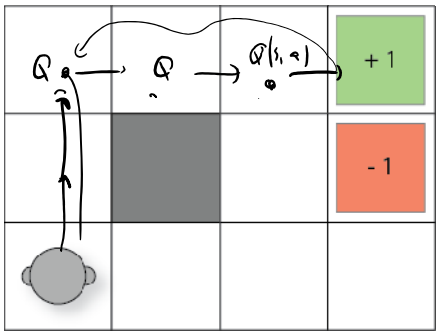
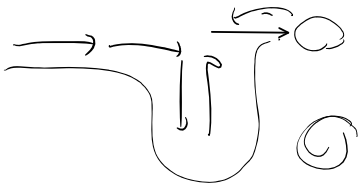
n-step temporal difference learning

Discounted future rewards

$G_t = r_t + \gamma V(s') = r_t + \gamma r_{t+1} + \gamma^2 V(s'') = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 V(s''')$

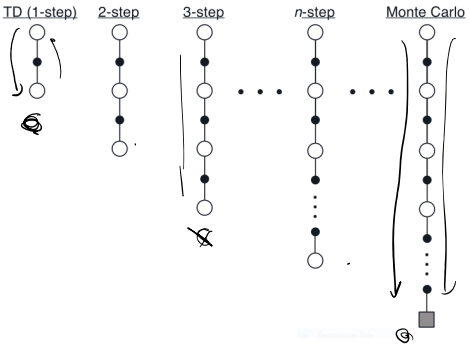


3 - step



$Q(s_t) \leftarrow Q(s_t) + \alpha(r + V(s'))$

Truncated future rewards



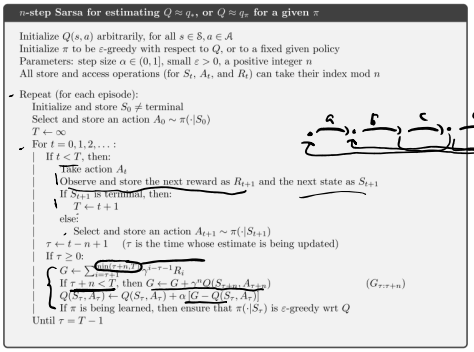
SARSA: Q(s,a) := Q(s,a) + α [r + γQ(s',a') - Q(s,a)]

Change the update:

$G_t \leftarrow r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} \cdots \gamma^n Q(s,a)$

$\sum_{i=t}^{i=t+n} \gamma^i r_i$

$Q(s,a) \leftarrow Q(s,a) + \alpha[G_t + \gamma^n Q(s',a')]$

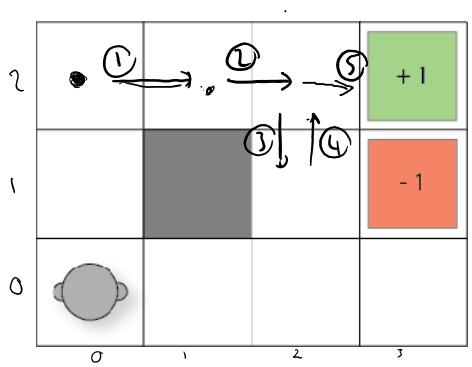


n-step Sarsa for estimating $Q \approx q_*$, or $Q \approx q_\pi$ for a given π

Initialize $Q(s,a)$ arbitrarily, for all $s \in \mathcal{S}, a \in \mathcal{A}$
Initialize π to be ε-greedy with respect to Q, or to a fixed given policy
Parameters: step size $\alpha \in (0,1]$, small $\varepsilon > 0$, a positive integer n
All store and access operations (for S_t, A_t, and R_t) can take their index mod n

Repeat (for each episode):
  Initialize and store $S_0 \neq$ terminal
  Select and store an action $A_0 \sim \pi(\cdot|S_0)$
  $T \leftarrow \infty$
  For $t = 0, 1, 2, \ldots$:
    If $t < T$, then:
      Take action A_t
      Observe and store the next reward as R_{t+1} and the next state as S_{t+1}
      If S_{t+1} is terminal, then:
        $T \leftarrow t+1$
      else:
        Select and store an action $A_{t+1} \sim \pi(\cdot|S_{t+1})$
    $\tau \leftarrow t - n + 1$ (τ is the time whose estimate is being updated)
    If $\tau \geq 0$:
      $G \leftarrow \sum_{i=\tau+1}^{\min(\tau+n,T)} \gamma^{i-\tau-1} R_i$
      If $\tau + n < T$, then $G \leftarrow G + \gamma^n Q(S_{\tau+n}, A_{\tau+n})$ $(G_{\tau:\tau+n})$
      $Q(S_\tau, A_\tau) \leftarrow Q(S_\tau, A_\tau) + \alpha[G - Q(S_\tau, A_\tau)]$
      If π is being learned, then ensure that $\pi(\cdot|S_\tau)$ is ε-greedy wrt Q
  Until $\tau = T - 1$

Exercise: Grid World



Compute 5-step SARSA update
$\alpha = 0.5$
$\gamma = 0.9$

G_1 = 0.9^4 * 1 = 0.6561

$Q(s_{(2,0)}, E)$ = 0 + 0.5*(0.6561 + 0 - 0) = 0.32805

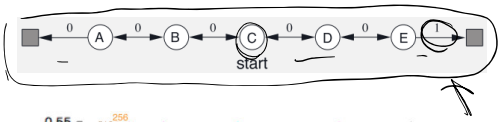
With 1-step learning

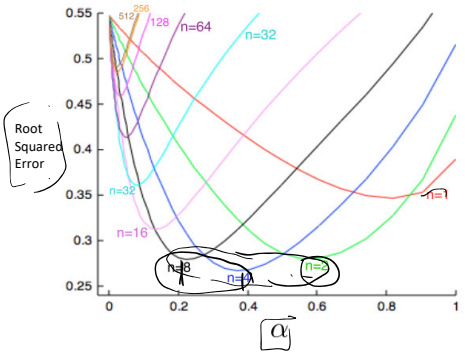
State	Action			
	North	South	East	West
(0,0)	0	0	0	0
(0,1)	0	0	0	0
(0,2)	0	0	0	0
. . .				
(1,2)	0	0	0	0
(2,1)	0	0	0	0
(2,2)	0	0	0.45	0
(2,3)	0	0	0	0
. . .				

With 5-step learning

State	Action			
	North	South	East	West
(0,0)	0	0	0	0
(0,1)	0	0	0	0
(0,2)	0	0	0.2953	0
. . .				
(1,2)	0	0	0.3281	0
(2,1)	0.405	0	0	0
(2,2)	0	0.3645	0.45	0
(2,3)	0	0	0	0
. . .				

Example: Random walk





MCTS + Reinforcement learning

AlphaZero: