Latent Space Policies for Hierarchical Reinforcement Learning

Tuomas Haarnoja, Kristian Hartikainen, Pieter Abbeel, Sergey Levine

***Abstract*—The paper proposes a novel method for learning hierarchical neural network policies by using a maximum entropy RL objective and augmenting latent random variables to each layer allowing higher layers to control lower layer behaviours and each layer is trained directly to solve the task.**

I. Preliminaries

Notation : $\tau = (s_0, a_0, \ldots, s_T)$ denotes a trajectory, $\rho_\pi(\tau)$ denotes its distribution under policy $\pi(a_t|s_t)$ and $p(s_0)$ denotes the initial state distribution. (Reward is bounded)

Maximum Entropy RL : The entropy augmented objective is given by :

$$J(\pi) = \mathbf{E}_{\tau\sim\rho_\pi(\tau)}\Big[\sum_t r(s_t, a_t) + \alpha\mathcal{H}(\cdot|s_t))\Big] \quad (1)$$

where α controls the stohasticity of the optimal policy.

Probabilistic Graphical Model for Control : The hierarchical model is composed of factors for the dynamics $p(s_{t+1}|s_t, a_t)$ and the action prior $p(a_t)$ is set to be Gaussian. The distribution over the optimal trajectories is given by :

$$p(\tau|\mathcal{O}_{0:T}) \propto p(s_0)\prod_{t=0}^{T} p(a_t)p(s_{t+1}|s_t, a_t)\exp(r(s_t, a_t)) \quad (2)$$

where $\mathcal{O}_t$ is the *optimality variable*, a binary random variable denoting whether state-action tuple at the time step was optimal and $p(\mathcal{O}_t|s_t, a_t) = \exp(r(s_t, a_t))$ assuming $r(s_t, a_t) < 0$.

Variational Inference : The posterior is approximated with a probabilistic model that constrains the policy to a parameterised distribution. Variational distribution is defined as :

$$q(\tau) = p(s_0)\prod_{t=0}^{T}\pi(a_t|s_t)p(s_{t+1}|s_t, a_t))$$

Fitting the distribution by maximising ELBO gives a KL divergence between q and p and simplifies the equatoin to give :

$$J(\pi) = \mathbf{E}_{\tau\sim\rho_\pi(\tau)}\Big[\sum_t r(s_t, a_t) - \mathrm{D_{KL}}(D(\pi(\cdot|s_t) \parallel p(\cdot)))\Big]$$

With a a uniform action prior, the equation becomes the maximum entropy objective (1).

II. Theory

To solve the optimal control problem, we have to infer the posterior action distribution $\pi^*(a_t|s_t) = p(a_t|s_t, \mathcal{O}_{t:T} = true)$. But the optimal action distribution found from (2) is not directly used as policy since it would give rise to an overly optimistic policy (stochastic state transitions can be modified for optimal behavior) and in continuous domains, the optimal policy is intractable (approximated by a distribution). When using variational inference to determine $p(a_t|s_t, \mathcal{O}_{t:T})$, (2) reduces to the maximum entropy reinforcement learning problem (1).

III. Architecture and Training details

The policy-augmented graphical model has a parameterised distribution over actions which is conditioned on latent variables, ie., the policy has 2 factors : $\pi(a_t|s_t, \boldsymbol{h}_t)$ and prior $p(\boldsymbol{h}_t)$ where $\boldsymbol{h}_t$ represents the latent variable which is sampled before the action. The training is proceeded in a bottom-up approach, where the lower level policy is trained first, and then used to provide a higher level action space ($\boldsymbol{h}_t$) for a higher level policy ($p(s_{t+1}|s_t, \boldsymbol{h}_t)$ obtained by marginalising out the actions) which solves a simpler problem and results in better performance. Each layer is trained on the same maximum entropy objective simplifies the task for the layer above it. It is to be noted that the higher levels in the hierarchy are given the power to fully invert the behavior of the lower layers. The process is repeated multiple times by adding new policy layers (called *sub-policies*) and learning new higher-level policies on top of the previous policy's latent space, constructing deeper hierarchical policy representation. For difficult and challenging problems, lower layers of the hierarchy were trained on shaped reward functions to bring forth desirable behaviors. The sub-policies must be tractable, they shouldn't suppress information flow from higher to lower levels and the conditional factor of each sub-policy should be deterministic. The authors model the conditionals based on a bijective transformations (since the higher layer can undo any behavior in the lower layer depending on the task requirement) of the latent variables into actions. The change of variable formula is given by :

$$\pi(a_t|s_t) = p(\boldsymbol{h}_t)\left|\det(\frac{df(\boldsymbol{h}_t; s_t)}{d\boldsymbol{h}_t})\right|^{-1} \quad (3)$$

where $a_t = f(\boldsymbol{h}_t; s_t)$. The transformations can be easily chained to form multi-level policies, and can be trained end-to-end as a single policy or layerwise as a hierarchical policy without the need to design the policy topology from scratch depending on the problem to be solved. The authors use soft actor-critic (robust & better sample-efficiency) for optimizing the policy. There is provision for giving different shaped reward distributions at each layer (and learn a maximum entropy policy to optimize the corresponding variational inference objective) with the last layer reward function corresponding to the actual task to be solved. Hierarchies are constructed in layerwise fashion, by training one latent variable policy at a time.