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 $i \geq 0, \ldots$:

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

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

Q_i used here

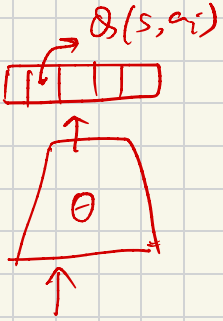
Q_i

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

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: $\quad$ updates-to-data $\quad \frac{K}{|data|}$ $\rightarrow$ can't be too large

② Hard target updates:

vs soft

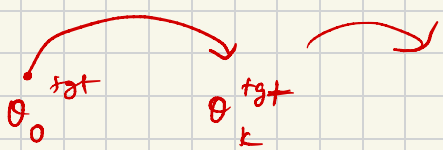
$\alpha = 0.005$

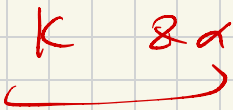
Soft target updates: $\theta^{tgt}_{i+1} \leftarrow (1-\alpha)\,\theta^{tgt}_{i} + \alpha\,\theta_i$ ("soft")

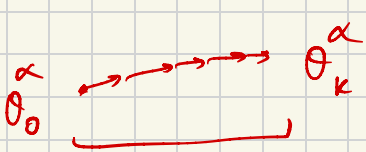
$$\theta' \leftarrow (1-\alpha)\,\theta + \alpha\,\theta_{new}$$

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'

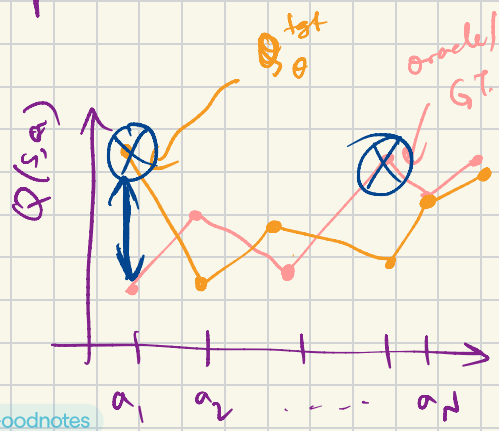
Compounding errors

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?

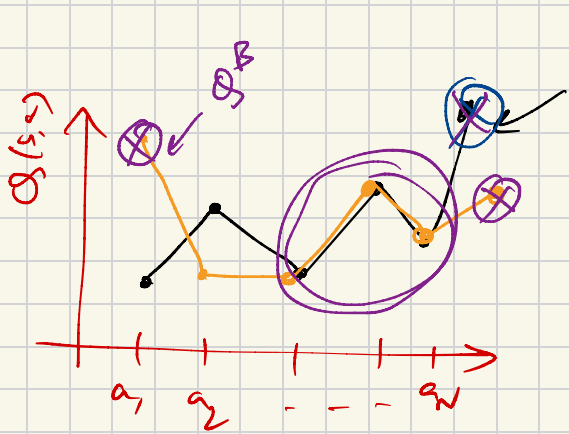
Idea: use argmax action from one Q-function & compute Q-values under the other

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

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

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

$Q(s,a)$

$r(s',a')$: next state

"N"

replay buffer

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

Practical details for Q-learning

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

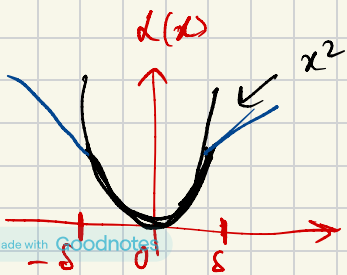
$x \in \mathbb{R} \Rightarrow$  $\Rightarrow$ Cross Entropy

Huber loss:

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

$x = (Q_\theta(s,a) - y(s,a))$

$|x|^2$



$\delta |x|$

$\frac{d|x|}{dx}$

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:

Lecture 7: Advanced PG

So far, we saw:

- Policy gradient methods
  (policy, maybe an on-policy $V(s)$)
  on-policy
- Q-learning methods
  (no policy, just a value fn)
  off-policy

Can we build a hybrid of these methods?

Should be possible!

Why?

Off-policy, actor-critic

off-policy

replay buffers

$\text{argmax}\ Q(s,a)$ ✗

$\pi_\theta(a|s)$ ↓

How can we get started in a principled manner??

$$\mathbb{E}_{a\sim\mu}\left[A^{\mu}(s,a)\right] = 0 \qquad A^{\mu}(s,\underline{a}) \hookrightarrow \text{arg action}$$

Performance Difference Lemma

$$A^{\mu}(s,a) = Q^{\mu}(s,a) - V^{\mu}(s)$$

For any two policies $\pi(a|s)$ & $\mu(a|s)$, the following holds:

$$J(\pi) - J(\mu) = \mathbb{E}_{\tau\sim p_{\pi}(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t A^{\mu}(s_t,a_t)\right]$$

$J(\pi)$: expected sum of discounted rewards under π (return of policy π); $J(\mu)$: policy μ

$$J(\pi) = \mathbb{E}_{\tau\sim p_{\pi}(\tau)}\left[\sum_t \gamma^t r(s_t,a_t)\right]$$

Proof:

$$\mathbb{E}_x[f(x)] = \mathbb{E}_{x,y,z,\dots}[f(x)]$$

$$J(\pi) = \mathbb{E}_{\tau\sim p_{\pi}(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] = \mathbb{E}_{s_0\sim p(s_0)}\left[V^{\pi}(s_0)\right]$$

$\tau = s_0 \sim p(s_0),\ a_0\sim\pi,\ s_1\sim p(s_{t+1}|s_t,a_t),\ a_1\sim\pi,\ \dots$ (independent of whatever is in τ beyond (s_0))

$$= \mathbb{E}_{\tau\sim p_{\pi}(\tau)}\left[V^{\pi}(s_0)\right] = \mathbb{E}_{\tau\sim p_{\mu}(\tau)}\left[V^{\pi}(s_0)\right]$$

$$= \mathbb{E}_{\tau\sim p_{\mu}(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t V^{\pi}(s_t) - \sum_{t=1}^{\infty}\gamma^t V^{\pi}(s_t)\right] \qquad \left(\sum_{t=0}^{\infty-1=\infty}\gamma^{t+1}V^{\pi}(s_{t+1})\right)$$

$$= \mathbb{E}_{\tau\sim p_{\mu}(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t\left(V^{\pi}(s_t) - \gamma V^{\pi}(s_{t+1})\right)\right]$$

$$V^{\pi}(s_0) = V^{\pi}(s_0) + \gamma V^{\pi}(s_1) - \gamma V^{\pi}(s_1) + \gamma^2 V^{\pi}(s_2) - \gamma^2 V^{\pi}(s_2)$$

$$A^{\pi}(s_t, a_t) = r(s_t, a_t) + \gamma V^{\pi}(s_{t+1}) - V^{\pi}(s_t)$$

$$J(\pi) = \mathbb{E}_{\tau \sim p_{\pi}(\tau)} \left[\sum_{t=0}^{\infty} \gamma^t \left(V^{\pi}(s_t) - \gamma V^{\pi}(s_{t+1}) - r(s_t, a_t) \right) \right] + \mathbb{E}_{\tau \sim p_{\pi}(\tau)} \left[\sum \gamma^t r(s_t, a_t) \right]$$

$$J(\pi) = -\mathbb{E}_{\tau \sim p_{\pi}(\tau)} \left[\sum_{t=0}^{\infty} \gamma^t A^{\pi}(s_t, a_t) \right] + J(\mu)$$

$$\Rightarrow \boxed{J(\mu) - J(\pi) = \mathbb{E}_{\tau \sim p_{\pi}(\tau)} \left[\sum_{t=0}^{\infty} \gamma^t A^{\pi}(s_t, a_t) \right]}$$

$$\max_{\pi} J(\pi) = \max_{\pi} \left(J(\pi) - J(\mu) \right)$$

$$= \max_{\pi} \mathbb{E}_{s \sim d^{\pi}} \left[A^{\mu}(s, a) \right]$$

Why is this useful?

A practical off-policy policy gradient algorithm

$$J(\pi) - J(\mu) = \mathbb{E}_{\tau \sim P_\pi(\tau)} \left[\sum_{t=0}^{\infty} \gamma^t A^\mu(s_t, a_t) \right]$$

$J(\mu)$: data collection policy

$A^\mu(s_t, a_t)$: estimated via a critic

Option 1: use states from μ, A^π instead

Option 2: use states from π, A^μ instead

Which one should we use?

An approximation

$$\max_{\pi} \; J(\pi) - J(\mu)$$

$$\mathbb{E}_{\tau \sim p_\pi(\tau)} \left[\sum_{t=0}^{\infty} \gamma^t A^\mu(s_t, a_t) \right]$$

$$= \sum_{t=0}^{\infty} \mathbb{E}_{s_t \sim p_\pi(s_t)} \left[\gamma^t A^\mu(s_t, a_t) \right]$$

$a_t \sim \pi(\cdot|s_t)$

ignore

utilize

$$\mathbb{E}_{s_t \sim p_\mu(s_t)} \left[\mathbb{E}_{a_t \sim \pi(\cdot|s_t)} \left[\gamma^t A^\mu(s_t, a_t) \right] \right]$$

only term involved in maximization

Different ways to solve this maximization when π & μ are close!!

Generic optimization problem:

$$\max_{\pi} \quad \mathbb{E}_{s \sim d^{\mu}(s)} \left[\mathbb{E}_{a \sim \pi(\cdot|s)} \left[A^{\mu}(s,a) \right] \right] - \alpha D(\pi, \mu)$$

what is this, precisely?

We want $P_{\pi}(s_t)$ & $P_{\mu}(s_t)$ to be close!

$P_{\pi}(s_t) - P_{\mu}(s_t) = ??$

say π & μ are deterministic & agree with probability $1-\varepsilon$, then:

$$P_{\pi}(s_t) = (1-\varepsilon)^t \, P_{\mu}(s_t) + \left(1 - (1-\varepsilon)^t\right) P_{diff}(s_t)$$

$$\Rightarrow \left| P_{\pi}(s_t) - P_{\mu}(s_t) \right| = \left(1 - (1-\varepsilon)^t\right) \left| P_{\mu}(s_t) - P_{diff}(s_t) \right|$$

$$\leq 2 \cdot \varepsilon \cdot t$$

This means that:

$$D(\pi, \mu) = D_{TV}(\pi(\cdot|s), \mu(\cdot|s))$$

total variations distance.

$$D_{TV}(P, Q) \leq \sqrt{\frac{1}{2} D_{KL}(P \| Q)}$$

&

$$D_{TV}(P, Q) \leq \sqrt{\frac{1}{2} D_{KL}(Q \| P)}$$

} Pinsker's inequality

Each of these choices gives a different algorithm:

① AWR → $D_{KL}(\pi \| \mu)$

② PPO / TRPO → $D_{KL}(\mu \| \pi)$

Advantage-weighted regression (AWR)

[Peng, Kumar, Li, Levine, 2019]

$$\max_{\pi} \; \mathbb{E}_{s \sim d^{\mu}(s)} \left[\mathbb{E}_{a \sim \pi(\cdot|s)} \left[A^{\mu}(s,a) \right] - \alpha D_{KL}(\pi \| \mu) \right]$$

Easier problem:

$$\max_{p(x)} \; \mathbb{E}_{x \sim p(x)} \left[f(x) \right] - \alpha D_{KL}(p \| q)$$

$$\Rightarrow \quad p^*(x) \propto q(x) \exp\left(\frac{f(x)}{\alpha} \right)$$

So, $\pi^*(a|s) \propto \mu(a|s) \exp\left[\frac{A^{\mu}(s,a)}{\alpha} \right]$

How do I use this oracle solution??

PPO (proximal policy optimization)

$$\max_\theta \; \mathbb{E}_{\substack{s\sim d^\mu(s) \\ a\sim \pi_\theta(\cdot|s)}} \left[A^\mu(s,a) \right] - \alpha D_{KL}\left(\mu(\cdot|s) \,\|\, \pi_\theta(\cdot|s) \right)$$

another way to compute this is via IS

$$\mathbb{E}_{x\sim p(x)}\left[f(x)\right] = \mathbb{E}_{x\sim q(x)}\left[\frac{p(x)}{q(x)} f(x) \right]$$

different dist.

How can we do this for policy grad?

$$\mathbb{E}_{\substack{s\sim d^\mu(s) \\ a\sim \mu(\cdot|s)}} \left[\frac{\pi_\theta(a|s)}{\mu(a|s)} A^\mu(s,a) \right] - \alpha D_{KL}\left(\mu(\cdot|s) \,\|\, \pi_\theta(\cdot|s) \right)$$

Do I need the KL term?

Answer: No!!

- Can I just throw out the KL?

Not really as the update will be too large!

But $D_{KL}(\mu \| \pi)$ effectively keeps π & μ close.

So, we can keep $\frac{\pi_\theta(a|s)}{\mu(a|s)}$ close to 1.

$$L^{CLIP}(\theta) = \mathbb{E}_{\substack{s \sim d^{\mu}(s) \\ a \sim \mu(\cdot|s)}} \left[\text{clip}\left(\frac{\pi_\theta(a|s)}{\mu(a|s)}, 1-\varepsilon, 1+\varepsilon \right) A^{\mu} \right]$$

Any problems with the clip objective?

(1) Can make an update even when we get clipped on the lower side!

$$L^{CLIP}(\theta) = \mathbb{E}\left[\min\left(r(\theta)\cdot A^{\mu},\ \text{clip}(r(\theta), 1-\varepsilon, 1+\varepsilon)\cdot A^{\mu}\right)\right]$$

when $r(\theta) < 1-\varepsilon$ & $A^{\mu}(s,a) > 0$

you see some signal.

but when $r(\theta) < 1-\varepsilon$ & $A^{\mu}(s,a) < 0$

you don't see signal.

Is this good or bad?

Likewise if

$r(\theta) > 1+\varepsilon$ but $A^{\mu}(s,a) < 0$

you see some signal

but when $r(\theta) > 1+\varepsilon$ but $A^{\mu}(s,a) > 0$

there is no signal.

This hurts exploration!!

sometimes useful to have ε_{high}, ε_{low}

"asymmetric clip"

used in DAPO