Paper summary: Stochastic latent actor-critic: deep reinforcement learning with a latent variable model

May 22, 2022

1 Idea in few sentances

2 Explanation of the central concept

3 Methodology

4 Initial rambly notes

4.1 Abstract

Stochastic latent actor-critic (SLAC): unifies stochastic sequential models and RL into a single method by learning a companct latent representation and then performing RL in the model's learned latent space. In continuos actions spaces, on images, simultaneous training of representations and policy, latent representation and latent-space dynamics are learned jointly.

4.2 Introduction

A predictive model is trained. It combines learning stochastic sequential models and RL into a single model, performing RL in the model's learned latent space. By formalizing the control problem as an inference problem within a POMDP, it can be show that variational inference leads to the objective of the SLAC algorithm. Unlike model-based approaches which compound model error when planning, SLAC performs infinite horizon optimization.

4.2.1 Preliminary: maximum entropy RL in fully observable MDPs

MDP, states $\boldsymbol{s}_t \in \mathcal{S}$, actions $\boldsymbol{a}_t \in \mathcal{A}$, rewards r_t, initial state distribution $p(\boldsymbol{s}_1)$, stochastic transition distribution $p(\boldsymbol{s}_{t+1}\boldsymbol{s}_t, \boldsymbol{a}_t)$. Standard RL aims to learn parameters ϕ under policy $\theta_\phi(\boldsymbol{a}_t|\boldsymbol{s}_t)$ s.t. the expected sum of rewards is maximized

under the induced trajectory distribution ρ_π. This can be modified to include an entropy term, such that the policy also maximizes the expected entropy $\mathcal{H}(\pi_\phi(\cdot|\boldsymbol{s}_t))$. This is built on here. The resulting objective in maximum entropy RL is:

$$\sum_{t=1}^{T} \mathbb{E}_{(\boldsymbol{s}_t,\boldsymbol{a}_t)\sim\rho_\pi} \left[r(\boldsymbol{s}_t,\boldsymbol{a}_t) + \alpha\mathcal{H}(\pi_{\phi(\cdot|\boldsymbol{s}_t)})\right] \tag{1}$$

where α is the temperature parameter for balancing between reward and entropy contributions to the objective. SAC uses this to derive soft policy iteration. SAC has the critic Q_θ and actor π_ϕ The soft Q-function parameters θ are optimized to minimize the soft Bellman residual:

$$J_Q(\theta) = \frac{1}{2}\left(Q_\theta(\boldsymbol{s}_t,\boldsymbol{a}_t) - \left(r_t + \gamma\mathbb{E}_{\boldsymbol{a}_{t+1}\sim\pi_\phi}\left[Q_{\bar{\theta}}(\boldsymbol{s}_{t+1},\boldsymbol{a}_{t+1}) - \alpha\log\pi_\phi(\boldsymbol{a}_{t+1}|\boldsymbol{s}_{t+1})\right]\right)\right)^2 \tag{2}$$

where $\bar{\theta}$ are delayed parameters. The policy parameters ϕ are optimized to update the policy towards the exponential of the soft Q-function, resulting in policy loss:

$$J_\pi(\phi) = \mathbb{E}_{\boldsymbol{a}_t\sim\pi_\phi}\left[\alpha\log(\pi_\phi(\boldsymbol{a}_t|\boldsymbol{s}_t)) - Q_\theta(\boldsymbol{s}_t,\boldsymbol{a}_t)\right] \tag{3}$$

4.2.2 Preliminary: sequential latent variable models and amortized variational inference in POMDPs

Latent variable models with amortized variational inference are used. Image is $\boldsymbol{x}$, generated latent representation is $\boldsymbol{z}$. The model is learned by maximizing the probability of each observed $\boldsymbol{x}$ from some training set under the entire generative process

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z} \tag{4}$$

This is intractable to compute due to the marginalization of the latent variables $\boldsymbol{z}$. In amortized variational inference, the evidence lower bound for log-likelihood is used:

$$\log p(\boldsymbol{x}) \geq \mathbb{E}_{\boldsymbol{z}\sim q}\left[\log p(\boldsymbol{z}|\boldsymbol{x})\right] - D_{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) \tag{5}$$

$\log p(\boldsymbol{x})$ is maximized by learning an encoder $q(\boldsymbol{z}|\boldsymbol{x})$ and a decoder $p(\boldsymbol{x}|\boldsymbol{z})$ and directly performing gradient descent of the right hand side of the equation. Then the distributions of interest are the prior $p(\boldsymbol{z})$, the observation model $p(\boldsymbol{x}|\boldsymbol{z})$ and the variational approximate posterior $q(\boldsymbol{z}|\boldsymbol{x})$.

To extend such models to sequential decision making, actions and temporal structure moust by incorporated to the latent state. Since we're now in the POMDP setting, we don't have states, only observations $\boldsymbol{x}_t \in \mathcal{X}$ and latent states $\boldsymbol{z}_t \in \mathcal{Z}$. The transition distributions are now $p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t,\boldsymbol{a}_t)$ and the observation model is $p(\boldsymbol{x}_t|\boldsymbol{z}_t)$. As VAEs, a generative model can be learned by maximizing the log-likelihood. Importantly, $\boldsymbol{x}_t$ alone can not give $\boldsymbol{z}_t$ — for that previous observations also must be taken into account. Hence the need for sequential latent variable models. The distributions of interest are $p(\boldsymbol{z}_1)$ and $p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t,\boldsymbol{a}_t)$, the observation model $p(\boldsymbol{x}_t|\boldsymbol{z}_t)$ and the approximate variational

posteriors $q(\boldsymbol{z}_1|\boldsymbol{x}_1)$ and $q(\boldsymbol{z}_{t+1}|\boldsymbol{x}_{t+1}, \boldsymbol{z}_t, \boldsymbol{a}_t)$. With these, the log-likelihood of the observations ca be bounded:

$$\log p(\boldsymbol{x}_{1:\tau+1}|\boldsymbol{a}_{1:\tau}) \geq \mathbb{E}_{\boldsymbol{z}_{1:\tau+1}\sim q}\left[\sum_{t=0}^{\tau} \log p(\boldsymbol{x}_{t+1}|\boldsymbol{z}_{t+1}) - D_{KL}\left(q(\boldsymbol{z}_{t+1}|\boldsymbol{x}_{t+1}, \boldsymbol{z}_t, \boldsymbol{a}_t)||p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t, \boldsymbol{a}_t)\right)\right] \tag{6}$$

Where a few obvious conditionals have been dropped. Prior work has dealt with these non-Markovian state transitions with recurrent networks or probabilistic state-space models. Here, a stochastic latent variable model is trained.

4.3 Joint modelling and control as inference

In the MDP setting, the control problem can be embedded into the MDP graphical model by introducing a binary random variable $\mathcal{O}_t$ which indicates if the time step t is optimal. If it's distribution is chosen to be $p(\mathcal{O}_t = 1|\boldsymbol{s}_t, \boldsymbol{a}_t) = \exp(r(\boldsymbol{s}_t, \boldsymbol{a}_t))$, then the maximization of $p(\mathcal{O}_{1:T})$ via approximate inference in that model yields the optimal policy for the maximum entropy objective. Now this is extended to the POMDP setting. Analogously, we get $p(\mathcal{O}_t = 1|\boldsymbol{z}_t, \boldsymbol{a}_t) = \exp(r(\boldsymbol{z}_t, \boldsymbol{a}_t))$. But now not only the likelihood of optimality variables in maximized, but also the maximum entropy policies, thus maximizing the marginal likelihood $p(\boldsymbol{x}_{1:\tau+1}, \mathcal{O}_{\tau+1:T}|\boldsymbol{a}_{1:\tau})$, in order to learn the model and the policy at the same time. This objective represent both the likelihood of the observed data from the past $\tau + 1$ steps, as well as the optimality of the agent's action for future steps, hence combining both representation learning and control into a single graphical model.

The variational distribution is factorized into a product of *recognition* terms $q(\boldsymbol{z}_{t+1}|\boldsymbol{x}_{t+1}, \boldsymbol{z}_t, \boldsymbol{a}_t)$, *dynamics terms* $p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t, \boldsymbol{a}_t)$ and *policy* terms $\pi(\boldsymbol{a}_t|\boldsymbol{x}_{1:t}, \boldsymbol{a}_{1:t-1})$:

$$q(\boldsymbol{z}_{1:T}, \boldsymbol{a}_{\tau+1:T}|\boldsymbol{x}_{1:\tau+1}, \boldsymbol{a}_{1:\tau}) = \prod_{t=0}^{\tau} q(\boldsymbol{z}_{t+1}|\boldsymbol{x}_{t+1}, \boldsymbol{z}_t, \boldsymbol{a}_t) \prod_{t=\tau+1}^{T-1} p(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t, \boldsymbol{a}_t) \prod_{t=\tau+1}^{T} \pi(\boldsymbol{a}_t|\boldsymbol{x}_{1:t}, \boldsymbol{a}_{1:t-1}) \tag{7}$$

The dynamics for future timesteps is used to prevent the agent from taking optimistic steps. This posterior can then be used to obtain the evidence lower bound (ELBO) of the likelihood. It is in turn separated into training the latent variable model and maximized the entropy of RL. I'm not going into this as I can't understand it without much more effort. The latent model is a specific VAE and I don't exactly understand the vannila VAE anyway. Also I never derived SAC and I don't know anything about messaging passing in statistics so I certainly can't follow how the maximum entropy actor-critic is employed here. Let's call it good because unlike other more complex generative models it actually integrates them into the POMDP (which is the case, but I have to take it on belief here).

4.4 Method

4.5 Other stuff