links:
tags: #ideas #project

DRA analytical gradient

The reason why DRA takes forever to allocate resources is because the gradient is super noisy. One way in which we stabilize the gradient update is by taking many samples of the entire trajectory to reduce the variance of the gradient update.

Stabilizing DRA's gradient update

We can try to do so numerically or analytically.

Numerical approach

If we were to follow the numerical approach, we might be able to do much better if, say, for each step, we compute the relevant terms for the gradient, $\beta\zeta$ and $\beta\zeta(1-\pi)$, on average rather than for a single sample. It is possible that this turns out to be the same thing in the end if we compare N trajectories sampled with single-sampling of steps vs. N-samples per step for a single trajectory (the one followed on the trial). Check to verify.

Analytical approach

Alternatively, we can compute the expected ζ and π analytically and substitute that directly into the update. The expectation of π can be computed as in here, and it turns out that for two options, it is $1-\Phi\left(\frac{-(\mu_1-\mu_2)}{\sqrt{\sigma_1^2+\sigma_2^2}}\right)$, where $\Phi(\cdot)$ is the CDF of the standard normal distribution, and (μ_i, σ_i) for $i \in \{1,2\}$ are the mean and std dev of the noisy values encoded in memory.

To obtain ζ, we would have to evaluate the following the integral $\int x_1 p(X_1|X_1 - X_2 > 0)dx$, followed by normalizing it with the respective mean and standard deviation.

Turns out that this is possible and it is:
$\zeta = \frac{\sigma_1}{\pi}\phi\left(\mu_1; \mu_2, \sqrt{\sigma_1^2+\sigma_2^2}\right)$,
where $\phi(x;\mu,\sigma)$ is the Gaussian pdf and $\pi = 1-\Phi\left(\frac{-(\mu_1-\mu_2)}{\sqrt{\sigma_1^2+\sigma_2^2}}\right)$ as mentioned before.

Unfortunately, we cannot do this analytically for more than 2 options. We would have to do the following integral numerically:
$\int dx_i p(x_i)\prod_{j\neq i} p(x_j < x_i)$, where $p(x_j < x_i) = \int_{-\infty}^{x_i} dx_j p(x_j)$. I could try to compute this integral numerically to check the tradeoff in speedup vs stability compared to the current version of DRA. This might come in handy especially for a neural network implementation of DRA as in DRA vs Max Ent RL.