Lecture - 6: Value-based RL

Last time:

① Value-iteration:

$$\pi(s) = \underset{a}{\operatorname{argmax}}\, Q(s,a)$$

$|A|$

$|S|$

$$Q(s,a) \leftarrow r(s,a) + \gamma V(s')$$

$$V(s) \leftarrow \max_{a'} Q(s,a')$$

② Fitted Q-iteration: (neural FQI)

for $i \geq 0, \ldots$:

$$\theta_{i+1} \leftarrow \underset{\theta}{\operatorname{argmin}} \; \mathbb{E}_{s,a,s'}\left[\left(Q_\theta(s,a) - y\right)^2\right]$$

y: Q_i used here

$$y(s,a) = r(s,a) + \gamma \max_{a'} Q(s',a')$$

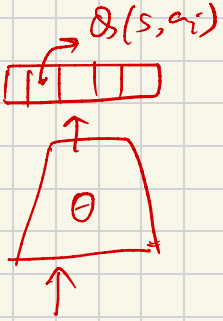
$Q(s',a')$: θ_i

③ DQN:

↳ 1) replay buffer →  history of (s,a,s')

2) target networks

Outline of DQN algorithm:



Exploration:

$\pi_{\theta_i}(s)$ $\quad \underset{a}{\text{argmax}}\ Q_{\theta_i}(s,a)$

"eps-greedy" $\rightarrow$ prob. ε

$a \sim \text{unif}(\mathcal{A})$

prob $(1-\varepsilon)$, π_{θ_i}

Update:

replay buffer

(s, a, r, s') $\Rightarrow$ $\min_\theta$ $\mathbb{E}_{s,a,r,s' \sim \mathcal{D}}$ $\left[\left(Q_\theta - (r + \gamma \max_{a'} Q_{\theta^{tgt}}(s', a')) \right)^2 \right]$

only K steps of grad. descent

θ_{i+1} $\quad$ main $\quad$ target network

Target networks

1. $\theta_0' = \theta_0^{tgt}$ (target values)
2. $\theta_0 \rightarrow$ grad. descent
3. update target net: $\theta_k^{tgt} = \theta_K$ $\quad$ "θ_i"

(tons) exploring $\rightarrow$ updates (few)

1. UTD ratio: $\frac{K}{|\text{data}|}$ $\rightarrow$ updates-to-data

   can't be too large

2. Hard target updates:

   vs soft

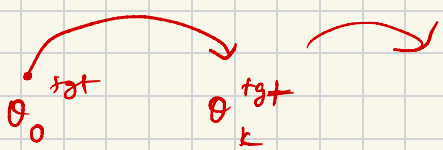
$\alpha = 0.005$

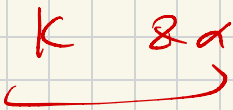
Soft target updates: $\theta^{tgt}_{i+1} \leftarrow (1-\alpha)\,\theta^{tgt}_{i} + \alpha\,\theta_i$ ("soft")

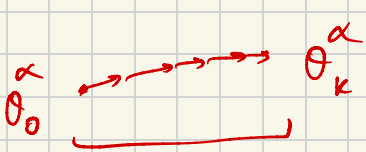
$$\theta' \leftarrow (1-\alpha)\,\theta + \alpha\,\theta_{new}$$

vanilla $\theta^{tgt}_{k} = \theta_k, \quad \theta^{tgt}_{2k} = \theta_{2k} \ldots$

Why prefer one over the other?







Challenges with DQN:

"Double Q-learning"

① Overestimation in Q-values

error accumulation

$$y = r(s,a) + \gamma \max_{a'} Q_{\theta}^{tgt}(s',a')$$

not trained on all s'

not trained on all a'

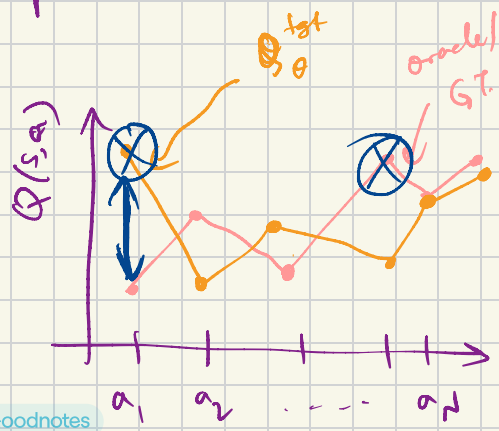
Compounding errors

what is this term supposed to be?

optimal value from s' ideal $V^*(s')$ but

Q is $Q_\theta \neq Q^*$

$\max_{a'} Q_\theta(s',a')$ ⇝ erroneous & we amplify the +ve errors



Q_0^{tgt} → erroneous

$\downarrow\downarrow$

y_0 → erroneous

$\downarrow\downarrow$

θ_1 → erroneous

$\vdots$

θ_k^{tgt} → erroneous

Double DQN

How can we fix this problem?

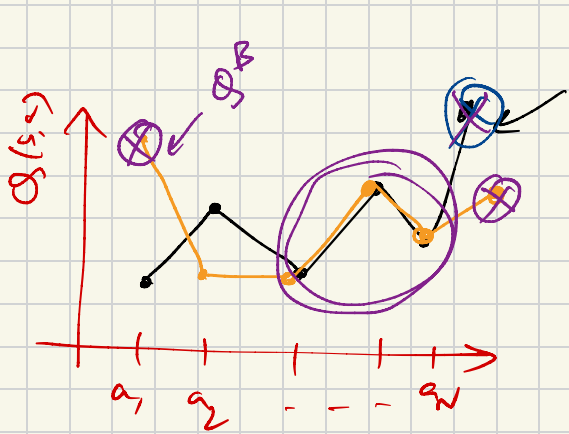
Idea: use argmax action from one Q-function & compute Q-values under the other

$(Q_\theta^A - y^A)^2$ $\quad$ $(Q_\theta^B - y^B)^2$

Q_θ^A $\quad$ Q_θ^B

initialized differently

$Q_\theta^{tgt,A}$, $\quad$ $Q_\theta^{tgt,B}$



$\max_a Q^{tgt}(s,a)$

1) $\text{argmax}_a \; Q^{tgt,A}(s,a) := a^*$

2) $Q^{tgt,B}(s,a^*)$

$$y = r(s,a) + \gamma \max_{a'} Q^{tgt}(s',a')$$

$Q^{tgt,B}(s',a'^*_A)$

$$y^A = r(s,a) + \gamma Q^{tgt,A}(s', a'^*_B)$$

Another way to fix this problem.

Idea: reduce error compounding by using N-step returns.

$$\left(Q_\theta(s,a) - y(s,a)\right)^2$$

$$\mathbb{E}_{s,a,s',r\sim\mathcal{D}}[\cdots]$$

$$y(s,a) = r(s,a) + \gamma \max_{a'} Q^{tgt}(s',a')$$

$\neq \arg\max_{a'}$

a' seen in the replay buffer

$$r(s,a) + \gamma\, r(s',a') + \gamma^2 \max_{a''} Q^{tgt}(s'',a'')$$

$Q(s,a)$

"N"

next state

replay buffer

$$r(s_i,a_i) + \gamma\, r(s_{i+1},a_{i+1}) + \gamma^2 r(s_{i+2},a_{i+2}) + \cdots + \gamma^N \max_{a_{N+1}} Q^{tgt}(s_{N+1},a_{N+1})$$

Practical details for Q-learning

1. Coverage really helps.
2. High UTD can destabilize training!
3. Improved loss functions:

Huber loss:

$$L(x) = \begin{cases} \frac{x^2}{2}, & |x| < \delta \\ \delta|x| - \delta^2/2, & |x| \geq \delta \end{cases}$$

Cross-entropy loss:

Advanced content:

Residual gradient vs TD-learning

$$\min_\theta \mathbb{E}\left[\left\| Q_\theta(s,a) - (r + \gamma Q_\theta(s',a')) \right\|_2^2\right]$$

$$\Rightarrow \min_\theta \mathbb{E}_{s,a,s'}\left[\left(Q_\theta(s,a) - \gamma Q_\theta(s',a') - r(s,a)\right)^2\right]$$

What's the difference?

At convergence, in theory:

In practice: