Tuesday, June 7, 2022 9:00 AM

(s, a, r, s', a')

$$Q^{\pi}_{t+1}(s,a) = Q^{\pi}_{t}(s,a) + \alpha_t \left(r(s,a) + \gamma\, Q^{\pi}_{t}(s', a') - Q^{\pi}_{t}(s,a) \right)$$

TD-learning SARSA → learn the fixed policy

$$Q^{\pi^*}_{t+1}(s,a) = Q^{\pi^*}_{t}(s,a) + \alpha_t \left(r(s,a) + \gamma\, Q^{\pi^*}_{t}(s', a^*) - Q^{\pi^*}_{t}(s,a) \right)$$

↳ $a^* = \underset{a}{\operatorname{argmax}}\, Q_t(s', a)$

Q-learning → learn optimal policy

$$Q_{t+1}(s,a) = Q_t(s,a) + \alpha_t \left(r(s,a) + \gamma \sum_{a^+} \pi(a^+|s')\, Q(s', a^+) - Q_t(s,a) \right)$$

expectation Q-learning.

¿ How to construct an alternate to exploring "at random"?

- "reward" exploring new states
- distribution should depend on current

$$Q(s,a) \longrightarrow \frac{e^{\eta Q(s,a)}}{\sum_{a'} e^{\eta Q(s,a')}} =: \pi(a|s)$$

s	a = 1	a = 2
1	-0.1	0.1
2	0.3	0.2

$$\rightarrow \frac{e^{-0.1}}{e^{-0.1} + e^{0.1}} \qquad \frac{e^{0.1}}{e^{-0.1} + e^{0.1}} \qquad \eta = 1$$

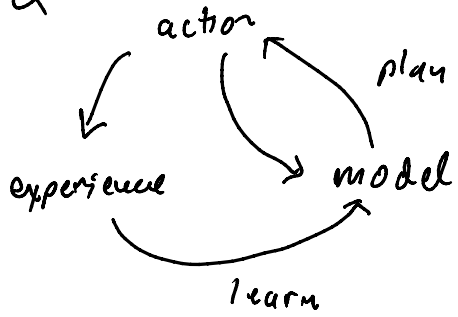
$$\rightarrow \frac{e^{0.3}}{\quad} \qquad \frac{e^{0.2}}{\quad}$$

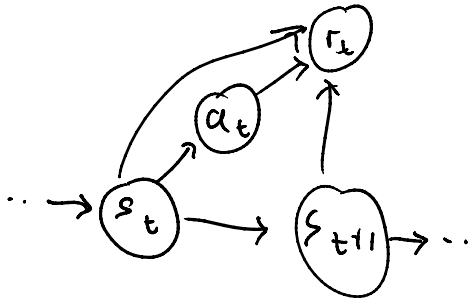
2 0.3 0.2 → $\frac{e^{0.3}}{e^{0.3}+e^{0.2}}$ $\frac{e^{0.2}}{e^{0.3}+e^{0.2}}$

Experience Replay Buffer & Dyna-Q

$n = 1{,}000$

$$\begin{bmatrix} s_0, a_0, r_0, s_1, a_1 \\ s_1, a_1, r_1, s_2, a_2 \\ \vdots \\ s_n, a_n, r_n, s_n, a_n \end{bmatrix}$$





$$\mathbb{P}(s_{t+1}=s', r_t=r \mid s_t=s, a_t=a) = B_{s,a}(s', r)$$

MLE

Q-learning has the danger of overestimating the value of certain actions from certain states.

→ double Q-learning

$$Q^{(1)}_{t+1}(s_t, a_t) = Q^{(1)}_t(s_t, a_t) + \alpha_t\left[r(s_t, a_t) + \gamma Q^{(2)}_t(s_{t+1}, a^*) - Q^{(1)}_t(s_t, a_t)\right]$$

↳ $a^* = \underset{a'}{\operatorname{argmax}}\, Q^{(1)}_t(s_{t+1}, a')$

similar for $Q^{(2)}$ (1) ↔ (2)

randomly choose to update $Q^{(1)}$ or $Q^{(2)}$

Wednesday, June 8, 2022 11:19 AM

model-~~free~~ agnostic RL

- no model of the world
- learn value / policy from experience (s, a, r, s')

model-based RL

- learn model from experience
- plan / optimise using sims from model

Dyna RL

- learn model from experience
- learn & plan from simulations & experience
- treat experience & model sims as equal.

Linear models

$$V^\pi(s) \sim V_\theta^\pi = \theta' g$$

fixed

$$g = \begin{pmatrix} g_1(s) \\ g_2(s) \\ \vdots \\ g_d(s) \end{pmatrix}$$

learn these parameters

$$V^\pi(s) \sim V_\theta(s)$$

minimise Loss associated with L^2 error of the approximation

$$\ell(\theta) := \frac{1}{2} \mathbb{E}^{\mathbb{P}_\pi}\left[(V^\pi(s) - V_\theta(s))^2\right]$$

$$\nabla_\theta \ell(\theta) = \mathbb{E}^{\mathbb{P}_\pi}\left[(V^\pi(s) - V_\theta(s))\, \nabla_\theta V_\theta(s)\right]$$

?

$$V^\pi(s) := \mathbb{E}^{\mathbb{P}_\pi}\left[r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots \mid s = s\right]$$

$$= \mathbb{E}^{\mathbb{P}_\pi}\left[r_1 + \gamma V^\pi(s') \mid s = s\right]$$

$V_\theta(s')$

$\theta \leftarrow \theta - \eta \nabla_\theta \ell(\theta)$ → gradient descent

$\sim \left(r + \gamma V_\theta(s') - V_\theta(s)\right) \nabla_\theta V_\theta(s)$

$$\theta_{t+1} = \theta_t - \eta_t \left(r_t + \gamma V_\theta(s_{t+1}) - V_\theta(s_t) \right) \nabla_\theta V_\theta(s_t)$$

$\eta_1 > \eta_2 > \cdots > 0 \qquad \lim_{t\to\infty} \eta_t = 0 \qquad \sum \eta_t = +\infty \qquad \sum \eta_t^2 < +\infty$

$\eta_t = \frac{c}{D+t}$

$$\ell(\theta) = \frac{1}{2}\mathbb{E}\left[\left(Q(s,a) - Q_\theta(s,a) \right)^2 \right]$$

$$\nabla_\theta \ell(\theta) = \mathbb{E}\left[\left(Q(s,a) - Q_\theta(s,a) \right) \nabla_\theta Q_\theta(s,a) \right]$$

$$\theta_{t+1} = \theta_t - \eta_t \left[\left(\left(r + \gamma Q_\theta(s', a') \right) - Q_\theta(s,a) \right) \nabla_\theta Q_\theta(s,a) \right]$$

Function approximation associated with SARSA

$$\theta_{t+1} = \theta_t - \eta_t \left[\left(\left(r + \gamma Q_\theta(s', a^*) \right) - Q_\theta(s,a) \right) \nabla_\theta Q_\theta(s,a) \right]$$

↳ $\arg\max_{a'} Q_\theta(s', a')$

FA for Q-learning

Double Q-learning with FA

$Q_\theta \qquad Q_{\tilde{\theta}}$

↳ $\arg\max_{a' \in A(s')} Q_\theta(s', a')$

$\theta - \theta - \eta \left[\left(r + \gamma Q_{\tilde{\theta}}(s', a^*) \right) - Q_\theta(s,a) \right] \nabla_\theta Q_\theta(s,a)$

$$\theta_{t+1} = \theta_t - \eta_t \left[\left(r + \gamma Q_{\tilde{\theta}}(s', a^*) \right) - Q_\theta(s,a) \right) \nabla_\theta Q_\theta(s,a) \right]$$

$a^* \in \arg\max_{a' \in A(s')} Q_\theta(s', a')$

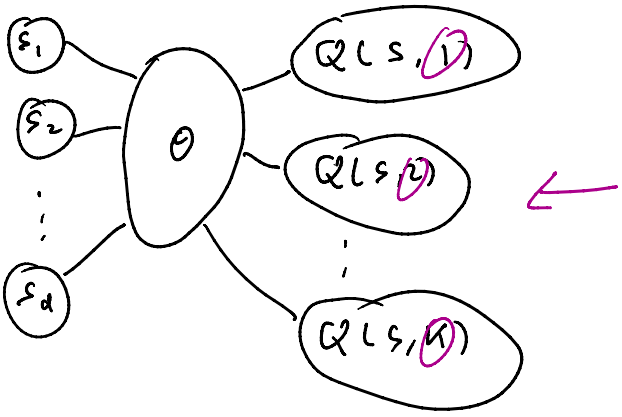
$$\tilde{\theta}_{t+1} = \tilde{\theta}_t - \eta_t \left[\left(r + \gamma Q_{\theta}(s', a^*) \right) - Q_{\tilde{\theta}}(s,a) \right) \nabla_{\tilde{\theta}} Q_{\tilde{\theta}}(s,a) \right]$$

$a^* \in \arg\max_{a' \in A(s')} Q_{\tilde{\theta}}(s', a')$

---

neural-net approximations

Deep Q-learning



define a modified loss

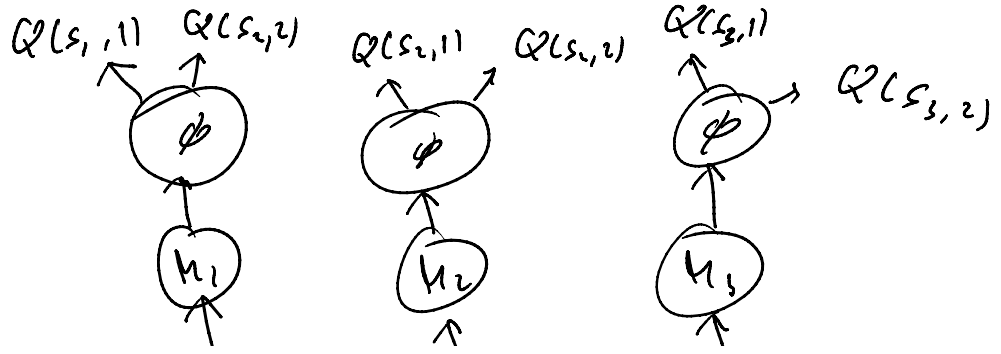
ε-greedy!

$$\mathcal{L}(\theta) := \mathbb{E}\left[\left(r + \gamma Q_{\tilde{\theta}}(s', a^*) - Q_\theta(s,a) \right)^2 \right]$$

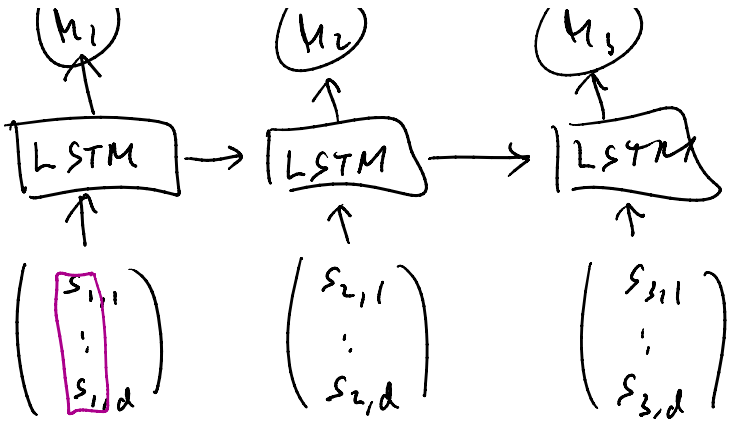
$a^* \in \arg\max_{a' \in A(s')} Q_\theta(s', a')$

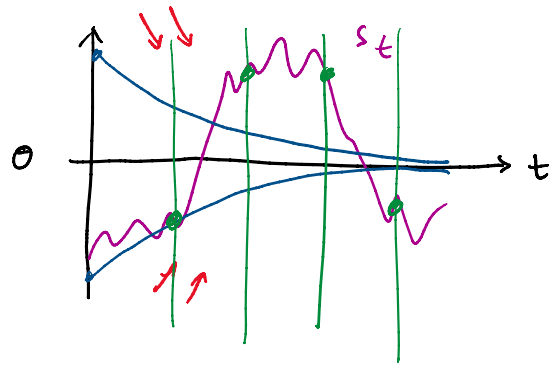
$$\approx \frac{1}{N} \sum_{n=1}^{N} \left(r^{(n)} + \gamma Q_{\tilde{\theta}}(s'^{(n)}, a^{*(n)}) - Q_\theta(s^{(n)}, a^{(n)}) \right)^2$$

$\nabla_\theta \mathcal{L}(\theta)$ → use backprop to compute



$\theta = \{\phi, w_{1:K}\}$


h_1
h_2
h_3
LSTM
LSTM
LSTM
$s_{1,1}$
$s_{1,d}$
$s_{2,1}$
$s_{2,d}$
$s_{3,1}$
$s_{3,d}$



Ornstein-Uhlenbeck (OU)

$S = (S_t)_{t \geq 0}$

$dS_t = \kappa(\theta - S_t)\,dt + \sigma\,dW_t$

$W = (W_t)_{t \geq 0}$ a Brownian motion

$\kappa, \theta, \sigma > 0$

$\dot{S}_t = \kappa(\theta - S_t)$

$S_t = \theta + (S_0 - \theta)\,e^{-\kappa t}$

$$S_t = \theta + (S_0 - \theta)\,e^{-\kappa t} + \sigma \int_0^t e^{-\kappa(t-u)}\,dW_u$$

$$S_{t+\Delta t} \,|\, \mathcal{F}_t = \underbrace{\left(\theta + (S_t - \theta)\,e^{-\kappa \Delta t}\right)}_{} + \sigma \int_t^{t+\Delta t} e^{-\kappa(t+\Delta t-u)}\,dW_u$$

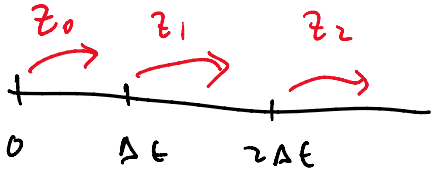
$\sim \mathcal{N}\left(0;\ \int_t^{t+\Delta t} e^{-2\kappa(t+\Delta t-u)}\,du\right)$

$= \dfrac{1 - e^{-2\kappa\Delta t}}{2\kappa}$

$:= \tilde{\sigma}$

$$= \underbrace{\theta(1-e^{-\kappa\Delta t})}_{\alpha} + \underbrace{e^{-\kappa\Delta t}}_{\beta} S_t + \underbrace{\sigma\sqrt{\frac{1-e^{-2\kappa\Delta t}}{2\kappa}}}_{\tilde{\sigma}} Z_t$$

$Z_t \sim \mathcal{N}(0,1)$ iid



AR(1)

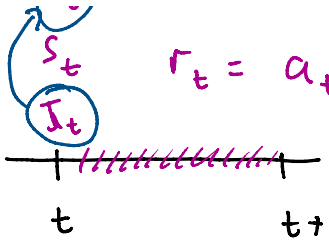
state $= (S_t, I_t)$ $S_t \in \mathbb{R}$

if finite horizon, include t as a dimension of state variable

$I_t \in \{+1, 0, -1\} =: \mathcal{A}$

$a_t \in \mathcal{A}$

$r_t = a_t(S_{t+\Delta t} - S_t) - \lambda\,|I_t - a_t|\,S_t$



$$r_t = a_t(S_{t+\Delta t} - S_t) - \lambda |I_t - a_t| S_t$$

$$S_{t+\Delta t} = \alpha + \beta S_t + \tilde{\sigma} z_t$$

$$I_{t+\Delta t} = a_t$$