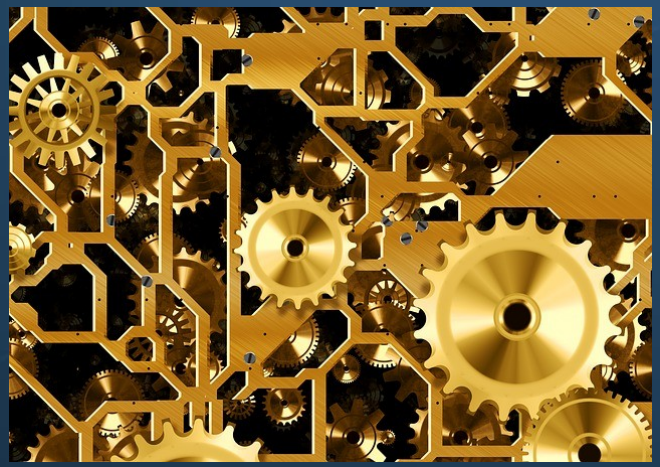
Deep Q Learning: From Paper to Code

Value Functions, Action Value Functions and the Bellman Equation

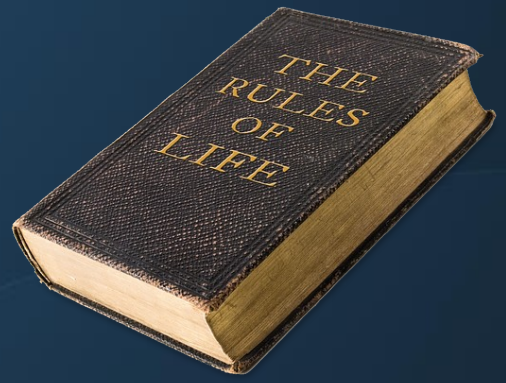
Last Time

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Maximize rewards over time



Works due to Markov property



Mapping of states to actions

Value Functions

Each time step carries reward

States & (states, actions) have value

Larger G_t larger value

Value Functions

$\Pi(s,a) \rightarrow$ probability of selecting a in s

$$v_\pi(s) = E_\pi[G_t | S_t = s] = E_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s] \; for\ all\ s \in S$$

$$q_\pi(s,a) = E_\pi[G_t | S_t = s, A_t = a] = E_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a] \; for\ all\ s \in S$$

Q as in Q learning!

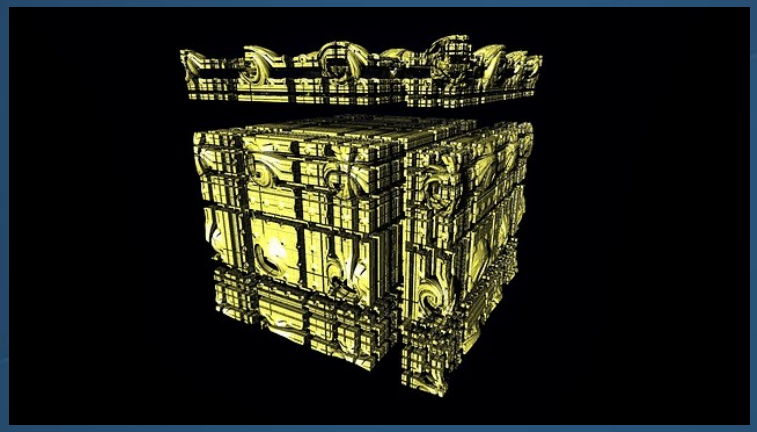
Expectation Values?

How to get E without probabilities?

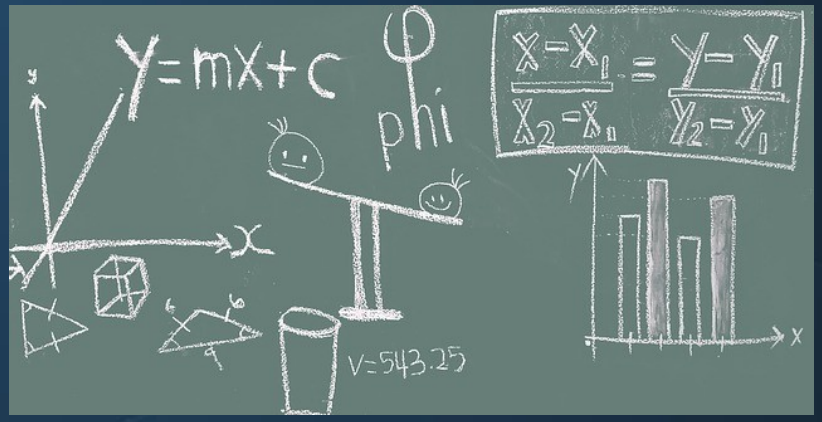
Interact with environment

Keep track of rewards

Problems Already ...



Not feasible for large state spaces



Parameterize state space

The Bellman Equation

$$v_\pi(s) = E_\pi[G_t | S_t = s] = E_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s] \; for\ all\ s \in S$$

$$G_t = R_{t+1} + \gamma G_{t+1}$$

$$v_\pi(s) = E_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s]$$

$$v_\pi(s) = \sum_a \pi(a,s) \sum_{s'} \sum_r p(s', r | s, a)[r + \gamma E_\pi[G_{t+1} | S_{t+1} = s']]$$

$$v_\pi(s) = \sum_a \pi(a,s) \sum_{s',r} p(s', r | s, a)[r + \gamma v_\pi(s')]$$

Bellman Equation

Don’t Panic

$$v_\pi(s) = \sum_a \pi(a,s) \sum_{s',r} p(s',r|s,a)[r + \gamma v_\pi(s')]$$

Bellman Equation

Recursive relationship between value functions

Makes life easier (believe it or not)

Rank Ordering Policies

- Can we rank policies using v?
- Tweak action and compare old vs. new
- Whole point of v and q is to rank policies!
- If one policy has a better v, then it is better
- At least one best policy → optimal policy

The Bellman Optimality Equations

$$v_*(s) = \max_{a \in A(s)} q_{\pi_*}(s,a)$$

$$v_*(s) = \max_a E_\pi[G_t | S_t = s, A_t = a]$$

$$v_*(s) = \max_a E_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a]$$

$$v_*(s) = \max_a E_\pi[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a]$$

$$v_*(s) = \max_a \sum_{s',r} p(s',r|s,a)[r + \gamma v_*(s')]$$

$$q_*(s,a) = \sum_{s',r} p(s',r|s,a)[r + \gamma \max_{a'} q_*(s',a')]$$

Summary

- Every policy has a v and q
- V and q obey the Bellman equation
- Policies can be ranked with v and q
- Bellman optimality equation is recursive

Up Next