1 Region-Based Goal Distributions for HER

1.1 Motivation

In research related to HER, a lot of work has been spent on parameter tuning, different off-policy algorithms and prioritization within the replay buffer (Energy-based, entropy-based, distribution-based, filter-like).

However, one very crucial point has not been tackled at all. While it has been shown that changing the distribution of virtual goals when sampling from the replay buffer improves performance in the majority of the environments, no one has analyzed the importance of the initial distribution of goals.

FetchPush

For example, consider the benchmark environment FetchPush: (note that we are talking about sampling goals, not virtual goals from the replay buffer)

Target goals (final target positions of the object to be pushed) are sampled from a uniform distribution over the set $G_T = \{(x, y, 0)\}$ with $\sqrt{(x - x_0)^2 + (y - y_0)^2} < r$. (x_0, y_0) denote the initial gripper position, r denotes the maximum goal distance from the initial gripper position.

Initial positions of the objects are sampled from a uniform distribution over the set $S_0 = (x, y, 0)$ with $a < \sqrt{(x - x_0)^2 + (y - y0)^2} < b$. (x_0, y_0) denote the initial gripper position, a and b with $0 < a < b < r$ denote the minimum and maximum initial object distance from the initial gripper position . One clearly observes that $S_0 \subset G_T$ and therefore $S_0 \cap G_T \neq \{\}$.

In such a scenario, where S_0 and G_S overlap, sampling from a uniform goal distribution makes sense, because

- When the agent is exploring randomly in the beginning, starting from the initial object position, (virtual) goals that are within or close to S_0 are reached more easily
- Since S_0 is (at least partly) contained in G_T, the agent will sooner or later receive positive rewards which are necessary for learning.
- Due to generalization capabilities of the neural network, the agent can generalize from past experience to goals in the surrounding of the ones it has already managed to reach (virtually or non-virtually).
- Since goals are uniformly sampled from G_T, the agent will constantly encounter goal in the surrounding of the ones it already know. Therefore, learning works well.

FetchPickAndThrow

Now consider a different environment FetchPickAndThrow: Here, Fetch must pick up an object from a box and throw it towards a goal located on the floor outside of the box.

For this environment, the initial state space S_0 denotes the set of initial positions of the object to throw. S_0 contains states in which the object is inside the box.

The target goal space G_T denotes the set of target goals the agent should learn to reach. Since we are only interested in successful throwing, G_T contains all states on the floor outside of the box.

This environment has two fundamental differences:

- Initial state space and target goal space do not overlap: $S_0 \cap G_T = \{\}$
- Initial state space and target goal space are (potentially) far from one another: There is a set of states that are neither contained in S_0 nor in G_T but that have to be traversed in order to move the object from S_0 to G_T.

In such a case, sampling goals uniformly from G_T does not lead to learning success:

- Random experience is generated for a goal sampled from G_T.
- Since G_T is far from S_0, the agent does not encounter positive rewards.
- The experience is replayed with a virtual goal that is close to S_0 since these are the states that are reached first.
- The agent learns how to reach this virtual goal (that is close to S_0 and still far from G_T.
- Again, a goal is sampled from G_t: However, since this goal is far from the virtual goals that are located near S_0, the neural network is unable to generalize. Random experience is generated and the agent does again not receive a positive reward.
- This process repeats itself, the agent never encounters a real positive reward.

The bottom line is that virtual goals and the corresponding virtual rewards are useless unless they are close to real goals. Neural networks can only generalize to fairly similar situations. In the above scenario, since the agent never encounters a real positive reward, learning is not guided from goals within S_0 to goals within G_T. False-positive virtual rewards and a state space to large to explore randomly lead to the agent being stuck close to S_0.

1.2 Region-Based Goal Distribution

HER successfully learns in FetchReach, FetchPush and FetchSlide environments. In all of these environments, $S_0 \cap G_T \neq \{\}$ and a uniform goal distribution is used. However, HER fails in environments FetchCurling, FetchPickAndThrow, and FetchPickAndPlace (to some extent, explained below). In these environments,$S_0 \cap G_T = \{\}$, and a uniform goal distribution fails.

In order to enable HER to learn these challenging environments, we introduce the concept of Region-Based Goal Distributions:

1.2.1 Definition: Region-Based Goal Distributions

Take the following sets as given:

- Overall state space $S = R^3$: contains all possible states of the object)
- Overall goal space $G = S = R^3$: all possible states are possible goals to be reached
- Initial state space $S_0 \subset S$: set of possible initial states of the object
- Target goal space $G_T \subset G$: set of goals the agent should learn to reach, i.e. the set of goals we are interested in)

For convenience, define the initial goal space $G_0 = S_0$ for goals lying within the initial state space.

Consider a case, in which $G_0 \cap G_T = \{\}$ (e.g. FetchPickAndThrow):

For the given environment, define N disjoint intermediate goal spaces $G_1, G_2, ..., G_N$ where

- $G_i \cap G_j = \{\} \, \forall \, i,j \in \{0, ..., N\}$
- $G_i \cap G_T = \{\} \, \forall \, i,j \in \{0, ..., N\}$

The sample goal space G_S is then given by the union of the intermediate goal spaces, the initial goal space and the target goal space:

$$G_S = G_0 \cup G_1 \cup ... \cup G_N \cup G_T$$

The intermediate goal spaces (regions) should be defined such that G_S contains states that are neither contained in S_0 nor in G_T but that have to be traversed in order to move the object from S_0 to G_T.

We now define a random variable $X = \{(X_1, X_2, X_3)^T\}$, which takes the value of a goal sampled from G. The corresponding Probability Density Function (PDF) $f_X(x)$ is defined such that $f_X(x) = 0 \forall x \notin G_S$.

Therefore, as per definition of a PDF:

$$Pr[X \in G_S] = \int_{G_S} f_X(x)dx = \int_G f_X(x)dx = 1$$

The PDF $f_X(x)$ is designed to be piecewise uniform on G_i:

$$f_X(x) = c_i \, if \; x \in G_i \, \forall \, i \in \{0, 1, ..., N, T\}, c_i \in R^+$$

The probability p_i that X takes a value within G_i can be obtained as follows:

$$p_i = Pr[X \in G_i] = \int_{G_i} f_X(x)dx = c_i \int_{G_i} dx$$

1.2.2 Short summary: Region-Based Goal Distribution

- We are given an environment with an initial state region $S_0 = G_0$ and a target goal region G_T.
- For this environment, we define N suitable intermediate regions G_i
- To each region G_i, we assign a probability p_i for sampling from that region G_i
- If we sample from a region G_i, we sample uniformly from it

1.2.3 Example Environments

Region-Based Goal Distributions can be applied to environments such as:

FetchCurling

FetchCurling is an extension of FetchSlide. The table length (x-axis) has been increased by a factor of 3. We divide the table in 6 equally large areas along the x-axis. These agent is supposed to slide the object towards goals located in the sixth area (the one that is the furthest from the initial gripper position). Therefore G_T is the set of states within the sixth area. The initial state space $S_0 = G_0$ is defined as the set of states within the first area. We can now apply Region-Based Goal Distributions with regions G_i corresponding to the $i + 1$ th area of the table.

FetchPickAndPlace

At first, the researchers in the HER paper wanted the agent to learn reach goals within G_T, defined as the set of states on the table. However, the agent would not learn to pick the object up and place it at the desired goal. Therefore, the researchers chose to introduce goals that are located in the air above the table (G_1). In their benchmark implementation, the probability of in-the-air goals being sampled is $p_1 = 0.5$. Through this trick, the agent was able to learn picking up and placing the object in their originally desired goals from G_T. As you can see, without further explaining this crucial trick, FetchPickAndPlace has already implemented Region-Based Goal Distribution with one intermediate region G_1. It will be interesting to see the effects of changing p_i, or introducing Dynamic Region-Based Goal Distributions (see below).

FetchPickAndThrow

For FetchPickAndThrow, G_0 and G_T have already been defined above. Similarly to FetchPickAndPlace, one intermediate region G_1 could be defined as the set of states in the area above the box (reachable by the gripper). Probably, the agent will learn how to pick up objects, which is crucial in order to throw them out of the box.

1.2.4 Extension: Dynamic Region-Based Goal Distribution

As an extension to Region-Based Goal Distributions, changing p_i within the course of learning could lead to improved learning performance.

Consider the FetchCurling Environment: Intuitively, in the beginning of the learning process, it makes sense to assign larger values to p_0 and p_1. Goals in G_0 and G_1 are more likely to be achieved in the beginning and help the agent to learn faster. However, when the agent has mastered goals in G_0 and G_1 (e.g. at a success rate of 80% for these goals), it makes sense to decrease p_0 and p_1 while gradually increasing p_2, p_3 and p_T. The agent encounters more "challenging", further-away goals which it is now able to learn since he has mastered the first episodes.

This strategy is in analogy to human learning behavior. Excessive demands (goals within G_T) at an early stage will not help learning. Too low demands (goals within G_0) at an advanced stage will also not help learning. The difficulty of goals should therefore be adapted to the learning progress.

Dynamic Region-Based Goal Distributions work in this exact way: Learning success is measured individually each goal state G_i. Sampling probabilities p_i are adapted whenever a certain threshold in learning success, e.g. $Success(G_i) > 0.8$ is met. It requires experimentation to come up with a suitable (and general) dynamic strategy.

1.2.5 Research Questions: Region-Based Goal Distribution

The following research questions have to be addressed for each environment individually. The overall goal is to maximize sample efficiency and final success rate by choosing suitable G_i and p_i. Generalizations might be possible after evaluating experiments for different environments. However, a strategy on how to maximize learning progress will be given based on experiments.

- How do we define suitable intermediate regions G_i (number and size of regions)?
- Which probabilities p_i lead to good sample-efficiency / final success rate?
- Can we achieve better sample-efficiency / final success rate with by dynamically changing p_i during the learning process?
- How can prioritization of the replay buffer (Energy-Based, Entropy-Based, Distribution-based, Filter-like) help to achieve even better results?