from last time:

REINFORCE algo

↳ subtracting a baseline

$$L(\theta) = \frac{-1}{N} \sum_{i=1}^{N} A_i \log \pi_\theta(y_i|x)$$

↳ advantage

$$A_i = r(x, y_i) - b$$

↳ GRPO: special case where baseline is computed via G rollouts from same prompt x

$$A_i = \frac{r(x, y_i) - \text{mean}(G)}{\text{std}(G)}$$

→ any policy gradient method

↳ one issue w/ REINFORCE is that sampling is slow, as G increases, and the rollout length increases

↳ w/ REINFORCE, we only perform one gradient update for a given rollout

↳ can we take several gradient steps per rollout/group w/o policy drift affecting the update

↳ PPO: Schulman et al. 2017

↳ idea: take multiple gradient steps on the same rollout, but make the steps small

---

PPO: we have a policy that generated the rollouts: π_{OLD}

↳ after a gradient step, current policy $\pi_\theta \neq \pi_{OLD}$

↳ policy ratio

$$\rho_i = \frac{\pi_\theta(y_i|x)}{\pi_{OLD}(y_i|x)}$$

how much more/less likely is y_i under the current policy compared to the one that generated it

$$L(\theta) = \frac{-1}{N}\sum_{i=1}^{N} \rho_i(\theta) A_i$$

$$E_{y\sim\pi_{OLD}(\cdot|x)}\left[\rho A(x,y)\right] = \sum_y \pi_{OLD}(y|x)\cdot\frac{\pi_\theta(y|x)}{\pi_{OLD}(y|x)} A(x,y)$$

$$= \sum_y \pi_\theta(y|x) A(x,y)$$

↳ this loss estimates $-(J(\theta)-b)$ at the new θ, but w/ old rollouts

↳ intuition: before any gradient steps, $\pi_{OLD} = \pi_\theta$, $\rho = 1$, $\frac{d\rho}{d\theta} = \frac{d\log\pi_\theta}{d\theta}$

↳ algorithm reduces to REINFORCE

↳ we don't want ρ to drift too far away from 1

↳ don't want ρ to move outside some small value $\pm\epsilon$

↓

± 0.2

↳ implement this by clipping

$$L(\theta) = \frac{-1}{N}\sum_{i=1}^{N} \min\Big(\rho_i A(x,y),\ \text{clip}(\rho_i, 1-\epsilon, 1+\epsilon)\, A(x,y)\Big)$$

clip: not differentiable

$1-\epsilon, 1+\epsilon$: trust region

advantage	goal
$A > 0$	increase ρ_i until $\rho_i = 1+\epsilon$
$A < 0$	decrease ρ_i until $\rho_i = 1-\epsilon$

↳ intuition: we stop updating the model after the probability has moved ϵ in the direction that the advantage dictates

↳ if ρ moved in the wrong direction the min function keeps the unclipped term so the next update can fix it

↳ with these constraints, we can safely take many gradient updates on the same batch of rollouts

↳ in practice, policy ratio computed per token, at position t

$$\rho_t = \frac{\pi_\theta(y_t | x, y_{<t})}{\pi_{old}(y_t | x, y_{<t})}$$

each token clipped separately

baseline in PPO is actually a trained model

↳ value network / function

---

in GRPO: baseline is just group statistics

in PPO: we only do one rollout per prompt

↳ so we learn the baseline

$$V(x, y_{<t})$$

↳ train value fn by regression

$$L(\theta_v) = \frac{1}{T} \sum_{t=1}^{T} \left(V(x, y_{<t}) - r(x, y) \right)^2$$

↳ once I have V_ϕ

$$A_t = r(x, y) - V(x, y_{<t})$$

↳ per token baseline

↳ downside: value network has to be a big model of its own