Exploration in Reinforcement Learning

An Introduction and Comparison of Three Recent Methods

Jan Rodríguex Miret

June 21, 2021

Contents

1 Introduction

Reinforcement Learning is increasingly becoming a hot topic and it is gaining more and more attention from researchers at universities and companies, especially with the recent techniques that use deep neural networks.

Nonetheless, there are lots of challenges that are still waiting to be solved, despite the efforts and advancements of the community, which include the ability to abstract knowledge and easily transfer it to other tasks, and to efficiently explore hard environments.

There is always a trade-off between exploiting and exploring in these agents, since the more the agent follows a given policy because it is the best action it can do, the less it will explore. On the other hand, it does not make sense to only explore and never select the best action because it will be the same as a completely random agent, and it will be impossible to learn anything either.

Nowadays, many techniques use intrinsic motivation (IM) as a bonus reward to train an agent. With this, the agent is not only looking to optimize the policy such as the reward observed from the environment is the highest, but also is encouraged to put itself in more unknown scenarios, which will be more useful for its learning.

There are lots of intrinsic motivators and many approaches to define the exploratory behaviour of an agent. Some of the most recent works in the field are presented in this document, which aim to provide some insight of the current solutions and limitations of these techniques.

In order to have a clearer view of the different kind of approaches in the literature, we will explain three different methods for exploration in deep reinforcement learning in Sections 2, 3 and 4. Some possible applications and advantages are mentioned in each method. Finally, a comparison between them regarding their similarities and differences are discussed in Section 5, along with some conclusions of the work.

2 Surprise-Based Intrinsic Motivation for Deep Reinforcement Learning

The first exploratory technique analyzed is "Surprise-Based Intrinsic Motivation for Deep Reinforcement Learning" [1], which was published by Achiam and Sastry from the University of California, Berkeley, in 2017.

In their work, they propose a new surprise-based approach that consists of two components. The first one is a model that learns the Markovian Decision Process (MDP)[1] transition probabilities while the second one is the policy itself. These two are learned concurrently in alternative update steps.

Surprise is formulated as the Kullback-Leibler (KL) divergence between the true transition probabilities and the learned model. This divergence is approximated by two different methods, which have similar but not identical behaviors. Neither of these approximations are computationally expensive.

[1] Along this document the common notation in the literature is used. An MDP is a tuple (S, A, R, P, μ), where S is the set of states, A the set of actions, R the reward function, P the transition probability function, and μ the starting state distribution.

The dynamics model is trying to be optimized by the following equation at each step:

$$\min_{\phi} -\frac{1}{|D|} \sum_{(s,a,s') \in D} log P_{\phi}(s'|s,a) + \alpha f(\phi), \tag{1}$$

where D is the dataset of transition tuples, P_{ϕ} is the model, f is a function used for regularization, and α is a regularization trade-off coefficient greater than zero.

On the other hand, the policy is updated accordingly to the following optimization problem:

$$\max_{\pi} L(\pi) + \eta E_{s,a \sim \pi}[D_{KL}(P||P\phi)[s,a]], \tag{2}$$

where $L(\pi)$ is the typical performance measure, η is a trade-off coefficient between exploration and exploitation. Note that apart from the usual performance L, the additional exploration incentive is used (second addend of the sum) for the optimization of the policy.

As already mentioned, by using the KL-divergence between P (the true transitions) and P_{ϕ} (what the model expected), the surprise of the agent can be computed. By optimizing Equation (2), the agent is encouraged to explore spaces for which it was not able to predict a correct transition.

The problem is that P is unknown in practice, this is why two approximations for the KL-divergence were proposed. The first one is to only use the surprisal of s' with the given context (s,a) as intrinsic motivation. The reshaped reward can be formulated as follows:

$$r'(s,a,s') = r(s,a,s') - \eta \log P_{\phi}(s'|s,a) \tag{3}$$

On the other hand, the divergence can be approximated by lower-bounding the surprise term in the objective (2). For that, it needs another set of parameters ϕ'. Their solution to that is to let ϕ' be the currently updated parameters and ϕ be the ones before the last k updates. Then, the reshaped reward for this approximation is:

$$r'(s,a,s') = r(s,a,s') + \eta(\log P_{\phi_t}(s'|s,a) - \log P_{\phi_{t-k}}(s'|s,a)) \tag{4}$$

Note that over time, the exploration vanishes because the dynamic model converges, i.e. $P_{\phi_t} \approx P_{\phi_{t-k}}$. This is something desirable because once the transitions are learned, we only want the agent to be guided using extrinsic rewards instead of focusing on noisy transitions (the ones that are still struggling to predict). However, if using the first approximation, they argue that it is subject to this problem, but it might not happen in practice as long as the agent starts to see some extrinsic rewards before converging the transitions.

The results obtained with both approximations were similar in performance to Variational Information Maximizing Exploration (VIME) [8], which was the state-of-the-art algorithm of the time, but with a much lower execution time to achieve the same results. VIME also uses a surprise approach but in a Bayesian framework.

3 Never Give Up: Learning Directed Exploration Strategies

The second approach that we will look at is the work of Puigdomènech et al. under the name "Never Give Up: Learning Directed Exploration Strategies" [3]. It was published in February 2020 by researchers of DeepMind.

In their approach, which they refer to as NGU (never give up), they use and combine a lot of techniques that have been proposed in these recent years. Basically, they used a novelty-based intrinsic reward that uses two different kinds of novelties. Later, they used the Universal Value Function Approximators (UVFA) framework to simultaneously learn multiple directed exploration policies with a single neural network.

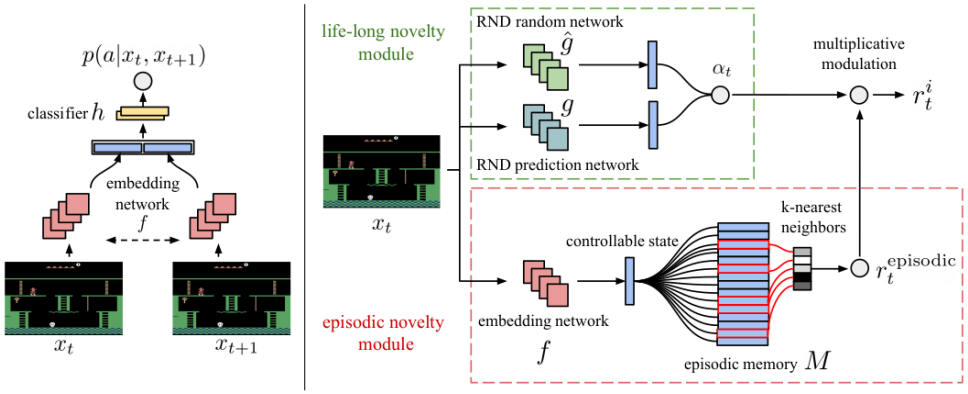


Figure 1: (left) Training architecture for the embedding network (right) NGU's reward generator. Image extracted from the original paper[3].

The first novelty is episodic-wise. With it, the agent is discouraged from returning to the same state or very similar ones during the same episode. This novelty thus ignores the interactions with other episodes. As shown in Figure 1 (right, in red), the episodic novelty module first applies an embedding function to the states $x_t, x_{t+1}, ...$ to obtain what they call the controllable state.

This embedding is carried out by an embedding network (Multi-layer Perceptron with one hidden layer and a softmax) and takes into account the fact that we do not want to capture differences in states that are not provoked by the agent, like in the case of navigating a busy city. The network tries to ignore this variability and just captures what the agent is really affecting.

This controllable state is then used to extract the most similar states already stored in the episodic memory (M) and compute the corresponding reward $r_t^{episodic}$, before being added to M

On the other hand, the life-long novelty module accounts for reducing the probability that a given state is repeatedly visited over the episodes. For that, they use Random Network Distillation [5], which computes the modulator α_t that is later combined into the final reward as follows:

$$r_t^i = r_t^{episodic} \cdot \min\{\max\{\alpha_t, 1\}, L\}, \tag{5}$$

where L is the maximum reward, which was defined as $L = 5$. With this reward, the agent has a balance between this per-episodic novelty and the life-long novelty.

The main focus of this algorithm is on hard exploration problems. It was tested in the most

complex Atari games, where it outperformed the state-of-the-art benchmarks, and achieved very high scores in the games that are not so hard to explore.

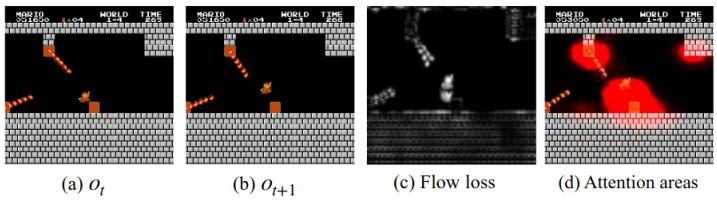
An advantage of this method is that it can be applied to distributed RL agents, meaning that it can be parallelized to hugely decrease the time needed for training.

4 Flow-based Intrinsic Curiosity Module

Lastly, in the work of Yang et al.[11], they propose a prediction-based novelty used by a flow-based intrinsic curiosity module (FICM) to reward the agent with a bonus for exploration.

For the predictions, they leveraged the optical flow between two consecutive frames, trying to make the agent aware of the dynamics of the environment and reward the agent if it puts itself into bad predicted scenarios to further improve the prediction.

An example of the prediction can be seen in Figure 2.



(a) o_t (b) o_{t+1} (c) Flow loss (d) Attention areas

Figure 2: Prediction errors over two consecutive steps (i.e. flow loss) and the attention regions of the agent. Image extracted from the original paper[11]

The objective of FICM is to obtain an exploration method that understands the motion features between consecutive observations, that encodes the observations of the environment in a useful way, and that is able to approximate the novelty of observations.

These motion features are obtained by using optical flow estimation. To formulate the optical flow, the usual approach is to use a vector field with the displacements of each pixel, region, or object between two consecutive frames. These optical flows are used in many computer vision tasks and have been improved recently with deep learning techniques like FlowNet [6] and FlowNet 2.0 [9].

FICM receives two observations o_t and o_{t+1} and then computes the prediction of a forward flow $F_{forward}$ and a backward one $F_{backward}$ using the flow predictor G with parameters Θ_f. Mathematically, they can be formulated as follows:

$$F_{forward} = G(o_t, o_{t+1}, \Theta_f) \tag{6}$$

$$F_{backward} = G(o_{t+1}, o_t, \Theta_f) \tag{7}$$

These flow predictions are used to generate the predicted pixel observations $\hat{o}_t(x)$ and $\hat{o_{t+1}}(x)$ by means of a warping function implemented with a bi-linear interpolation method, similarly to [6] and [9]:

$$\hat{o}_t(x) = o_{t+1}(x + F_{forward}(x) \odot \beta) \tag{8}$$

$$\hat{o_{t+1}}(x) = o_t(x + F_{backward}(x) \odot \beta) \tag{9}$$

where x is the pixel index, β is a scaling factor for the flow, and $\odot$ is the Hadamard product (element-wise multiplication).

From the predicted observations $\hat{o}_t(x)$ and $\hat{o_{t+1}}(x)$, the parameters of the flow predictor can be optimized to minimize the flow loss function L_G, using the mean squared errors (MSE) between real and predicted. The optimization problem can be formulated as:

$$min_{\Theta_f} L_G = min_{\Theta_f}(L^f + L^b) = min_{\Theta_f}(||o_{t+1} - \hat{o_{t+1}}||^2 + ||o_t - \hat{o}_t||^2) \tag{10}$$

In their approach, this L_G is used to quantify the novelty of an observation. Therefore, the intrinsic reward r^i can be shaped as:

$$r^i = r^f + r^b = \frac{\zeta}{2}(L^f + L^b) = \frac{\zeta}{2}(||o_{t+1} - \hat{o_{t+1}}||^2 + ||o_t - \hat{o}_t||^2), \tag{11}$$

where ζ is the reward scaling factor, and r^f and r^b are the intrinsic rewards scaled from L^f and L^b. Significantly, the intrinsic reward is not dependent on the action chosen by the agent. This is a difference between this and the other methods.

The paper proposes two different implementations for FICM: FICM-S and FICM-C, which are depicted in 3. FICM-S encodes a stacked version of the observation, while FICM-C includes an explicit correlation of feature maps and usually outperforms the stacked implementation[9].

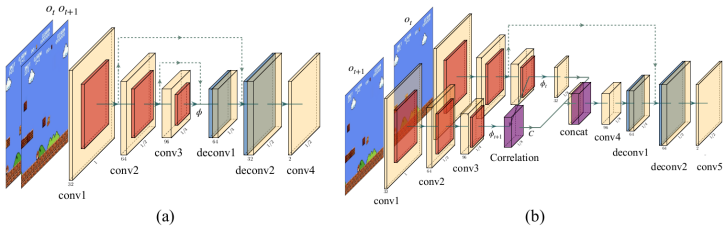


Figure 3: (a) FICM-S and (b) FICM-C architectures for the flow predictor. Image extracted from the original paper[11]

This technique achieved state-of-the-art performance in Atari games, Super Mario Bros, and ViZDoom. FICM is especially useful and good for environments that contain several moving objects.

Another benefit of this technique is that to compute the motion features, it only requires two consecutive frames, while other approaches need more observations to achieve the same level of motion estimation performance.

5 Comparison and conclusions

In this work, some recent advancements in the field of exploration in reinforcement learning have been introduced, as well as the contextualization of the use of these techniques. By having looked at some of the approaches, we can have a clearer view of the actual state of the art and its remaining challenges.

Among these approaches, several similarities can be found as well as many differences. Significantly, all works presented and most works in the field state that exploration is key to success in large-state or large-action spaces and they are a must to address environments with sparse rewards.

Therefore, they are all now using intrinsic motivation to encourage the agent to explore, with more complex rewards that increase the efficiency of exploration by being directed to more useful state-action spaces to them. In the case of Never Give Up and FICM, they both use Random Network Distillation.

Nonetheless, they differ in what the exact motivator should be. Some of them aim to use surprise (Section 2), state novelty (3), flow prediction (4), or empowerment, among others, as the intrinsic motivator.

This field of exploration is related to other fields in reinforcement learning, like *curriculum learning* [4], in which the agent is learning a more complex problem by learning a decomposition of simpler tasks [10].

Although these approaches improve the reusability of tasks in the same domain, there is still much room for improvement. Furthermore, many of these methods lack a good representation of the environment, which could help to fasten the training, and also an abstraction of actions, which are often referred to as *options*[2].

Many of these works on the field are also aiming at building more explainable agents, so that can have a clearer idea of what is going on internally, discover knowledge, and improve things when the agent fails. The attention mechanism shown in Figure 2 is an example.

Some other works regarding exploration also use a Partially Observable Markov Decision Process (POMDP) to model the environment, whenever it is not fully observable, i.e. we lack some information about it. Others use other learning approaches like adversarial-based exploration[7].

To conclude, reinforcement learning is a very active topic and exploration is a key aspect to take into account if the agent has to perform actions in a complex, sparse, large-space environment. Intrinsic motivation is a very good approach to assess this exploration and ensure that it is more efficient but many challenges are still remaining and require new methods to be developed.

References

[1] J. Achiam and S. Sastry. Surprise-based intrinsic motivation for deep reinforcement learning, 2017.

[2] A. Aubret, L. Matignon, and S. Hassas. A survey on intrinsic motivation in reinforcement learning, 2019.

[3] A. P. Badia, P. Sprechmann, A. Vitvitskyi, D. Guo, B. Piot, S. Kapturowski, O. Tieleman, M. Arjovsky, A. Pritzel, A. Bolt, and C. Blundell. Never give up: Learning directed exploration strategies, 2020.

[4] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA, 2009. Association for Computing Machinery.

[5] Y. Burda, H. Edwards, A. Storkey, and O. Klimov. Exploration by random network distillation, 2018.

[6] A. Dosovitskiy, P. Fischery, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 2758–2766, USA, 2015. IEEE Computer Society.

[7] Z.-W. Hong, T.-J. Fu, T.-Y. Shann, Y.-H. Chang, and C.-Y. Lee. Adversarial active exploration for inverse dynamics model learning, 2020.

[8] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. D. Turck, and P. Abbeel. Vime: Variational information maximizing exploration, 2017.

[9] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks, 2016.

[10] C. Linke, N. M. Ady, M. White, T. Degris, and A. White. Adapting behaviour via intrinsic reward: A survey and empirical study, 2020.

[11] H.-K. Yang, P.-H. Chiang, M.-F. Hong, and C.-Y. Lee. Flow-based intrinsic curiosity module. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, Jul 2020.