RL studies

phil saad

July 2025

1 Introduction

...

2 Entropy in RL

Let's consider the entropy for the response of a policy π given a prompt p

$$\mathcal{H} = -\mathbb{E}_{t\sim\pi(\cdot|p)}[\log\pi(t|p)]$$

Here t is a sequence of tokens (let's fix the length L)

We want to study the change in the entropy over a gradient step, in which the parameters θ_α change as

$$\theta_\alpha \to \theta_\alpha + \delta\theta_\alpha = \theta_\alpha + \eta\partial_\alpha J$$

with

$$J = \mathbb{E}_{t\sim\pi}[A(t)]$$

where A is the advantage (we imagine probably using GRPO type advantage).

Note to self: J isn't $\mathbb{E}[SA]$, when we take the gradient we get $\partial_\alpha J = \partial_\alpha \sum_t \pi(t)A(t) = \sum_t \pi(t)\frac{\partial_\alpha\pi(t)}{\pi(t)}A(t) = \mathbb{E}_{t\sim\pi}[\partial_\alpha S(t)A(t)]$.

First let's just stick with linear order in $\delta\theta_\alpha$ or η

$$\delta\mathcal{H} \approx \sum_\alpha \partial_\alpha\mathcal{H}\delta\theta_\alpha$$

Let's address these two terms in order

2.1 Linear order

Let's look at the linear order change in the entropy for a fixed prompt, for simplicity - we just average over prompts in the end

$$\begin{aligned}
\delta\mathcal{H} &\approx \sum_\alpha \partial_\alpha \mathcal{H} \delta\theta_\alpha && (1)\\
&= -\sum_\alpha \delta\theta_\alpha \times \partial_\alpha \sum_{\mathbf{t}} \pi_\theta(\mathbf{t}|p) \log \pi_\theta(\mathbf{t}|p) && (2)\\
&= -\sum_\alpha \delta\theta_\alpha \times \sum_{\mathbf{t}} \partial_\alpha \pi_\theta \log \pi_\theta && (3)\\
&= -\sum_\alpha \delta\theta_\alpha \times \mathbb{E}_{\mathbf{t}\sim\pi(\cdot|p)}[\log \pi \ \partial_\alpha \log \pi] && (4)\\
& && (5)
\end{aligned}$$

where in the third line we used $\sum_{\mathbf{t}} \partial_\alpha \pi = 0$

The gradient of the objective J is familiar

$$\partial_\alpha J = \partial_\alpha \sum_t \pi(t) A(t) = \sum_t \pi(t) \frac{\partial_\alpha \pi(t)}{\pi(t)} A(t) = \mathbb{E}_{t\sim\pi}[\partial_\alpha \log \pi(t) A(t)]$$

So altogether

$$\delta H = -\eta \sum_\alpha \mathbb{E}[\log \pi \partial_\alpha \log \pi] \times \mathbb{E}[\partial_\alpha \log \pi \ A] + \mathcal{O}(\eta^2)$$

This is only true on expectation though - in practice we use a noisy estimate of the expectation value of the policy gradient, so that the linear order variation is

$$\delta H_1 \to \eta \sum_\alpha \mathbb{E}_t[g^{\mathcal{H}}_\alpha(t)] \times \frac{1}{G} \sum_{i=1}^{G} g^J_\alpha(t_i)$$

where $g^{\mathcal{H}}_\alpha(t) = -\log \pi(t) \partial_\alpha \log \pi(t)$ and $g^J_\alpha = A(t) \partial_\alpha \log \pi(t)$. On expectation this is the same as the earlier line, but that won't be true to second order since we will have two noisy gradients.

Okay let's explore this formula a bit more. We estimate these expectation values via sampling sequences of response tokens. Let $G(p)$ denote the set of sampled responses for a given prompt. For simplicity let's imagine there is just one prompt for now (trivial to include more). Let's also refer to $\log \pi_\theta(t|p)$ as $S_\theta(t|p)$, dropping any arguments and subscripts when they are clear from context. Later on we might also want to refer to individual tokens, let's call them $\tau_a \in t$.

It will be useful to subtract a baseline from $g^{\mathcal{H}}_\alpha(t)$:

$$-\mathbb{E}_t[g^{\mathcal{H}}_\alpha(t)] = \mathbb{E}_t[S(t) \partial_\alpha S(t)] = \mathbb{E}_t[(S(t) - \mathbb{E}[S]) \partial_\alpha S(t)]$$

we can do this because $\mathbb{E}[\partial_\alpha S] = 0$. The advantage is already baseline subtracted so we don't need to do this for g^J.

Now it's also useful to rewrite our equation for $\delta\mathcal{H}_1$ as

$$\delta\mathcal{H}_1 = \frac{-\eta}{G}\sum_{i=1}^{G}\mathbb{E}_t\Big[(S(t)-\overline{S})\;K_1(t,t'_i)\;A(t'_i)\Big]$$

where $\overline{S} = \mathbb{E}_t[S(t)]$ and the (first order in learning rate) kernel K_1 is

$$K_1(t,t') = \sum_\alpha \partial_\alpha S(t)\partial_\alpha S(t')$$

I think this might be called the "fisher kernel". Clearly it is related to the Fisher information - If $w_{\alpha t} = \partial_\alpha S(t)$, then $K_1(t,t') = w^T w$ and $F_{\alpha\beta} = ww^T$.

The fisher kernel has a pretty simple interpretation. Consider the first order step change in the logprob for sequence t. Using the previous derivations we see

$$\delta\log\pi(t) = \eta\frac{1}{B}\sum_{i=1}^{B}K(t,t'_i)A(t'_i)$$

So it is a measure of the effect a sequence t'_i used in the update has on the logprob of getting sequence t (the advantage just tells us the strength and direction of that effect)

- We've assumed that the objective function is the on-policy objective, with the normalization $1/G$ per prompt. In my current implementation of my RL trainer, I am using the dr. grpo normalization, where I also normalize by the maximum response length per prompt.
- We should also average over prompts

Together, these simply modify

$$\delta\mathcal{H}_1 \rightarrow -\eta\frac{1}{BG}\sum_{p\in B}\frac{1}{L_{max}(p)}\sum_{i=1}^{G}\mathbb{E}_{t\sim\pi(\cdot|p)}\Big[(S(t)-\overline{S})\;K_1(t,t'_i)\;A(t'_i)\Big]$$

In this form we can see a more clear relationship with the result of Cui et al. Let's imagine K was diagonal, then

$$\mathbb{E}[\delta\mathcal{H}_1] \rightarrow \mathrm{Cov}(S(t),A(t))$$

The difference with Cui et al's result is this is per-sequence, not per-token.

But K is likely not diagonal (this is something we should study experimentally!). Let's try and make some predictions for the behavior of K_1. The matrix elements $w_{\alpha t}$ hopefully scale like $1/\sqrt{N_{params}}$ since apparently initializing the weights to be of order $1/\sqrt{N_{params}}$ is typical, and it is apparently standard practice to have training keep the weights of order this size, for stability. Then the diagonal elements of K_1 would be of order one (in terms of the scaling with the number of model parameters). So this is a "nice" quantity to study.

Perhaps as a very crude model, we might model $w_{\alpha t}$ as a Wishart random matrix.

However, a totally random matrix is probably a poor approximation. In reality, this matrix probably has a very blocky structure. Let's incorporate the fact that we would have different prompts as well as many responses per prompt, so that we should really think of the sequence index as like $(t|p)$, where p ranges over some fixed number D of prompts and t ranges over all V^L possible response sequences. The matrix elements of w will probably be correlated if the prompts are the same, and even more so if the responses t share subsequences. So e.g. for two sequences $(t_1|p), (t_2|p)$ that differ by one token, the matrix elements of w will be probably highly correlated.

If w was a true Wishart random matrix, we would have some interesting crossover at $V^L \sim N_{params}$, where K would go from nearly the identity for $V^L < N_{params}$ to very much not. But perhaps we can still get some intuition from this. Let's first note that the sequence length L does not have to be very big before K will necessarily have some zero eigenvalues. With $V \sim 10^6$ and $N_{params} \sim 10^9 - 10^{11}$ (I wonder how LoRA affects all of this??) then L need only be order one. But that doesn't necessarily mean that these zero eigenvalues are relevant. Most sequences are junk, so most of the matrix is irrelevant. (In other words, the typical sequences dominate the expectation values, so matrix elements for atypical sequences can basically be ignored - if we projected onto the typical subspace that would be fine)

2.1.1 LET'S WRITE DOWN SOME GOALS

- Estimate $\delta\mathcal{H}_1$ during training, compare to real step change in entropy. Compute matrix elements of $K_1(t, t')$, look at diagonal, between sequences in same prompt, and sequences from different prompts to get a sense of how big contributions from different pairs of sequences are. Use some matrix sketching technques as well. We should take into account conditioning factors from adam vs sgd $\delta\theta_\alpha \to \eta P_\alpha \partial_\alpha J$, though maybe it doesn't make much of a difference in the structure of fisher kernel. In the end we want to understand 1) if linear order is a good enough estimation of the entropy change per step and 2) beyond the sequences identified in cui et al (sequences with negative advantage and a token with outlier very negative logprob), are there other types of sequences which may contribute to entropy collapse? In situations with long sequences, off-diagonal sequences may become more important, try and study this in theory, though unlikely we can do experiments...

2.1.2 Estimating $g_\alpha^{\mathcal{H}}$

So in practice we want to measure $\delta\mathcal{H}_1$ and compare it to the true step change in entropy. In order to do that we need to estimate $\mathbb{E}_t[g_\alpha^{\mathcal{H}}]$ (or after rearranging, the $\mathbb{E}_t$ in $\frac{1}{G}\sum_i \mathbb{E}_t[(S - \overline{S})K_1 A]$. If we use the same set of samples

2.2 Comparison to the gradient noise scale

I think it would be good to compare with the math behind the gradient noise scale for the loss from the paper "An Empirical Model of large batch training". Let's first go through a derivation of the relevant formulas so we can see if any of their techniques are useful.

We have the "True" loss

$$L_{true} = \mathbb{E}_{t \sim D}[-\log \pi_\theta(t)]$$

where t are sequences of tokens drawn from a "true" distribution D.

To first order in the learning rate we have (using ordinary SGD for now)

$$\begin{aligned}
\delta L_1 &= \sum_\alpha \partial_\alpha L_{true} \delta\theta_\alpha && (6)\\
&= -\underbrace{\mathbb{E}_{t\sim D}[\partial_\alpha \log \pi(t)]}_{-G_\alpha} \underbrace{\frac{\eta}{B}\sum_{i=1}^{B} \partial_\alpha \log \pi(t'_i)}_{-\eta G^{est}_\alpha} && (7)\\
&= -\frac{\eta}{B}\mathbb{E}_{t\sim D}\Big[\sum_\alpha \partial_\alpha \log \pi(t) \partial_\alpha \log \pi(t'_i)\Big] && (8)\\
&= -\frac{\eta}{B}\sum_{i=1}^{B} \mathbb{E}_{t\sim D}\Big[K_1(t, t'_i)\Big] && (9)
\end{aligned}$$

Okay so it involves exactly this kernel we ecountered, the Fisher Kernel. If we generalize from ordinary SGD in the perfectly conditioned case, we still get the Fisher Kernel, but the appropriately conditioned one.

This illustrates again the interpretation of the Fisher kernel as a meausure of how much a seqeunce t' used in an update influences the log probability of getting sequence t

Let's go to second order. I'll drop the explicit sums over parameters since its always simple einstein convention

$$\begin{aligned}
\delta L_2 &= \frac{1}{2}\partial_\alpha \partial_\beta L_{true} \delta\theta_\alpha \delta\theta_\beta && (10)\\
&= \frac{\eta^2}{2}\underbrace{\mathbb{E}_{t\sim D}\Big[\partial_\alpha \log \pi(t) \partial_\beta \log \pi(t) - \frac{\partial_\alpha \partial_\beta \pi(t)}{\pi(t)}\Big]}_{H_{\alpha\beta}} \underbrace{\frac{1}{B^2}\sum_{i,j=1}^{B} \partial_\alpha \log \pi(t'_i) \partial_\beta \log \pi(t''_j)}_{G^{est}_\alpha G^{est}_\beta} && (11)\\
&= \frac{\eta^2}{2}\frac{1}{B^2}\sum_{i,j=1}^{B} \mathbb{E}_{t\sim D}\Big[K_1(t'_i, t) K_1(t, t''_j) - \tilde{K}_2(t'_i, t''_j)\Big] && (12)
\end{aligned}$$

where $\tilde{K}_2(t'_i, t''_j) = \mathbb{E}_{t\sim D}\Big[\partial_\alpha \log \pi(t'_i) \partial_\beta \log \pi(t''_j) \frac{\partial_\alpha \partial_\beta \pi(t)}{\pi(t)}\Big]$

From here on I'll define $S = \log \pi \; S_\alpha = \partial_\alpha S$.

We're interested in the expected value of the change in loss. to understand this we consider the mean and covariance of the noisy gradients

$$G_\alpha = \mathbb{E}_{t\sim D}[G_\alpha^{est}] = \mathbb{E}_{t\sim D}[S_\alpha(t)]$$

$$\frac{1}{B}\Sigma_{\alpha\beta} = \mathbb{E}_{t,t'\sim D}[(G_\alpha^{est} - G_\alpha)(G_\beta^{est} - G_\beta)] = \mathbb{E} = \frac{1}{B}Cov(S_\alpha, S_\beta)$$

Then the expected value of the change in loss, to second order, is

2.3 Second order in learning rate

It is easy to extend this computation to second order in the learning rate.

$$\delta\mathcal{H} = \sum_\alpha \partial_\alpha \mathcal{H}\delta\theta_\alpha + \frac{1}{2}\sum_{\alpha,\beta}\partial_\alpha\partial_\beta\mathcal{H}\delta\theta_\alpha\delta\theta_\beta + \mathcal{O}(\eta^2)$$

We have $\delta\theta_\alpha = -\eta F_\alpha^1 = -\eta\mathbb{E}_t[(SA(t) - \mathbb{E}[SA])\partial_\alpha S]$. So we just need to compute $\partial_\alpha\partial_\beta\mathcal{H}$

$$\begin{aligned}
\partial_\alpha\partial_\beta\mathcal{H} &= -\partial_\alpha\partial_\beta\sum_t \pi(t)S(t) && (13)\\
&= -\partial_\alpha\sum_t \partial_\beta\pi(t)S(t) && (14)\\
&= -\sum_t \pi(t)\partial_\alpha S(t)\partial_\beta S(t) - \sum_t \pi(t)S(t)\frac{\partial_\alpha\partial_\beta\pi(t)}{\pi(t)} && (15)\\
&= -\mathbb{E}_t[\partial_\alpha S(t)\partial_\beta S(t)] - \mathbb{E}_t[S(t)\pi(t)^{-1}\partial_\alpha\partial_\beta\pi(t)] && (16)\\
&= -\mathbb{E}_t[\partial_\alpha S(t)\partial_\beta S(t)] - \mathbb{E}_t[(S(t) - \mathbb{E}[S])\pi(t)^{-1}\partial_\alpha\partial_\beta\pi(t)] && (17)\\
& && (18)
\end{aligned}$$

In the last line we subtracted a baseline in the second term.

From here it is straightforward to express the second order variation of the entropy as

$$\delta\mathcal{H}_2 = \frac{\eta^2}{2}(SA - \overline{SA})^T K_2(SA - \overline{SA})$$

where again $SA - \overline{SA}$ is a vector, which we should think of as living in a function space, where the functions map token sequences to $\mathbb{R}$, and since the set of possible sequences is finite, this is a finite dimensional vector space.

The kernel K_2 is equal to (where I'll unfortunately be inconsistent with notation and write the first order kernel $K = \sum_\alpha \partial_\alpha S(t)\partial_\alpha S(t') = K_1(t,t')$

$$K_2(t,t') = \sum_{t''} K_1(t,t'')K_1(t'',t') + \sum_{\alpha,\beta}\mathbb{E}_{t''}[(S - \overline{S})\pi(t'')^{-1}\partial_\alpha\partial_\beta\pi(t'')]\partial_\alpha S(t)\partial_\beta S(t')$$

or, to use a more matrixy notation with $W_{\alpha t} = \partial_\alpha S(t)$

$$K_2(t, t') = K_1^2 + W^T \tilde{K} W$$

The variance of the estimator for $\delta\mathcal{H}_1$ is clearly related to this second order term. It would be good to take the approach of Kaplan et al in their optimal batch size paper to analyze this further.

2.3.1 Next steps