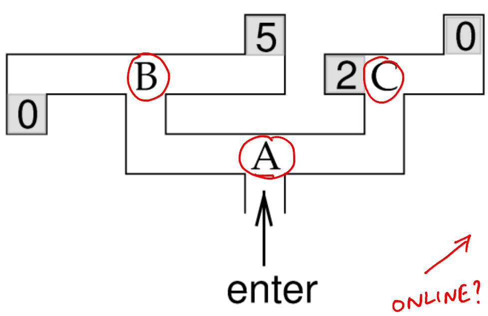
States = locations A, B, or C

Actions= L (go left) or R (go right)

If the rat chooses L or R *at random* (random "policy"), what is the expected reward (or "*value*") *v* for each state?

Image Source: Dayan & Abbott textbook

Policy Evaluation



ONLINE?

For random policy:

$$v(B) = \frac{1}{2}\cdot 0 + \frac{1}{2}\cdot 5 = 2.5$$

$$v(C) = \frac{1}{2}\cdot 2 + \frac{1}{2}\cdot 0 = 1$$

$$v(A) = \frac{1}{2}\cdot v(B) + \frac{1}{2}\cdot v(C) = 1.75$$

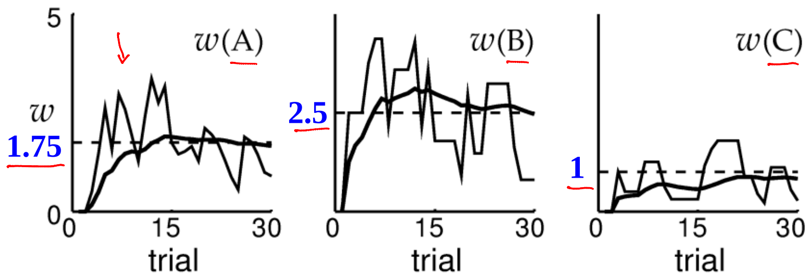
Let value of state *u* $v(u)$ = weight $w(u)$

Can learn value of states using TD learning:

$$w(u) \leftarrow w(u) + \varepsilon\,[r(u) + v(u') - v(u)]$$

(Location, action) ⇨ new location i.e., (u,a) ⇨ u'

TD Learning of Values for Random Policy

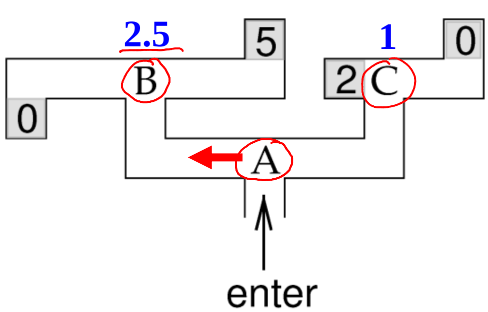


(For all three, $\varepsilon = 0.5$)

Once I know the values, I can pick the action that leads to the higher valued state!

Image Source: Dayan & Abbott textbook

Selecting Actions based on Values



Values act as surrogate immediate rewards → Locally optimal choice leads to globally optimal policy for "Markov" environments

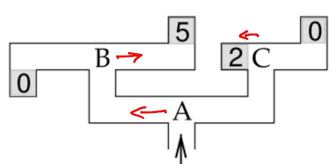
(Related to *Dynamic Programming*)

Putting it all together: Actor-Critic Learning

- Two separate components: Actor (selects action and maintains policy) and Critic (maintains value of each state)

1. Critic Learning ("Policy Evaluation"):
   Value of state $u = v(u) = w(u)$
   $$w(u) \leftarrow w(u) + \varepsilon\,[r(u) + v(u') - v(u)]$$ (same as TD rule)
2. Actor Learning ("Policy Improvement"):
   $$P(a;u) = \frac{\exp(\beta Q_a(u))}{\sum_b \exp(\beta Q_b(u))}$$ SOFTMAX
   EXPLORE
   Probabilistically select an action a at state u
   For all actions a':
   $$Q_{a'}(u) \leftarrow Q_{a'}(u) + \varepsilon[r(u) + v(u') - v(u)](\delta_{aa'} - P(a';u))$$
3. Repeat 1 and 2

Actor-Critic Learning in our Barn Example



Probability of going Left at each location

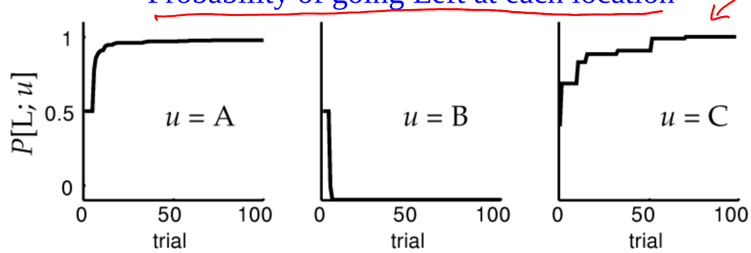
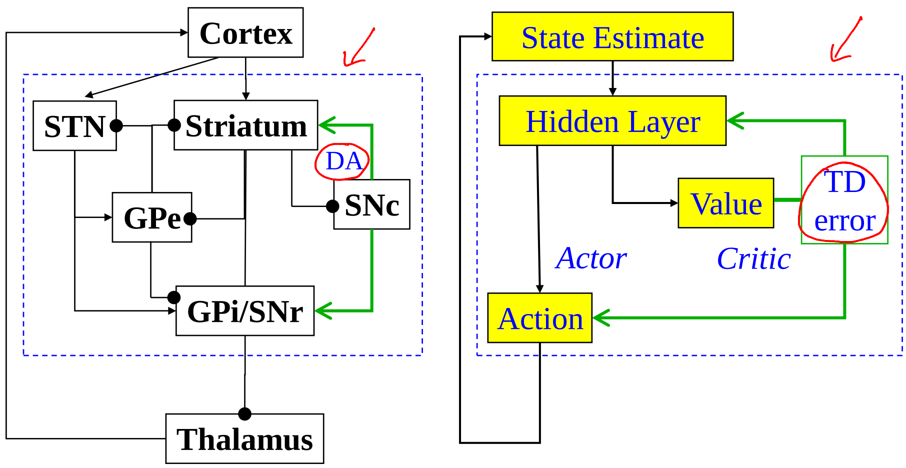


Image Source: Dayan & Abbott textbook

Possible Implementation of the Actor-Critic Model in the Basal Ganglia



(See Supplementary Materials for references)

Reinforcement learning has been applied to many real-world problems!

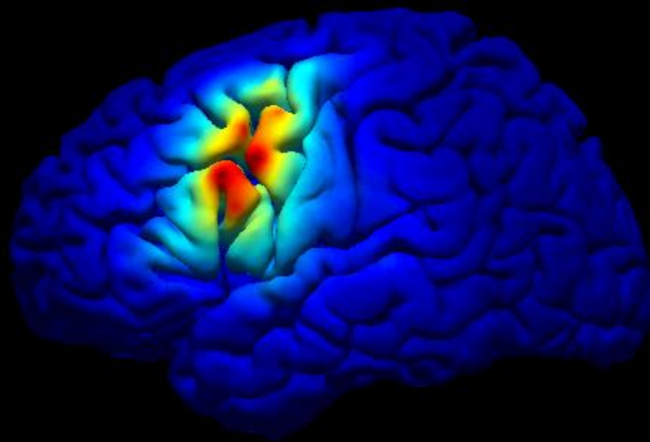
Example:

Autonomous Helicopter Flight

(learned from human demonstrations)

(Videos and papers at: http://heli.stanford.edu/)

Computational Neuroscience



Rajesh P. N. Rao

Adrienne Fairhall

University of Washington, Seattle, USA