



Strojové učenie II

prednáška 5 – Temporal Difference metódy

Ing. Ján Magyar, PhD.

Katedra kybernetiky a umelej inteligencie

Technická univerzita v Košiciach

2021/2022 letný semester

Temporal-difference učenie posilňovaním

- prístup založený na odhade hodnotových funkcií
- model-free – nepotrebujeme úplnú znalosť prostredia, stačí skúsenosť s prostredím
- interakcia s prostredím
 - skutočná
 - simulovaná
- schopný pracovať s
 - neúplnými epizódami
 - kontinuálnymi úlohami
- inkrementálny v zmysle krok po kroku

Rozdiel v princípoch odhadu

$$\begin{aligned} v_{\pi}(s) &= E_{\pi}[G_t | S_t = s] \\ &= E_{\pi}[R_{t+1} + \gamma G_{t+1} | S_t = s] \\ &= E_{\pi}[R_{t+1} + \gamma v_{\pi}(S_{t+1}) | S_t = s] \end{aligned}$$

- MC
 - cieli na **odhad**
 - $V(S_t) \leftarrow V(S_t) + \alpha[G_t - V(S_t)]$
 - odhad na základe skúseností
 - čaká na znalosť G_t (epizóda musí dobehnúť)
- TD
 - cieli na **odhad**
 - $V(S_t) \leftarrow V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$
 - odhad na základe skúsenosti a iného odhadu
 - čaká iba jeden krok

TD chyba

- člen v zátvorke vyjadruje chybu

$$\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$$

- rozdiel medzi aktuálnym odhadom hodnoty pre stav S_t a lepším odhadom pomocou okamžitej skúsenosti a odhadu pre nasledujúci stav S_{t+1}
- ak by sa hodnoty odhadov V nemenili počas epizódy ale iba po nej

$$\begin{aligned} G_t - V(S_t) &= R_{t+1} + \gamma G_{t+1} - V(S_t) + \gamma V(S_{t+1}) - V(S_{t+1}) \\ &= \delta_t + \gamma (G_{t+1} - V(S_{t+1})) \\ &= \delta_t + \gamma \delta_{t+1} + \gamma^2 (G_{t+2} - V(S_{t+2})) \\ &= \sum_{k=t}^{T-1} \gamma^{k-t} \delta_k \end{aligned}$$

Algoritmus TD odhadu v_π

Tabular TD(0) for estimating v_π

Input: the policy π to be evaluated

Algorithm parameter: step size $\alpha \in (0, 1]$

Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(\text{terminal}) = 0$

Loop for each episode:

 Initialize S

 Loop for each step of episode:

$A \leftarrow$ action given by π for S

 Take action A , observe R, S'

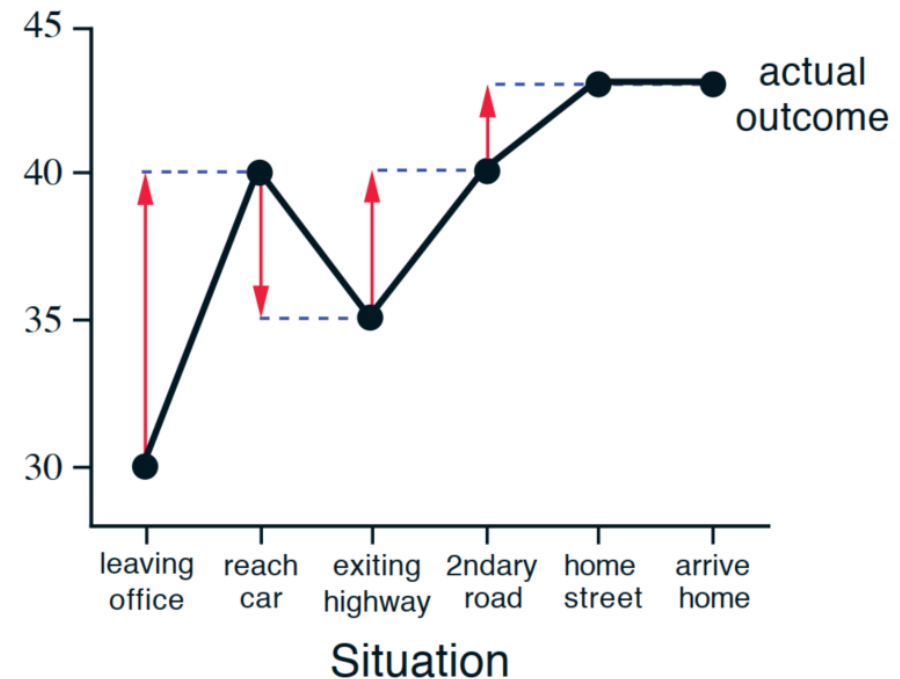
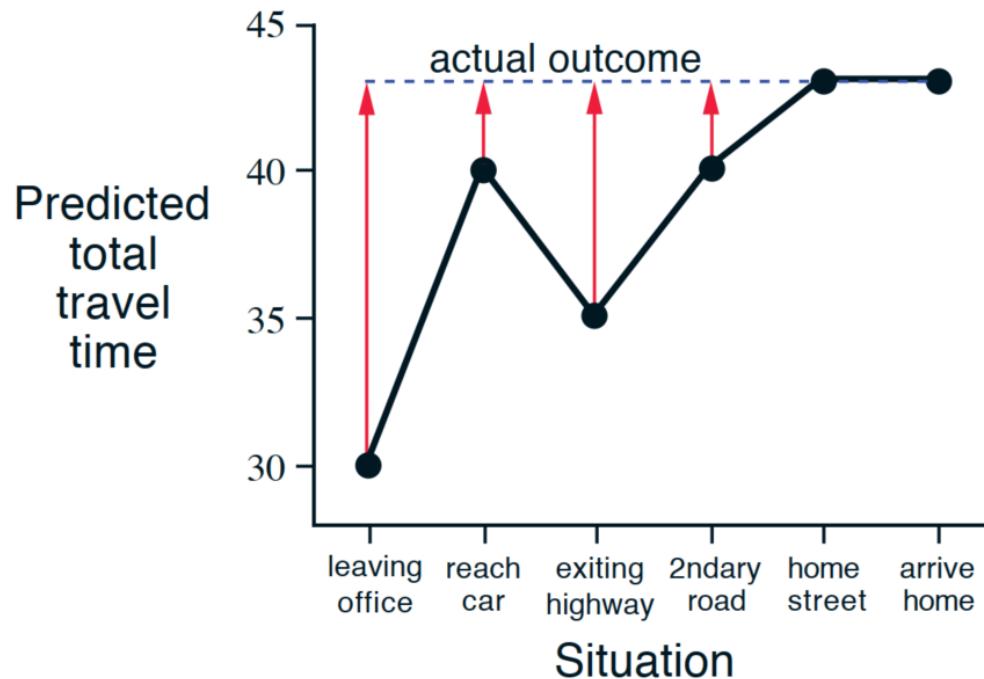
$V(S) \leftarrow V(S) + \alpha[R + \gamma V(S') - V(S)]$

$S \leftarrow S'$

 until S is terminal

Ukážka: cesta domov

- TD – bezprostredne môže updatovať ako reakciu na aktuálnu situáciu
- MC – musí čakať až na príchod domov



Zdroj: Sutton-Barto: Reinforcement Learning, 2nd ed., 2018

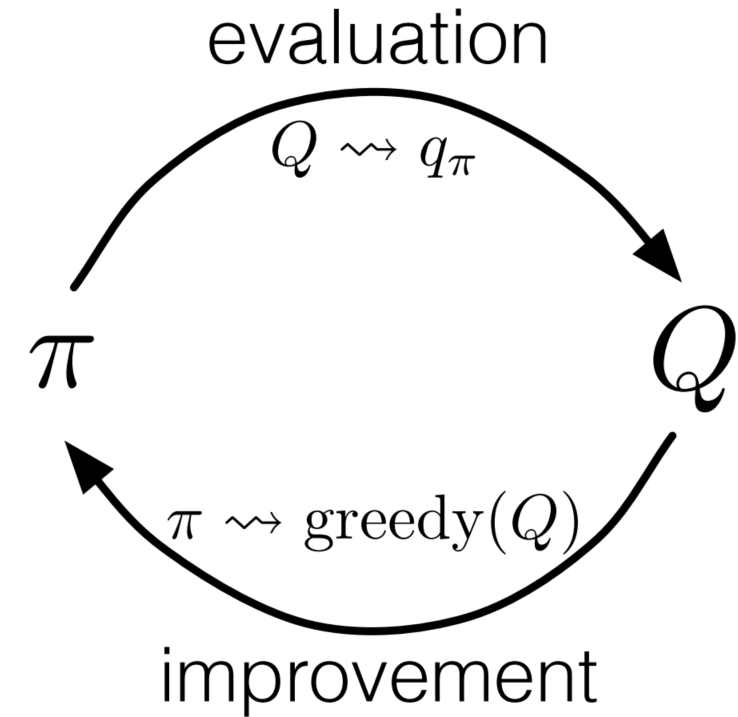
Konvergenca

- V konverguje k v_π , ak
 - α je konštantné a „dostatočne“ malé (konverguje približne)
 - α sa postupne znižuje tak, že platia vzťahy
$$\sum_{n=1}^{\infty} \alpha_n = \infty \text{ a } \sum_{n=1}^{\infty} \alpha_n^2 < \infty$$
(konverguje s pravdepodobnosťou 1)
- čo konverguje rýchlejšie: MC alebo TD?
 - dávkový update: $TD < MC$
 - nedávkový update: $TD ? MC$

Učenie politiky

$\dots, S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}, R_{t+2}, \dots$

- založené na všeobecnej iterácii politiky
- použitie $Q(s, a)$ ako odhadu $q_\pi(s, a)$
- problém explorácie
- on-policy ($\pi = b$) vs off-policy ($\pi \neq b$)



Zdroj: Sutton-Barto: Reinforcement Learning, 2nd ed., 2018

Aktualizácia odhadu Q

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha [R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]$$

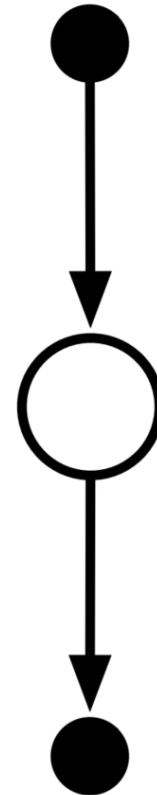
$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right]$$

$$\begin{aligned} Q(S_t, A_t) &\leftarrow Q(S_t, A_t) + \alpha [R_{t+1} + \gamma E[Q(S_{t+1}, A_{t+1}) | S_{t+1}] - Q(S_t, A_t)] \\ &= Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \sum_a \pi(a | S_{t+1}) Q(S_{t+1}, a) - Q(S_t, A_t) \right] \end{aligned}$$

Aktualizácia: $\dots + \gamma Q(S_{t+1}, A_{t+1}) - \dots$

$\dots, R_t, S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}, R_{t+2}, \dots$

- $Q(S_t, A_t)$ je
 - pre neterminálny stav aktualizované podľa päťice vybranej zo sekvencie
 - nulové pre terminálny stav
- on-policy odhad
- konvergencia
- Sarsa algoritmus



Algorithmus Sarsa

Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$

Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+$, $a \in \mathcal{A}(s)$, arbitrarily except that $Q(\text{terminal}, \cdot) = 0$

Loop for each episode:

 Initialize S

 Choose A from S using policy derived from Q (e.g., ε -greedy)

 Loop for each step of episode:

 Take action A , observe R, S'

 Choose A' from S' using policy derived from Q (e.g., ε -greedy)

$Q(S, A) \leftarrow Q(S, A) + \alpha [R + \gamma Q(S', A') - Q(S, A)]$

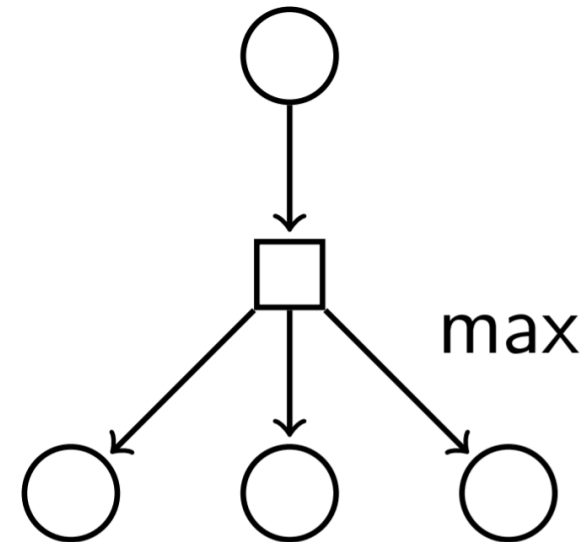
$S \leftarrow S'; A \leftarrow A';$

 until S is terminal

Aktualizácia: $\dots + \gamma \max_a Q(S_{t+1}, A_{t+1}) - \dots$

$\dots, R_t, S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}, R_{t+2}, \dots$

- $Q(S_t, A_t)$ je
 - pre neterminálny stav aktualizované podľa **štvorice** vybranej zo sekvencie
 - nulové pre terminálny stav
- off-policy odhad
- konvergencia
- Q-learning algoritmus



Algorithmus Q-learning

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$

Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(\text{terminal}, \cdot) = 0$

Loop for each episode:

 Initialize S

 Loop for each step of episode:

 Choose A from S using policy derived from Q (e.g., ε -greedy)

 Take action A , observe R, S'

$Q(S, A) \leftarrow Q(S, A) + \alpha [R + \gamma \max_a Q(S', a) - Q(S, A)]$

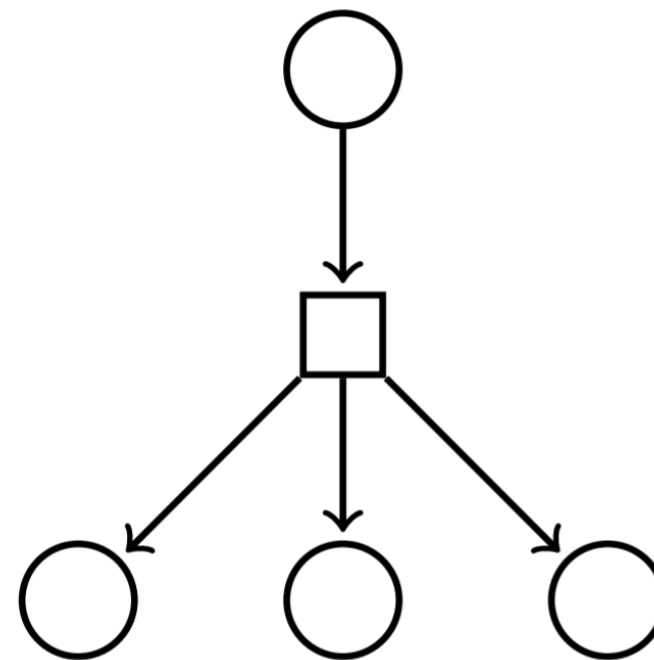
$S \leftarrow S'$

 until S is terminal

Aktualizácia: $\dots + \gamma E[Q(S_{t+1}, A_{t+1}) | S_{t+1}] - \dots$

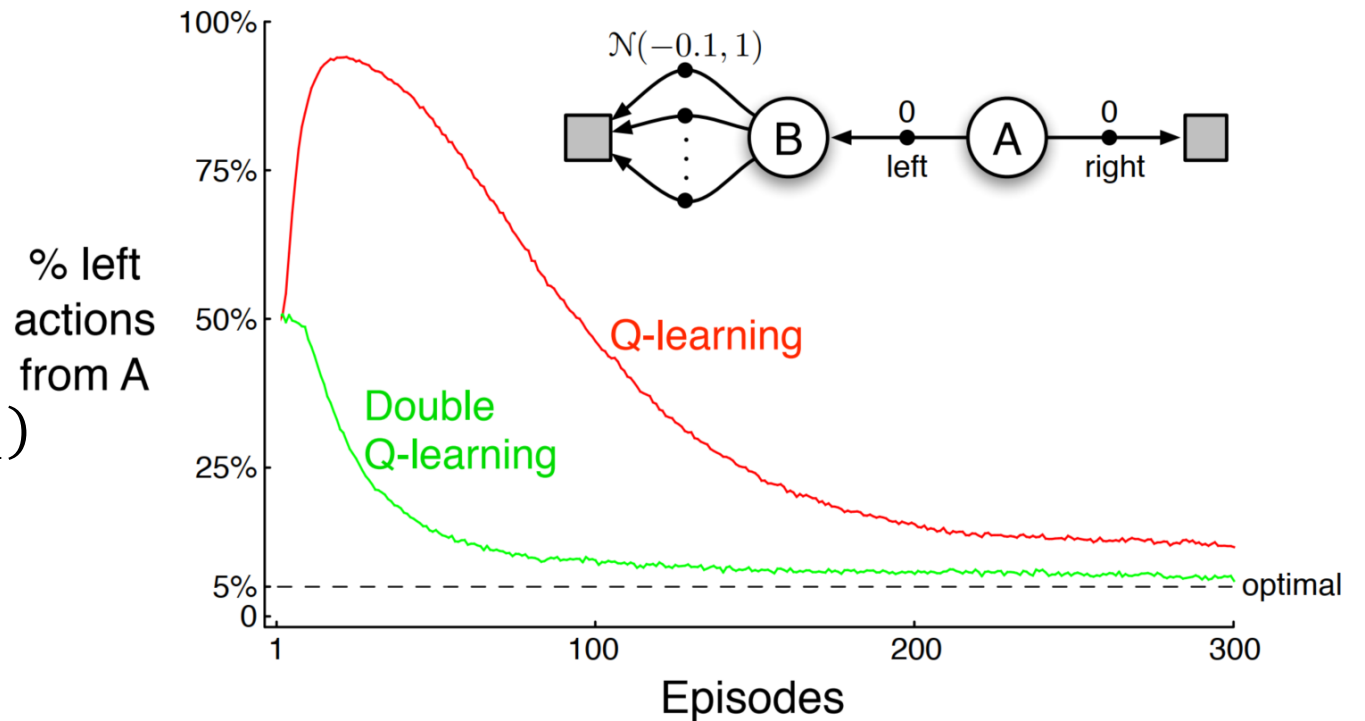
$\dots, R_t, S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}, R_{t+2}, \dots$

- $Q(S_t, A_t)$ je
 - pre neterminálny stav aktualizované podľa **štvorice** vybranej zo sekvencie
 - nulové pre terminálny stav
- spriemerňovanie akcií v S_{t+1} :
$$E[Q(S_{t+1}, A_{t+1}) | S_{t+1}] = \sum_a \pi(a | S_{t+1}) Q(S_{t+1}, a)$$
- on-policy/off-policy odhad
- konvergencia
- Expected Sarsa algoritmus



Dvojité učenie

- maximalizačná odchýlka
 - $E[R \in N(-0.1, 1)] = -0.1$
 - $\max[R \in N(-0.1, 1)] > 0$
- použitie jedného odhadu Q
 - pre určenie najlepšej akcie A_{t+1} v stave S_{t+1}
 - pre odhady hodnoty $q(S_{t+1}, A_{t+1})$
- použitie dvoch nezávislých odhadov Q_1 a Q_2
 - $Q_2 \left(S_{t+1}, \underset{a}{\operatorname{argmax}} Q_1(S_{t+1}, a) \right)$
 - $Q_1 \left(S_{t+1}, \underset{a}{\operatorname{argmax}} Q_2(S_{t+1}, a) \right)$
 - update iba Q_1 alebo Q_2
 - striedanie rolí Q_1 a Q_2



Algoritmus Dvojitý Q-learning

Double Q-learning, for estimating $Q_1 \approx Q_2 \approx q_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$

Initialize $Q_1(s, a)$ and $Q_2(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, such that $Q(\text{terminal}, \cdot) = 0$

Loop for each episode:

 Initialize S

 Loop for each step of episode:

 Choose A from S using the policy ε -greedy in $Q_1 + Q_2$

 Take action A , observe R, S'

 With 0.5 probability:

$$Q_1(S, A) \leftarrow Q_1(S, A) + \alpha \left(R + \gamma Q_2(S', \arg\max_a Q_1(S', a)) - Q_1(S, A) \right)$$

 else:

$$Q_2(S, A) \leftarrow Q_2(S, A) + \alpha \left(R + \gamma Q_1(S', \arg\max_a Q_2(S', a)) - Q_2(S, A) \right)$$

$S \leftarrow S'$

until S is terminal

Dvojitá Sarsa

$$\pi(a|s) = \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{|A(s)|} & a = \operatorname{argmax}_a (Q_1(s, a) + Q_2(s, a)) \\ \frac{\varepsilon}{|A(s)|} & \text{inak} \end{cases}$$

$$Q_1(S_t, A_t) \leftarrow Q_1(S_t, A_t) + \alpha [R_{t+1} + \gamma Q_2(S_{t+1}, A_{t+1}) - Q_1(S_t, A_t)]$$
$$Q_2(S_t, A_t) \leftarrow Q_2(S_t, A_t) + \alpha [R_{t+1} + \gamma Q_1(S_{t+1}, A_{t+1}) - Q_2(S_t, A_t)]$$

Dvojitá Expected Sarsa

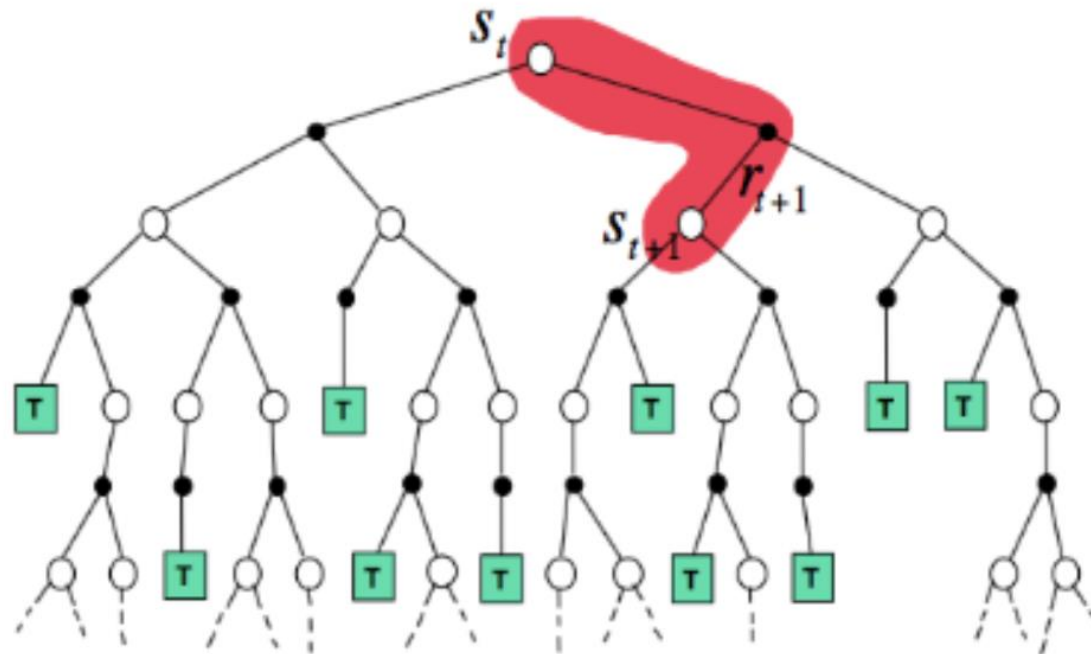
$$\pi(a|s) = \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{|A(s)|} & a = \operatorname{argmax}_a (Q_1(s, a) + Q_2(s, a)) \\ \frac{\varepsilon}{|A(s)|} & \text{inak} \end{cases}$$

$$Q_1(S_t, A_t) \leftarrow Q_1(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q_2(S_{t+1}, a) - Q_1(S_t, A_t) \right]$$
$$Q_2(S_t, A_t) \leftarrow Q_2(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q_1(S_{t+1}, a) - Q_2(S_t, A_t) \right]$$

Backup diagram

Temporal-Difference

$$V(S_t) \leftarrow V(S_t) + \alpha (R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$$



Zdroj: <https://lilianweng.github.io/lil-log/2018/02/19/a-long-peek-into-reinforcement-learning.html>