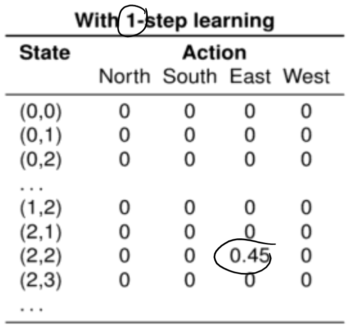
n-step temporal difference learning

Discounted future rewards

$G_t = r_t + V(s')$

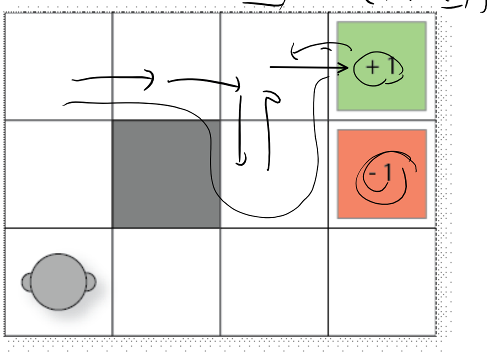
$= r_t + \gamma r_{t+1} + \gamma^2 r_{t+1} \dots + V(S_{t+n})$



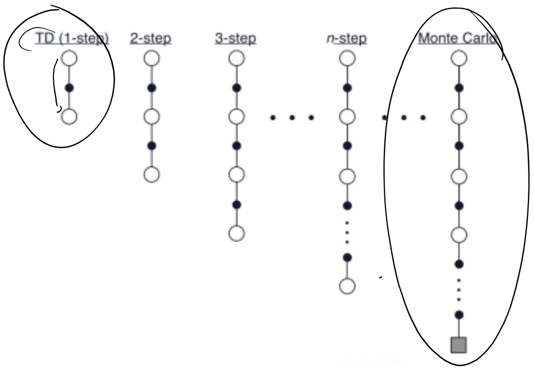
With 1-step learning

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
...				

$Q(s,a) \leftarrow Q(s,a) + \alpha(r + v(s')]$



Truncated future rewards

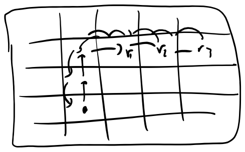


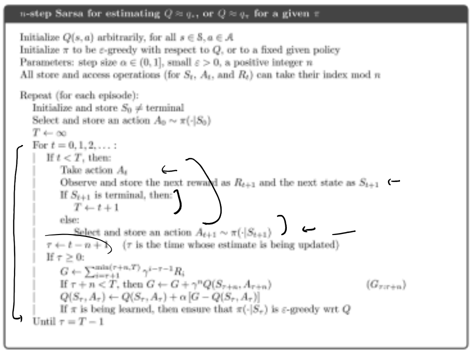
SARSA: Q(s,a) := Q(s,a) + α [r + γQ(s',a') - Q(s,a)]

Change the update:

$G \leftarrow \sum_{i=t+1}^{t+n} r_i$ $\quad n=1$

$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma^n Q(s_{t+1}, a_{t+1}) - Q(s,a)]$



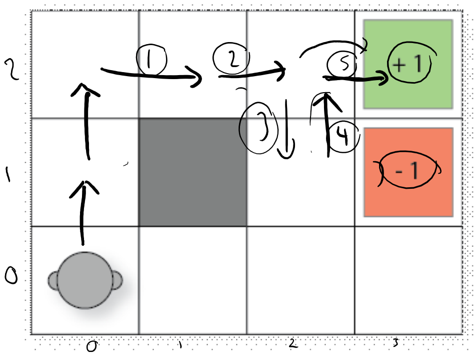


```
n-step Sarsa for estimating $Q \approx q_*$, or $Q \approx q_\pi$ for a given $\pi$

Initialize $Q(s,a)$ arbitrarily, for all $s \in \mathcal{S}, a \in \mathcal{A}$
Initialize $\pi$ to be $\varepsilon$-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 $t = 0, 1, 2, \dots$ :
        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 $A_{t+1} \sim \pi(\cdot|S_{t+1})$
        $\tau \leftarrow t - n + 1$   ($\tau$ 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 $\pi$ is being learned, then ensure that $\pi(\cdot|S_\tau)$ is $\varepsilon$-greedy wrt $Q$
    Until $\tau = T - 1$
```

Exercise: Grid World



Compute 5-step SARSA update

$\alpha = 0.5$

$\gamma = 0.9$

G_5 = $\gamma \cdot 1$

G_4 = $0 + \gamma^2 \cdot 1$

G_3 = $0 + 0 + \gamma^3 \cdot 1$

G_2 = $\vdots$

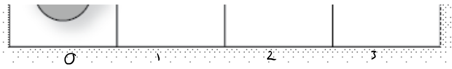
G_1 = $\gamma^5 \cdot 1$

Q(s$_{(2,2)}$,E) = $0 + 0.5\,[\overset{0.9}{\gamma_5} + 0 - 0] = 0.45$

Q(s$_{(1,2)}$,N) = $0 + 0.5\,[0.81 + 0 - 0] = 0.405$

...

Q(s$_{(2,2)}$,S) =



$Q(s_{(2,2)},E) = 0 + 0.5\,[r_5 + 0 - 0] = 0.45$

$Q(s_{(1,2)},N) = 0 + 0.5\,[0.81 + 0 - 0] = 0.405$

...

$Q(s_{(2,2)},S) =$ "

$Q(s_{(2,1)},E) =$ "

$$Q(s,a) = Q(s,a) + \alpha\left[r_t + \overset{0}{\gamma^n Q(s',a')} - \overset{0}{Q(s,a)}\right]$$

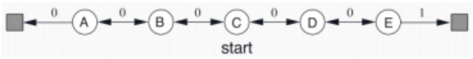
With 1-step learning

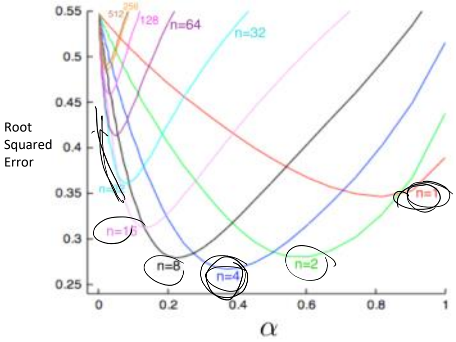
State	North	South	East	West
(0,0)	0	0	0	0
(0,1)	0	0	0	0
(0,2)	0	0	0	0
. . .				
(1,2)	0	0	0	0
(2,1)	0	0	0	0
(2,2)	0	0	0.45	0
(2,3)	0	0	0	0
. . .				

With 5-step learning

State	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
. . .				

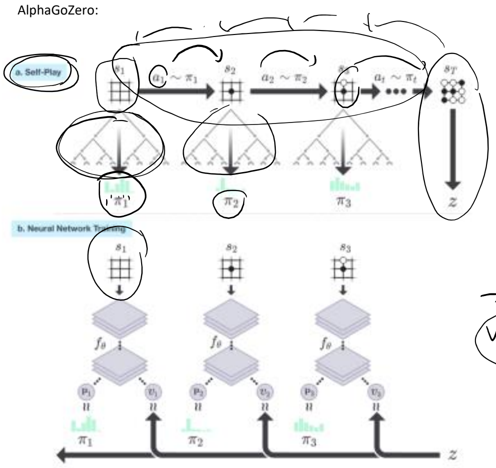
Example: Random walk





MCTS + Reinforcement learning

AlphaGoZero:


a. Self-Play
s_1
$a_1 \sim \pi_1$
s_2
$a_2 \sim \pi_2$
s_3
$a_t \sim \pi_t$
s_T
π_1
π_2
π_3
z
b. Neural Network Training
s_1
s_2
s_3
f_θ
p_1
v_1
p_2
v_2
p_3
v_3
π_1
π_2
π_3
z