Exploration by Random Network Distillation, ICLR 2019

Yuri Burda & Harrison Edwards et al.

Contents

1. Introduction
2. Method
3. Experiments
4. Code review

Introduction

- Exploration is important in RL
  - Rewards are often sparse and hard to find
    - ATARI: Montezuma Revenge
  - Successful policy learning requires good trajectory samples
  - How humans perform trial-and-error for improving (or discovering) their skills ?

Introduction

- How to quantify the novelty of new experience ?
    - Imagine a next observation predictor for current observation and action

Method

1. Exploration Bonuses

$$r_t = e_t + i_t$$

- It is desirable for i_t to be higher in novel state in frequently visited one
- Previous exploration methods are difficult to scale-up
    - Count based
        - ex) $i_t = \frac{1}{\mathrm{n_t(s)}}$ in a tabular setting
- Prediction Error (related to agent's transitions) based

Method

2. Random Network Distillation

- Auxiliary networks for producing intrinsic reward
  - A fixed and randomly initialized target state embedding network $f\colon \mathcal{O} \rightarrow \mathbb{R}^k$
  - A predictor $\hat{f}\colon \mathcal{O} \rightarrow \mathbb{R}^k$
  - Distillation loss on $\hat{f}$: $\min_{\theta} \left\| \hat{f}(x;\theta) - f(x) \right\|^2$
- The prediction error $\left\| \hat{f}(x;\theta) - f(x) \right\|^2$ is the intrinsic reward i_t
  - It is expected to be higher for novel state dissimilar to the ones the predictor has been trained on.

Method

MNIST Toy example

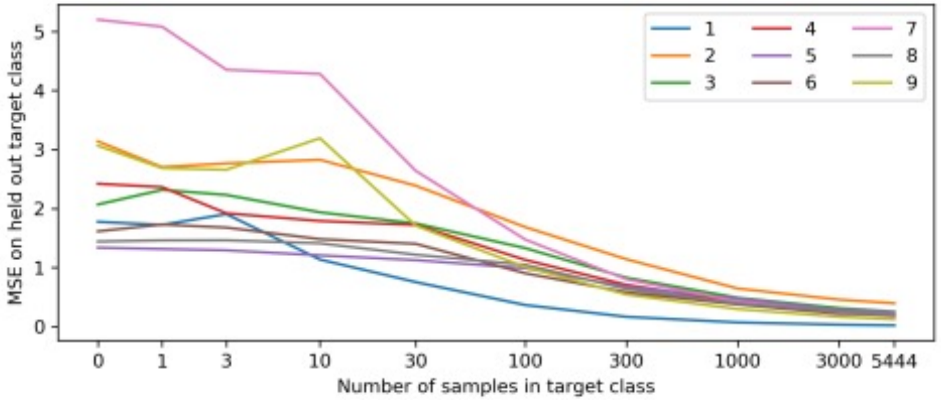

Figure 2: Novelty detection on MNIST: a predictor network mimics a randomly initialized target network. The training data consists of varying proportions of images from class "0" and a target class. Each curve shows the test MSE on held out target class examples plotted against the number of training examples of the target class (log scale).

- Tested the predictor on unseen test examples
- After train the predictor with label 0 and target class (not 0) varying the proportion of the classes

Method

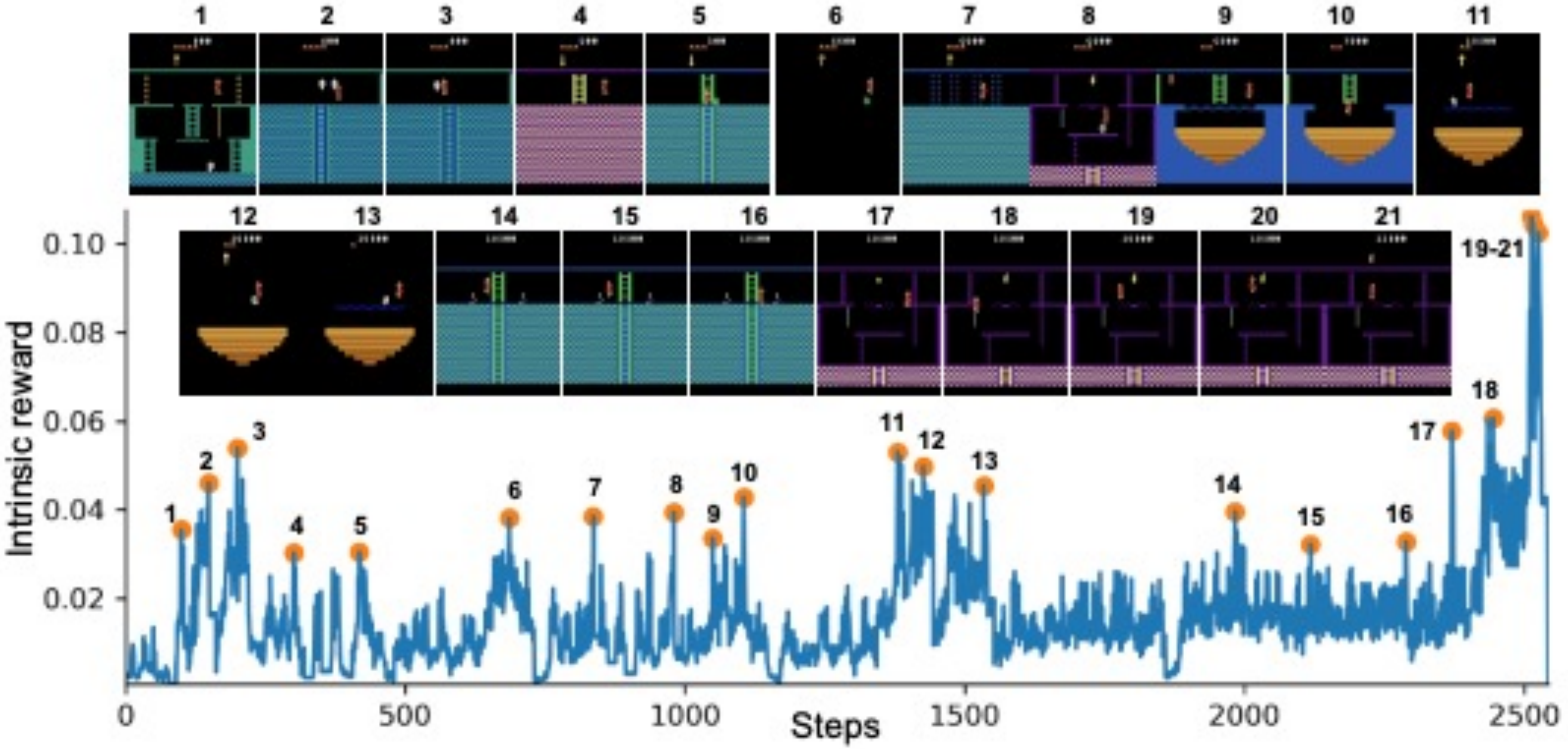

Figure 1: RND exploration bonus over the course of the first episode where the agent picks up the torch (19-21). To do so the agent passes 17 rooms and collects gems, keys, a sword, an amulet, and opens two doors. Many of the spikes in the exploration bonus correspond to meaningful events: losing a life (2,8,10,21), narrowly escaping an enemy (3,5,6,11,12,13,14,15), passing a difficult obstacle (7,9,18), or picking up an object (20,21). The large spike at the end corresponds to a novel experience of interacting with the torch, while the smaller spikes correspond to relatively rare events that the agent has nevertheless experienced multiple times. See here for videos.

Method

Some descriptions about RND

- Prediction errors can be attributed following 4 factors:
  1. Amount of training data – desirable factor
  2. Stochasticity
  3. Model misspecification
  4. Learning dynamics

  RND tackles 2, 3 since the target networks can be chosen to be deterministic and inside the model-class of the predictor.

- Distillation error could be seen as a quantification of uncertainty in predicting the constant zero function

Let $\mathcal{F}$ be the distribution over functions $g_\theta = f_\theta + f_{\theta^*}$, where θ^* is drawn from $p(\theta^*)$ and θ is given by minimizing the expected prediction error

$$\theta = \arg\min_\theta \mathbb{E}_{(x_i, y_i)\sim D} \| f_\theta(x_i) + f_{\theta^*}(x_i) - y_i \|^2 + \mathcal{R}(\theta), \tag{1}$$

Method

3. Dual values

- Combining episodic & non-episodic reward
- Each value network for extrinsic and intrinsic rewards with different discounting factors
- $V = V_E + V_I$

4. Normalization

- observations: ((x − x.mean)/x.std).clip(-5, 5)
- intrinsic rewards: int_r/int_r.std

Experiments

	Gravitar	Montezuma's Revenge	Pitfall!	PrivateEye	Solaris	Venture
RND	**3,906**	**8,152**	-3	8,666	3,282	**1,859**
PPO	3,426	2,497	0	105	3,387	0
Dynamics	3,371	400	0	33	3,246	1,712
SOTA	2,209[1]	3,700[2]	**0**	**15,806**[2]	**12,380**[1]	**1,813**[3]
Avg. Human	3,351	4,753	6,464	69,571	12,327	1,188

Table 1: Comparison to baselines results. Final mean performance for various methods. State of the art results taken from: [1] (Fortunato et al., 2017) [2] (Bellemare et al., 2016) [3] (Horgan et al., 2018)

Code review