RL Summary EN

Georg Pernice, ..

12.02.2026

Contents

1 Preword

Obtain the Latex source code for this document here to compile translate it or create your own version: https://github.com/g14p/learninglatex

2 00 - Orga and Intro

3 01 - Bandits and Explore

4 02 - Opt. Dec. Making

5 04 - Value Based Meth.

6 05 - Policy Gradient

7 06 - Offpolicy Actor critic

8 07 - Control as Inference

9 08 - Trust Region Methods

10 09 - Imitation Learning

11 10 - Offline RL

12 12 - Motion Primitives and Temp. Extend. Actions

13 13 - Trafos

14 14 - Diffusion

15 14b Diffusion2

16 15 - Diverse Behavior and Skill Discovering

17 16 - Diffusion RL and Scaling RL

17.1 Self Check Questions

- What is problem when reusing off-policy data more often to update network parameters?
  - Reusing off-policy data corresponds to increasing Replay Ratio or Update to Data (UTD) Ratio. The fact that train data **and** tar-

gets change over time states a problem, leading to Placity Loss and Primary Bias. (See them explained below)

- How to counteract the primacy bias?
  - **Primacy Bias** = tendency to overfit initial experiences which damages the learn proess regarding other experiences. Resetting the environment in intervals helps especially when running SAC algorithm in 'humanoid run' environment.
- What kinda regularization exist to make RL scalable?
  - Regularization in ML refers to prevention of overfitting by punishing model complexity with additional loss term. (Ideas like Dropout, Punishing model size, ..)
  - The BRO (Bigger Regularized Optimistic) algorithm uses
    * Weight Decay
    * Layer Norm
    * ..

    It scales the Network by multiple blocks of
    * Dense Layer
    * Layer Norm
    * ..
  - In general in RL Neuron Resetting seems a valid regularization, used in the 'ReDo' (Recycling Dormant Neurons) algorithm.
  - **Batch Normalization** acc. to Summary slide seems to be as well a regularization technique in RL
- What is the downside to increasing the number of updates?
  - Updating more often even does harm to the network return. As the targets will change updating too much on them doesnt make lots of sense. $\rightarrow$ Placicity Loss(lose ability to learn from new XP) and Primary Bias (overfit initial XP) occur! This question seems very similar to the first one imho.

18 Exercises with Math

Exercise 2: Policy differentiation Compute the gradient of $logp_\theta(tau)$, where $logp_\theta(tau)$: trajectory distribution induced by params theta theta : policy parameters

18.1 Multivariate Policy Exercise 3

Consider multi-variate policy distribution $\pi(a|s) = \frac{1}{\sqrt{(2\pi)^{d_e}|\sum|}}\, exp\left\{-\frac{1}{2}(a-\mu(s))^T \sum^{-1}(a-\mu(s))\right\}$ where $a \in \mathbb{R}^{d_e}$, $s \in \mathbb{R}^{d_e}$ and we consider an isotropic covariance i.e. $\sum = \sigma^2\mathbf{I}$ where $\mathbf{I} \in \mathbb{R}^{d_e \times d_e}$ and $\sigma \in \mathbb{R}^+$

Derive $\nabla_\sigma\ log\ \pi(a|s)$ **Solution Approach:**

$$\nabla_\sigma log\pi(a|s) = \nabla_\sigma log\left\{\frac{1}{\sqrt{(2\pi)^{d_e}|\sum|}} exp\left\{-\frac{1}{2}(a-\mu(s))^T \sum^{-1}(a-\mu(s))\right\}\right\}$$

Split the logs the exp dies because of right log. Because we get a sum of two logs we can split the gradient as well.

$$\nabla_\sigma log\pi(a|s) = \nabla_\sigma log\left\{\frac{1}{\sqrt{(2\pi)^{d_e}|\sum|}}\right\} + \nabla_\sigma\left\{-\frac{1}{2}(a-\mu(s))^T \sum^{-1}(a-\mu(s))\right\}$$

With the two identities $\sum^{-1} = \frac{1}{\sigma^2}\mathbf{I}$ and $|\sum| = (\sigma^2)^{d_e}$ for the covariance we arrive at:

$$\nabla_\sigma log\pi(a|s) = \nabla_\sigma log\left\{\frac{1}{\sqrt{(2\pi)^{d_e}|(\sigma^2)^{d_e}|}}\right\} + \nabla_\sigma\left\{-\frac{1}{2}\frac{1}{\sigma^2}(a-\mu(s))^T\mathbf{I}(a-\mu(s))\right\}$$