SOFT ACTOR-CRITIC

→ OFF-POLICY, reuse past experience (not directly feasible with standard policy gradient) → ON-POLICY requires new samples ∀ gradient step
→ MAXIMUM ENTROPY
→ STOCHASTIC ACTOR

→ augments the STANDARD maximum reward RL objective with an ENTROPY MAXIMIZATION TERM
  → better exploration and robustness to estimation error
  → SAME PROB of acting for example two actions that have the same (maximum) entropy

3 ingredients
1) ACTOR-CRITIC architecture with separate policy and value fcn
2) OFF-POLICY to reuse previously collected data
3) ENTROPY MAXIMIZATION to enable stability and exploration

1) Start from POLICY ITERATION → POLICY EVALUATION, → POLICY IMPROVEMENT using the value fcn

Policy → ACTOR
Value fcn → CRITIC

3) Maximum entropy RL optimizes policies to maximize both the expected return and the expected entropy of the policy

→ continuous state and action spaces

$MDP = (S, A, p, r) \rightarrow S \times A \rightarrow [r_{min}, r_{max}]$

↳ $S \times S \times A \rightarrow [0, \infty)$ = prob of $s_{t+1} \in S$ given $s_t \in S$ and $a_t \in A$

$\rho_\pi(s_t)$ = state
$\rho_\pi(s_t, a_t)$ = state-action
} marginals of trajectory distrib. induced by $\pi(a_t|s_t)$

→ TEMPERATURE param determines the importance of EM term

$$* J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t, a_t) \sim \rho_\pi} \underbrace{[r(s_t, a_t)}_{\text{STANDARD reward}} + \alpha H(\pi(\cdot|s_t))] \quad (1)$$

pro - wider exploration → gives up on clearly unpromising avenues
- can capture multiple modes of near-optimal behavior
↳ equal prob. to those actions

ENTROPY Measure of chaos in terms of "how random a RV is"

↳ $= * H(P) = \mathbb{E}_{x \sim P}[-\log P(x)]$, x RV with distribution P

SOFT POLICY ITERATION

1) SOFT POLICY EVALUATION, compute value of π according to ME objective in (1)

For a fixed π, the SOFT Q-value can be computed iteratively from any $Q: S \times A \rightarrow \mathbb{R}$ and a modified Bellman operator T^π

$$* T^\pi Q(s_t, a_t) \triangleq r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim \rho}[V(s_{t+1})],$$

where $V(s_t) = \mathbb{E}_{a_t \sim P}[Q(s_t, a_t) \underbrace{- \log \pi(a_t|s_t)}_{\text{ENTROPY!}}]$

$V(s_t)$ = SOFT STATE-VALUE FN

LEMMA Q^k (with $Q^{k+1} = T^\pi Q^k$) will converge to the soft Q-value of π as $k \to \infty$

2) SOFT POLICY IMPROVEMENT, update policy towards the exp of the new Q-fn

* $\pi_{new} = \underset{\pi' \in \Pi}{\arg\min}\ D_{KL}\left(\pi'(\cdot|s_t) \,\middle\|\, \frac{\exp\left(Q^{\pi_{old}}(s_t, \cdot)\right)}{Z^{\pi_{old}}(s_t)} \right)$ (2)

we restrict Π to a Gaussian family

↳ convenient to use KULLBACK-LEIBLER DIVERGENCE

↳ PARTITION FN that normalizes the distrib. Can be ignored because it is intractable and does not affect π_{new}

LEMMA $Q^{\pi_{new}}(s_t, a_t) \geq Q^{\pi_{old}}(s_t, a_t)$, $\forall (s_t, a_t) \in S \times A$ and $\pi_{old}, \pi_{new} \in \Pi$

But we need to approximate it to represent the Q-values (too expensive and work only in the tabular case)

⇒ **SOFT ACTOR-CRITIC** to deal with action/state spaces that are continuous

We use a parametrized $V_\psi(s_t)$, $Q_\theta(s_t, a_t)$ and $\pi_\phi(a_t|s_t)$ where ψ, θ and ϕ are the params of the network

↳ value fns modeled as NNs, policy as Gaussian with μ and σ given by NNs

→ Soft value fn trained to minimize the SQUARED RESIDUAL ERROR

↳ * $J_V(\psi) = \mathbb{E}_{s_t \sim D}\left[\frac{1}{2}\left(V_\psi(s_t) - \mathbb{E}_{a_t \sim \pi_\phi}\left[Q_\theta(s_t, a_t) - \log \pi_\phi(a_t|s_t) \right] \right)^2 \right]$

where D is the distrib. of previously sampled states and actions

↳ REPLAY BUFFER

→ Soft Q-fn params can be trained to minimize the soft Bellman residual

↳ * $J_Q(\theta) = \mathbb{E}_{(s_t, a_t) \sim D}\left[\frac{1}{2}\left(Q_\theta(s_t, a_t) - \hat{Q}(s_t, a_t) \right)^2 \right]$

as usual ← $r(s_t, a_t) + \gamma\, \mathbb{E}_{s_{t+1} \sim p}\left[V_{\bar{\psi}}(s_{t+1}) \right]$

N.B. Both $J_V(\psi)$ and $J_Q(\theta)$ can be optimized with stochastic gradients

→ Finally, policy params learned by minimizing the expected D_{KL} in (2)

But no way to back-propagate with the sampling

⇒ REPARAMETRIZATION TRICK using a NN transformation, so samples are obtained according to

* $\tilde{a}_\theta(s, \xi) = \tanh\left(\mu_\theta(s) + \sigma_\theta(s) \odot \xi \right)$, $\xi \sim N(0, I)$ (SPHERICAL GAUSSIAN)

Now, gradients flow through μ_θ and σ_θ (it's the only way to allow it)

Two Q-fns to mitigate positive bias in policy improvement step that degrades performance of value-based method

↳ these Q-fns are parametrized with θ_i and train independently to optimize $J_Q(\theta_i)$

↳ then use the minimum of the Q-fns for the value gradient