From last class (PPO):

b = batch size
L = seq^{max} length

1. given a batch of prompts $x_1, x_2, \dots x_b$ we generate corresponding responses $y_1, y_2, \dots y_b$ from policy model

2. for each example (x_i, y_i), we compute reward $r(x_i, y_i)$

3. compute advantages by subtracting baseline / KL penalty

$$A(x,y) = r(x,y) - b$$

$r(x,y)$: $b \times 1$

b: varies
$b \times 1$ if using batch mean
$b \times L$ using value fn

$$A^{KL}(x,y) = A(x,y) - \beta D_{KL}(P_\theta \| P_{SFT})$$

$A(x,y)$: $b \times L$

$D_{KL}(P_\theta \| P_{SFT})$: $b \times L$

4. Compute policy ratios, take gradient steps

$$\rho_i = \frac{\rho_\theta(y_i \mid y_{<i}, x)}{\rho_{\theta_{old}}(y_i \mid y_{<i}, x)} \Rightarrow 1 \text{ before the first gradient steps}$$

$\rho_{\theta_{old}}$ → frozen

ρ of dimensionality $b \times L$

$$J(\theta) = \sum_{i=1}^{L} \min\left(\rho_i A^{\pi_\theta}(x,y),\ \text{clip}(\rho_i, 1-\varepsilon, 1+\varepsilon) A^{\pi_\theta}(x,y)\right)$$

↳ $+A \Rightarrow$ increase ρ_i

$-A \Rightarrow$ decrease ρ_i

policy ratio

↳ before any grad steps, $\rho_i = 1$

bc $\rho_{\theta_{old}} = \rho_\theta$

↳ we normally take multiple steps on one batch (↩ 1-4)

↳ why? can't we just take one step w/ a big LR

↳ Stability, if we take one big step, we'll move too far and many tokens will fall into clipped region

↳ rollouts (i.e. generating y from P_θ) are very expensive, compared to just doing multiple rounds of backprop

---

PPO → GRPO

↳ remove reward model, replace w/ verifiable reward

↳ RL gets cheaper

↳ reduce reward hacking

↳ can't reliably compute reward for many tasks

↳ remove value fn, replace w/ group mean

↳ KL not included in advantage computation, but included as separate penalty

↳ often completely dropped

1. given a prompt x, generate a group of G (4-32) different responses: $y_1, y_2, \dots y_G$

2. compute reward for each response in the group $r(x, y_i)$

3. compute group mean and std. dev

$$\mu_x = \frac{1}{G} \sum_{i=1}^{G} r(x, y_i) \qquad \sigma_x$$

4. compute group relative advantages

$$A^{GR}(x, y_i) = \frac{r(x, y_i) - \mu_x}{\sigma_x}$$

$b \times 1$

$\hookrightarrow b \times 1$

5. compute loss, do gradient updates

$$J(\theta) = \frac{1}{G}\sum_{i=1}^{G}\left[\sum_{t=1}^{L} \min\left(\rho_t A_t^{GR}(x, y_{i,t}),\ \text{clip}(\rho_t, 1-\varepsilon, 1+\varepsilon) A_t^{GR}(x, y_{i,t})\right)\right]$$

$$- \beta D_{KL}(P_\theta \| P_{SFT})$$

can be dropped

---

okay but I still have $r(x, y_i)$

↳ verifiable reward

↳ binary (0 or 1)

↳ correctness, proper formatting

↳ graded by some simple rules

x: compute 219 + 408

$r(x,y)$: if $y = 627 \Rightarrow r(x,y) = 1$

else $r(x,y) = 0$

] correctness reward

↳ thinking tokens ∋ reasoning

↳ special subsequence of y used for reasoning

x: let $2x+5=17$. solve for x.] problem

thinking request format [put your thinking steps between <think></think> tokens. box your final answer <answer></answer>

y: <think> well okay so $2x+5=17$, that means $2x=12$, so $x=6$. </think>
<answer> 6 </answer>

↳ reward computed only over the final answer

↳ thinking tokens still part of backprop

↳ separate format reward