Paper summary: Image augmentation is all you need: regularizing deep reinforcement learning from pixels

May 21, 2022

1 Idea in few sentences

2 Explanation of the central concept

3 Methodology

4 Initial rambly notes

4.1 Abstract

The method is dubbed DrQ: Data regularized Q. Can be combined with any model-free RL alg. Works better than CURL.

4.2 Introduction

Direct quote:

> Simultaneously training a convolutional encoder alongside a policy network is challenging when given limited environment interaction, strong correlation between samples and a typically sparse reward signal. Naive attempts to use a large capacity encoder result in severe over-fitting, and smaller encoders produce impoverished representations that limit task performance.

The authors identify that you can go with the following:

1. pretraining with self-supervised learning (SSL), followed by supervised learning

2. supervised learning with an additional auxiliary loss

3. supervised learning with data augmentation

The problem is, in sample-efficient RL you're working with $10^4 - 10^5$ transitions from a few hundred trajectories. The authors opt for a data augmentation approach. This is used, of course, generate more samples without sampling trajectories, but also to regularize the Q-function — and nothing more. Hence, no additional losses are needed.

4.3 Method

The idea is to use optimality invariant state transformations. This is defined as a mapping $f : \mathcal{S} \times \mathcal{T} \rightarrow \mathcal{S}$ that preserves the Q-values:

$$Q(\boldsymbol{s}, \boldsymbol{a}) = Q(f(\boldsymbol{s}, \boldsymbol{\nu}), \boldsymbol{a}), \forall \boldsymbol{s} \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A} \text{ and } \boldsymbol{\nu} \in \mathcal{T} \tag{1}$$

where $\boldsymbol{\nu}$ are the parameters of $f(\cdot)$ drawn from the set of all possible parameters $\mathcal{T}$,ex. random image translations. This enables variance reduction in Q-value estimates. This works as follows. For an arbitrary distribution of states $\mu(\cdot)$ and policy π, instead of using a single sample $\boldsymbol{s}^* \sim \mu(\cdot), \boldsymbol{a}^* \sim \pi(\cdot|\boldsymbol{s}^*)$, with the expectation:

$$\mathbb{E}_{\boldsymbol{s}\sim\mu(\cdot),\boldsymbol{a}\sim\pi(\cdot|\boldsymbol{s})}\left[Q(\boldsymbol{s}, \boldsymbol{a})\right] \approx Q(\boldsymbol{s}^*, \boldsymbol{a}^*) \tag{2}$$

we can generate K samples via random transformations and use the following estimate with lower variance:

$$\mathbb{E}_{\boldsymbol{s}\sim\mu(\cdot),\boldsymbol{a}\sim\pi(\cdot|\boldsymbol{s})}\left[Q(\boldsymbol{s}, \boldsymbol{a})\right] \approx \frac{1}{K}\sum_{k=1}^{K} Q(f(\boldsymbol{s}^*, \boldsymbol{\nu}_k), \boldsymbol{a}_k) \tag{3}$$

where $\boldsymbol{\nu}_k \in \mathcal{T}$ and $\boldsymbol{a}_k \sim \pi(\cdot|f(\boldsymbol{s}^*, \boldsymbol{\nu}_k), \boldsymbol{a}_k)$. This suggest two distinct ways to regularize the Q-function. First, use data augmentation to compute target values for every transition tuple $(\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}'_i)$ as

$$y_i = r_i + \gamma\frac{1}{K}\sum_{k=1}^{K} Q_\theta(f(\boldsymbol{s}'_i, \boldsymbol{\nu}'_{i,k}), \boldsymbol{a}'_{i,k}) \tag{4}$$

where $\boldsymbol{a}'_{i,k} \sim \pi(\cdot|f(\boldsymbol{s}'_i, \boldsymbol{\nu}'_{i,k}))$ and where $\boldsymbol{\nu}'_{i,k} \in \mathcal{T}$ corresponds to a transformation parameter of $\boldsymbol{s}'_i$. Then the Q-function is updated using these targets through an SGD update with learning rate λ_θ:

$$\theta \leftarrow \theta - \lambda_\theta \nabla_\theta \frac{1}{N}\sum_{i=1}^{N}(Q_\theta(f(\boldsymbol{s}_i, \boldsymbol{\nu}_i), \boldsymbol{a}_i) - y_i)^2 \tag{5}$$

4 can also be used for different augmentations of $\boldsymbol{s}_i$, resulting in the second regularization approach:

$$\theta \leftarrow \theta - \lambda_\theta \nabla_\theta \frac{1}{NM}\sum_{i=1,m=1}^{N,M}(Q_\theta(f(\boldsymbol{s}_i, \boldsymbol{\nu}_{i,m}), \boldsymbol{a}_i) - y_i)^2 \tag{6}$$

where $\boldsymbol{\nu}_{i,m}$ and $\boldsymbol{\nu}'_{i,k}$ are drawn independently.

4.4 DrQ

If $[K = 1, M = 1]$ is exactly RAD up to choice of hyper-parameters and data augmentation functions. DrQ uses $[K = 2, M = 2]$.