A Research on Reward Setting and Curiosity Encoding Methods

Da Yang

School of International Education, Nanjing Institute of Technology, Jiangning, 211167, China
x00202201234@njit.edu.cn

Abstract. Agents in reinforcement learning relies on reward to make movement, improve algorithms, and reach the final goal. However, reward setting is a subject that requires much engineering skills and experiences. Two types of reward, extrinsic reward and intrinsic reward, are totally different in ways of setting. A typical type of intrinsic reward is curiosity. Although there have been many studies on curiosity reward mechanisms in algorithms, the comparison and analysis of different methods are not comprehensive enough. The paper: (a) made detailed introduction to general types of extrinsic reward setting methods and their applications. (b) investigated the encoding methods for curiosity intrinsic reward and make comparisons among various derivations of different type of encoding methods. (c) demonstrated the agent performance implement different encoding methods and prove that encoding method has great influence on the reward setting of curiosity. Finally, the paper summarizes and looks forward to the full text.

Keywords: Reward setting; curiosity; reinforcement learning; encoding method

1 Introduction

Reinforcement learning (RL) relies on rewards given after action by agent in the environment to improve. Each action taken by the agent aims at maximizing the reward. This requires the rewards in environment to be dense and align to the task well. However, under most circumstances, it requires much effort to annotate the environment with extrinsic rewards which are hand-designed to be dense. But though achievable, this way is not recommended as it is not scalable.

To make the learning progress more autonomous and reduce dependence on extrinsic rewards, intrinsic rewards are introduced to join in. In this way, the notorious challenging engineering problem of designing a well-shaped reward function is filled with dense intrinsic rewards. In this background, a mutual topic for both extrinsic and intrinsic rewards to pay attention to is how to set them properly in order to achieve the ultimate goal.

In comparison, the process for setting extrinsic reward is relatively easier. For example, in most game scenarios, beat the opponent, arrive at the designated location,

P. Kar et al. (eds.), *Proceedings of the 2023 International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2023)*, Advances in Computer Science Research 108,
https://doi.org/10.2991/978-94-6463-300-9_63

get powerful power-ups can lead to high rewards. Instead, miss the target, loss of health, or even be killed can lead to bad punishments. For intrinsic rewards, the progress can be much harder. For example, curiosity as a frequently used intrinsic reward which uses prediction error as reward signal, has to define what features to be included in the calculation of prediction error. In general, Pixels, Random Features (RF), Variational Autoencoders (VAE), Inverse Dynamics Features (IDF) are some commonly used features in calculation. Although there are some papers on how to set extrinsic rewards on the market, there are very few explanatory papers on the setting and comparison of intrinsic rewards. Therefore, this paper focuses most on a typical type of intrinsic reward--curiosity.

This paper presents the fundamental knowledge about curiosity and some of its derivations. For each derivation, there are discussion on the advantages and disadvantages, moreover, the scenes it can provide high performance.

2 Reward setting and curiosity encoding methods

First the paper takes a look at the reward setting of extrinsic rewards, for most scenarios in the market implementing deep reinforcement learning, it is aligned with the final goal. However, the ever-changing environments along with the proceeding of tasks can bring certain difficulties.

For example, in hard-exploration scenarios, Extrinsic rewards can be sparse and deceptive. Define a good reward function requires good understanding of the goal and the skill to design algorithms. A typical straight forward function, get reward after reaching the goal, can be easy for design but lead to potential sparse reward condition. It is feasible to apply the function to some simple scenarios. But when it comes to the scenarios where it takes the agent many steps to reach the goal, the performance of agent would be low. A common solution could be implementing a denser reward function. However, if not moderated properly, it can lead the robot drop into a dead end, reach local optima, or even worse, causing safe problems [1].

Intrinsic rewards were first introduced as aids to extrinsic rewards, but with the development of the field, it is possible now to use intrinsic rewards only to accomplish certain tasks. The core of intrinsic reward settings is reward function, and at the heart of the reward function is the way and parameters used to do the calculation.

Curiosity, an advanced method, is general for augmenting intrinsic reward in to the environment to make it denser. In the calculation of prediction error, two aspects are involved. One is reward function, which is always set manually by experts to coincide with the target. Another is state encoding, which plays a vital part on the agent's performance [2]. A good feature space generated should be compact, sufficient, and stable [3]. In the market there are many state-of-the-art state encoding methods. The paper mainly introduced four categories and their derivations, Pixels, Random Features (RF), Inverse Dynamics Features (IDF), Online Variational Autoencoder (VAE). The encoding methods above are classic and mostly used in different scenarios but only a small sample of possible encoding methods. Other method, for

example, reference [4] introduces a method that focus on common features at large and distinct features specifically, which also appears to be very interesting and worth further discovery. Typical ways of giving curiosity rewards involve entering either novel [5] or surprising states [6]. The paper focus on the latter, i.e., surprising states. The reward function, also called the surprisal, is defined as mean-squared error corresponding to a fixed-variance Gaussian density [3].

2.1 Pixel-based methods

Pixel-based method trains agent to learn features from pixels directly in the observation space with no extra feature learning component. As a result, Pixels are stable. From another perspective, as Pixels contain all information, the feature space is sufficient for exploring relevant aspect of the environments. Fit the model in the observation space, using Pixels for training, is the simplest case. While it has been popular for early stages of exploring tasks, it is no more suitable for current ones with high-dimensional and complex observation space. In these scenarios, learn directly from raw pixels appears to be very challenging and often generate bad results.

For now, some measures have been proposed to improve the performance of the pixel-based method, such as encoding pixels in to features or encoding invariances from raw pixels. However, not all features contained in pixels are relevant with the task. Therefore, embedding specific feature learning method according to various task is a more popular approach. As pixels contains all the information, when implementing it into curiosity, the value can be precise. However, it requires more computation power in the resolving progress and does not coincide with the compact standard.

The BASS method (Basic Abstraction of the ScreenShots), derived from Naddaf's BASS (2010), concentrates on encoding colors embedded in the environment [7]. Instead of using pixels directly, BASS distinguishes features by combining a group of pixels [8]. By subtracting the background as the first pre-procession, BASS is then able to encode the group of pixels (presented as SECAM palette colors) at a low resolution (Fig.1). In this way, features are stable, compact, and sufficient for early-stage game scenarios where the environment is relatively simple.

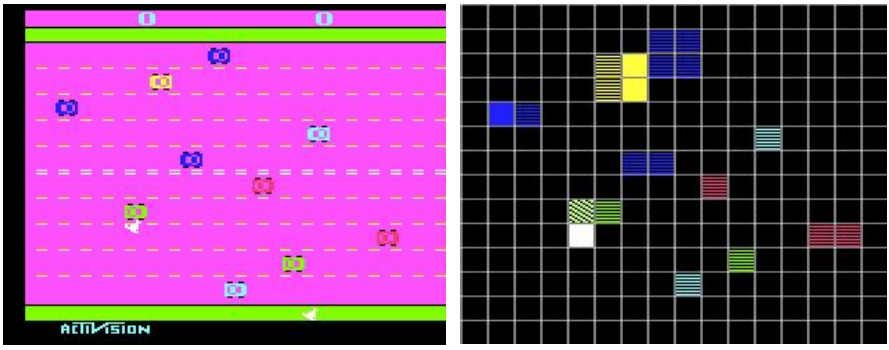

Fig. 1. Left: original colors. Right: encoded colors using BASS [9].

However, the display capability of hardware has greatly improved, and designs of the game scene are way more complicated. Some environments are highly dynamic which leads the BASS to be unstable. Furthermore, aside from pixels, there are textures, light, shadow, and particle effects all combine together to present good visuals. Just extracting the environment and grouping pixels together can also be a challenging task. The idea of Basic method is to detect the practical effects the additional features bring in BASS method. It omits the pairwise combinations while encoding the presence of the 128 colour to present colour more accurately. However, it uses identical features (Fig.2).

The pre-procession part of DISCO is basically the same as in Basic and BASS. During the actual training, DISCO presents a novel approach to infer the category label of observed objects and encodes their location and speed by tile coding method [10]. This helps the method better understand the movement in high-dimension by achieving function approximation.

Fig. 2. Left: Seaquest game scenario. Right: different objects observed by DISCO [9].

The LSH method, taking 2600 Atari game screens as dataset, achieves to map the dataset into several binary features [10]. During the mapping process, random projections are used to maintain the disparity from high dimensions to low ones. In this way, the features generated from more similar environments would overlap to a large extent, keeping the feature space stable. Completely different from the other four method, the RAM utilizes the memory as observation space. It generates a binary feature for every bit and records the logical relationship for each combination of two bits. It is achievable for that Atari was consisted of 1024 random access memory bits. The feature space could be compact. But as the game memory grows larger and contains more dimension, it would be very hard to implement the RAM method.

2.2 Random Features (RF)

RF was first introduced to deal with classification problems. A substantial literature of the random features method is on random projections and more generally randomly initialized neural networks. It has been proved whilst random features can fulfill the simpler classification tasks, feature learning outperforms random features once the task becomes complex enough. As state encoding method also focus on disparity

between different observation spaces, the random features pattern can be also applied [11]. Using RF as the state encoding method for reward function to set curiosity value is a popular and good choice when it comes to easy exploration tasks.

The random features method in state encoding utilizes embedding network to accomplish encoding, mostly convolutional network. The network is first randomly initialized, and then fixed. As for the reason, the feature space is very stable. Depending on the number of layers embedded in the network, the feature space could be either compact or not. It is believed that random features may not be sufficient for specific tasks, however, it appears to be a "surprisingly strong baseline" [3].

2.3 Variational Autoencoders (VAE)

Variational autoencoders, first inspired by the Helmholtz Machine [12], serves as a principled framework for deep latent-variable models learning process and inference models correspondence [13]. The resulting models of VAE are usually highly intuitive and interpretable. Thus, it provides compactness. Moreover, the generative process of data naturally expresses casual relations of the observation space, which holds priority to better generalize new situations, i.e., exploration tasks. The VAE can be viewed as a mixture consisting of two models, the recognition model(encoder), and the generative model(decoder). By working collaboratively, the VAE achieves to model the relation between input and latent variables using a set of parameters. In this way, the feature space can be made low-dimensional but at the same time sufficient enough for prediction. However, as the two parts would improve in the training progress, the method is not stable and may still contain some irrelevant details, i.e., noises. A problem that exists when applying VAE to reward function is that neither a fixed VAE nor an online VAE can fulfill the requirements of a good feature space. A fixed VAE can be gradually out of date during the process of moving from initial observation space to another one, while an online VAE introduces low stability to the function [2]. To provide better performance than Random Features method in more complex scenarios, some experts come up with the idea to embed innate knowledge into original VAE method. Some of the self-exclusion knowledge about the core areas of early development seems to be innate to humans [14]. Implementing the idea, a new derivation of VAE method emerged.

Fixed β-VAE Encoding, introduced by Lehuger and Crosby, focus on better performance rivaling with other state-of-the-art methods when it comes to sparse reward 3D environments. A 4-layer convolutional network is chosen for the VAE encoding architecture to balance the compactness and sufficiency. By encoding relevantly identical graphs for the overall performance, curiosity value is well set adjusting to the task performance. Ideally, the method could maintain compactness, sufficiency, and stability at a relatively high level at the same time. The percentage of reaching the final goal is in Table 1. The data illustrates that VAE agent achieves higher performance than the other two benchmark methods after same number of steps. It proves that the Fixed β-VAE Encoding method can solve complex tasks whilst maintaining the ability to accomplish easier ones.

Table 1. Performance score for different scenarios with standard deviation.

Methods	Score of Training	Score of Test
Variational Autoencoders	93.6% ± 0.046	46.2% ± 0.088
Inverse Dynamics Features	87.1% ± 0.113	38.2% ± 0.06
Random Features	55.9% ± 0.334	29.1% ± 0.178

3 Performance Comparison of Algorithms

The paper first took a look at the performances of 5 derivations of pixel-based algorithm mentioned in the paper. Games for the experiment was chosen from the 123 games listed on Wikipedia. They all contain an individual mode for player, are not fully-explored or prototypes, and can be performed emulation in ALE. Five games altogether consist the set for training: ASTERIX, BEAM RIDER, FREEWAY, SEAQUEST and SPACE INVADERS. Then researched methods were evaluated on a sample size of 50 from the testing set. On each game, algorithms were executed for 10 episodes. For evaluation, to generate more compact summary statistics, score distribution aggregate and paired tests were utilized to help generating the comparison of performance across various domains. The score distribution, unlike the average and median scores, represents accurate scores based on the agent's performance ignoring the distribution of individual score Baseline and inter-algorithm score distributions are showed in Fig.3. For paired tests, A two-tailed Welsh's t-test with 99% confidence intervals was performed in order to distinguish the significant difference between scores of various algorithms. Table 2 presented the numbers of scenarios where an algorithm outperforms the other by an obvious degree [8].

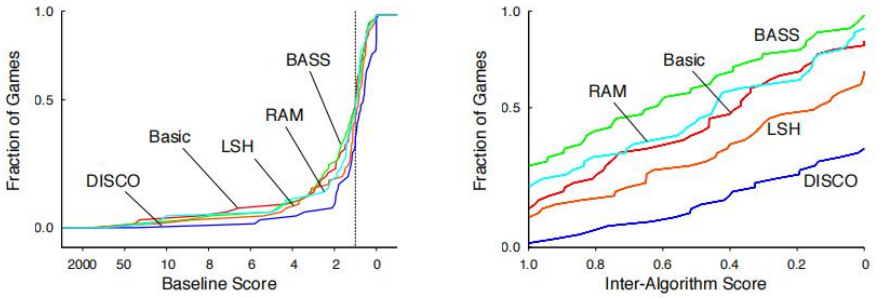

Fig. 3. Score distribution over all games [9].

Table 2. Paired tests over all games. For each comparison, if left algorithm outperforms the right algorithm, then count as one.

	Basic	BASS	DISCO	LSH	RAM
Basic	—	18-32	39-13	34-18	22-25
BASS	32-18	—	48-5	36-17	29-20
DISCO	13-39	5-48	—	17-33	9-41
LSH	18-34	17-36	33-17	—	15-36
RAM	25-22	20-29	41-9	36-15	—

Then the paper focus on the disparity among Fixed β-VAE, Online-VAE, IDF, RF, and Pixel methods. Breakout, a very commonly used benchmark, was used to generate a representative dataset before the experimental steps. A four-layer convolutional network was embedded into the VAE encoding architecture concerning compactness and sufficiency at the same time. The experiment used a previous dataset of observations to create the VAE encoding [2]. Moreover, Experimental results also included the almost identical replication of Burda et al, providing a strong baseline for comparison. The results are shown in Fig.4, establishing the comparison of performance among five different types of agents in environments.

From the experimental results [2,8], the conclusion was generated. The derivations of encoding methods have been strongly proved to be useful when solving specific problems. By resolving pixels, it can make the feature space more compact, therefore, improving the performance of curiosity-driven agents in hard exploration tasks. For VAE, the derivations can get higher scorer by improving the stability. At the same time, the structure of VAES has been proven to be changeable according to the curious agents.

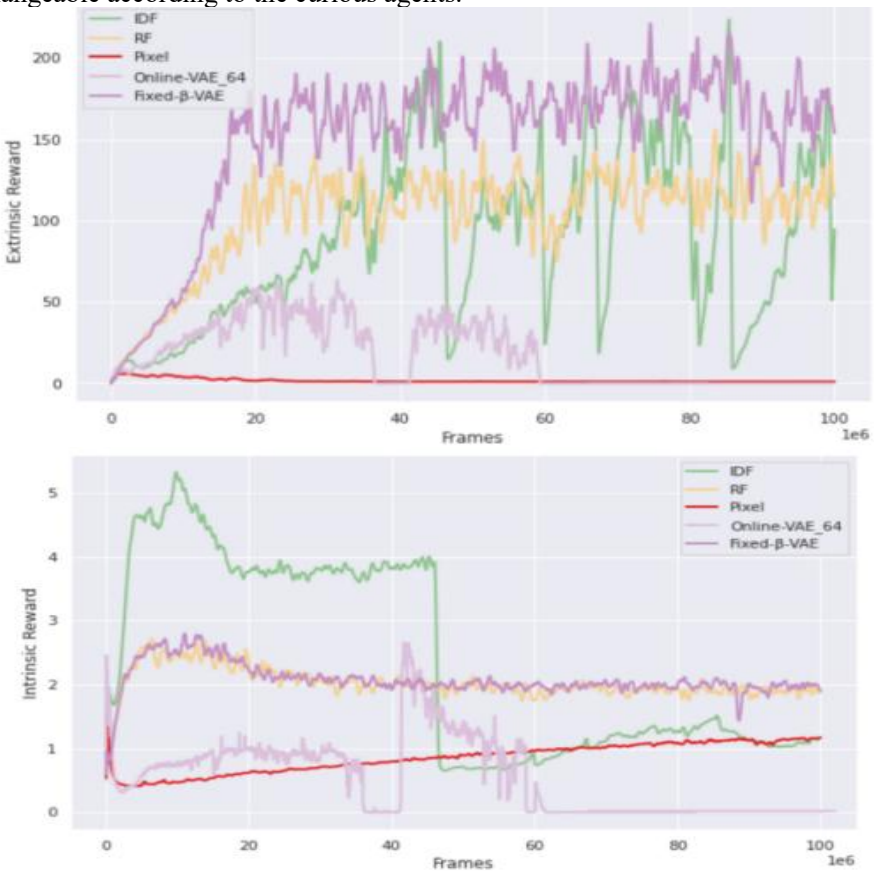

Fig. 4. Extrinsic (Up) and Intrinsic (Down) reward on Atari Breakout [16].

4 Conclusion

The paper presented different derivations of curiosity encoding methods and their specific advantages. Comparison between different derivations were made to test the performance of agents with different encoding methods. For each encoding method type and their derivations, the indicators were talked through in detail. The paper analyzed the availability to use raw pixel as data source and resolve it to lower dimension to achieve higher performance. For several derivations of the pixel-based encoding method, they have successfully improved the performance of agents to some extent. The paper confirmed the flexibility of VAE encoding method about its ability to change according to the curious agents. In most cases, RF outperformed the other methods, showing that using RF are very sufficient.

Though the advantages and disadvantages of various derivations and their influence on the curiosity reward setting have been researched thoroughly, it still remains questions what is the best encoding methods for specific tasks and the extent curiosity rewarding setting affected by them. For further research, a more open-ended environments is needed to make training, testing, and comparison of derivations. Then, the solution of encoding method concerning specific tasks can be found. However, to break down the influence of encoding methods on curiosity reward setting, it requires much more work to be done. it is an interesting and unprecedented period which can promote the learning progress of reward setting.

References

1. Ecoffet, A., Huizinga, J., Lehman, J., Stanley, K. O., & Clune, J. (2021). First return, then explore. *Nature*, *590*(7847), 580-586.
2. Lehuger, A., & Crosby, M. (2021). Fixed $\beta $-VAE Encoding for Curious Exploration in Complex 3D Environments. *arXiv preprint arXiv:2105.08568*.
3. Burda, Y., Edwards, H., Pathak, D., Storkey, A., Darrell, T., & Efros, A. A. (2018). Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*.
4. Anand, A., Racah, E., Ozair, S., Bengio, Y., Côté, M. A., & Hjelm, R. D. (2019). Unsupervised state representation learning in atari. *Advances in neural information processing systems*, *32*.
5. Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., & Munos, R. (2016). Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, *29*.
6. Achiam, J., & Sastry, S. (2017). Surprise-based intrinsic motivation for deep reinforcement learning. *arXiv preprint arXiv:1703.01732*.
7. Naddaf, Y. (2010). Game-independent ai agents for playing atari 2600 console games.
8. Bellemare, M. G., Naddaf, Y., Veness, J., & Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, *47*, 253-279.
9. Marc G. Bellemare, Yavar Naddaf, Joel Veness, Michael Bowling, (2013) The Arcade Learning Environment: An Evaluation Platform for General Agents, *Journal of Artificial Intelligence Research 47*, 2013, 264.
10. Sutton, R. S., & Barto, A. G. (1998). *Introduction to reinforcement learning* (Vol. 135, pp. 223-260). Cambridge: MIT press.

11. Gionis, A., Indyk, P., & Motwani, R. (1999, September). Similarity search in high dimensions via hashing. In *Vldb* 99, 6, pp. 518-529.
12. Jarrett, K., Kvukcuoglu, K., Gregor, K., & LeCun, Y. (2016). What is the Best Feature Learning Procedure in Hierarchical Recognition Architectures?. *arXiv preprint arXiv:1606.01535*.
13. Dayan, A., & Thomas, J. R. (1995). Development of automatic and effortful processes in memory for spatial location of movement. *Human Performance*, *8*(1), 51-66.
14. Kingma, D. P., & Welling, M. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, *12*(4), 307-392.
15. Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and brain sciences*, *40*, e253.
16. Auguste Lehuger, Matthew Crosby, Fixed β-VAE Encoding for Curious Exploration in Sparse Reward 3D Environments *arXiv preprint arXiv:2105.08568, 2021), 8. pag.*