When should agents explore?

Zabolotnyi Artem [1]

Abstract

Reinforcement learning nowadays is a very important direction of machine learning and deep learning. The core idea is to use an agent which operates in the environment and has the main goal to maximize some reward. Training such an algorithm could be difficult due to the huge amount of possible actions. An agent should explore a variety of actions and at the same time choose more relevant ones. Trade between such actions is an exploration vs exploitation problem, which is one of the most important in RL. In this project, we focus on applying different strategies of choosing when to explore and when to exploit. The main goal of this project is to compare different strategies of exploration and exploitation and compare them using Atari games.

1. Problem statement

Reinforcement learning helps to solve problems when we have a complex environment and could not collect and label all possible samples from it. We can put an agent into this environment and define the reward function without any explanation of how to optimize it. Agent must discover the environment and actions to maximize reward not only for the current action but for the further situation and all possibles of action sequences. The agent will understand the environment after a big amount of different trials some of them could be errors. Agent develops some strategy for what to do in a different situation. Based on this experienced agent could find an optimal path through all possible actions to maximize reward.

One of the core challenges of reinforcement learning is to make the algorithm flexible in terms of trade between exploration and exploitation. To get maximum reward must use actions that it tried before and has a positive influence. But sometimes there are actions that agents do not take into account in the past but using them potentially get better rewards than others. Exploitation - is an action choosing using previous experience, when exploration choose probably not the best action at this time which probably lead to a better global solution. It is a very hard and still not solved problem of an optimal algorithm of the proper way of combining exploration and exploitation.

[*]Equal contribution [1]Skolkovo Institute of Science and Technology, Moscow, Russia. Correspondence to: Zabolotnyi Artem <zaaabik@gmail.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

2. Strategies of exploration and exploitation

The main goal of RL is to maximize reward. To obtain good results algorithm should use diverse experiences to explore the different situations and sometimes use policy which optimizes reward. Obtaining the best algorithm could be decomposed into two parts: granularity and switching mechanism.

2.1. Granularity

Switching between modes could be done in several ways: **Step-level** - In each step of agent decides to follow policy or does not optimal step. The canonical example of such a method is ϵ-greedy. The core idea is to move within policy but with the small probability, ϵ makes different actions.

Experiment-level The extreme case when behaviour during all process of training agent is exploration without any policy. Learned policy using only for evaluation.

Episode-level exploration is using the one strategy during the whole episode (training games versus tournament matches in a sport).

Intra-episodic method falls between step and episode. Exploration should contain several steps but should not be during the whole episode.

2.2. Switching methods

The way to decide when to start to explore and when to stop could be done in several ways:

Blind switching - the easiest method of switching does not use any information about the current state and actions are done. Could be implemented with a counter (every 100 steps start to explore with 10 steps) or probabilistic when we have a probability of starting exploration mode for the

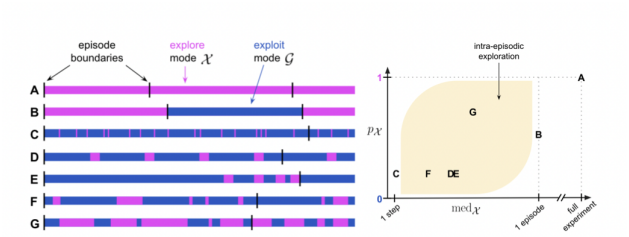

Figure 1. Illustration of different types of temporal structure for two-mode exploration. Left: Each line A-G depicts an excerpt of an experiment (black lines show episode boundaries, experiment continues on the right), with colour denoting the active mode (blue is exploit, magenta is explore). A is of experiment-level granularity, B episode-level, C step-level, and D-G are of intra-episodic exploration granularity. Right: The same examples, mapped onto a characteristic plot of summary statistics: overall exploratory proportion pX versus typical length of an exploratory period medX . The yellow-shaded area highlights the intra-episodic part of space studied in this paper (some points are not realisable, e.g., when $pX \approx 1$ then medX must be large). C, D, E, F share the same $pX \approx 0.2$, while interleaving exploration modes in different ways. D and E share the same medX value, and differ only on whether exploration periods are spread out, or happen toward the end of episode.

parameterized amount of steps.

Informed switching next method use the information of the agent to switch strategies. Agent return two scalars, the first is *trigger* could be interpreted as uncertainty. When the trigger value is high, then the agent will switch to explore mode. One of the ways to measure such value is 'value promise discrepancy'.

$$D_{promise}(t-k,t) := \|V(s_{t-k}) - \sum_{i=0}^{k-1} \gamma^i R_{t-i} - \gamma^k V(s_t)\| \tag{1}$$

where V(s) estimation of value at sate s, R - reward value and γ discount factor.

2.3. Combining approaches

To improve the exploration process and prevent hyperparameter tuning author propose to add modification.

Bandit adaptation makes the algorithm more flexible using two parameters, the first how often we can enter into exploration mode and the second how quickly exit from it. Such parameters could be duration, probability or target rate. Also, this optimization could be done by a meta-controller which maximizes episodic return.

Homeostasis Value range of $D_promise$ could change over a time, due to training which improve accuracy. If we use the same threshold for all time it will be not optimal. For this reason adaptive threshold algorithm used. Core idea is to track signal for switching during the training and adopt it to a specific average *target rate*. Such trick helps to make signal independent of scale of a trigger signal.

2.4. Intra-episodic exploration

Such technique of combination methods discussed in 2. There are several thing we can change:

- Explore mode
- Explore duration
- Blind or informed trigger
- Exploit duration

Combining different choices in each category we obtain new algorithm. We will compare performance between each other and several baselines.

3. Baselines

For this project we take 3 baselines which often used. Such baselines are simple and efficient for different tasks.

- Pure explore mode ($p_x = 1 = \epsilon$)
- Pure exploit mode ($p_x = 0 = \epsilon$)
- $\epsilon - greedy$ strategy with ($p_x = 0.01 = \epsilon$)

,where p_x probability of exploration action in each step.

4. Related works

One of the most important $\epsilon - greedy$ algorithm adopted for creating chunks of exploration actions where length sampled from distribution with heavy tails ((Dabney et al., 2020)). The paper (Bagot et al., 2020) propose intrinsic reward pursuit invoked by the agent. Authors propose to learn such reward function thought exploration options, i.e additional temporally-extended actions to call separate polices.

5. Data

For our experiments we use Atari Learning Environment (Bellemare et al., 2013) - one of the biggest benchmark for the study exploration. Benchmark contains of big amount of old atari games adapted for RL agents agents.

References

Bagot, L., Mets, K., and Latré, S. Learning intrinsically motivated options to stimulate policy exploration. 2020.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Dabney, W., Ostrovski, G., and Barreto, A. Temporally-extended ϵ-greedy exploration. *arXiv preprint arXiv:2006.01782*, 2020.