# Hindsight Experience Replay*

Andrychowicz et al.

2018

## 1 What

Hindsight Experience Replay (HER), a technique which allows training an RL algorithm in an environment with sparse and binary rewards. You can combine it with any off-policy algorithm, the authors demonstrate the results on DQN [Mnih et al., 2015] and DDPG [Lillicrap et al., 2015].

## 2 Why

RL is on the rise. However, it's tremendously hard to come up with a proper reward function. Sometimes we don't even know what the reward signal should be like. On the other hand, we, humans, can analyse the outcome of our behaviour and reason an extent to which extent we have achieved the desired result.

The authors come up with the following analogy. We play hockey and hit the puck. Unfortunately, we miss the goal. RL can't learn a lot from this experience apart from the fact, that we've been unsuccessful. Humans, at the same time, can reason in the following way: "If the net had been a bit right, we would have been scored."

## 3 How

Here we have a usual DQN-like setup with the experience replay. The difference is in the policy/value input; we concatenate state with the goal.

Since changing the reward function does not influence the dynamics of the environment, we're free to recalculate the reward for the transition in the rollout and put it to the memory as well. The authors call this reward recalculation a 'replay' if I'm not missing anything.

Amount of additional goals and the way we choose them is defined by a sampling strategy. The authors show that the best one is to use $k$ random states which come from the same episode and come after the transition in question as goals.

*Notes by Vitaly Kurin `https://yobibyte.github.io/`

# 4 Evaluation

On the bit-flipping environment, where you need to flip the bits of a vector so that it equals to some target vector, HER destroys DQN. Would be interesting to see how UFVA [Schaul et al., 2015] behaves for this one since they mention and get inspiration from it.

I really liked that one: "There are no standard environments for multi-goal RL and therefore we created our own environments [1]". However, people have tried multi-goal RL setup for Atari. But! As I will say in the Comments section, this approach only works for goal-directed behaviour.

I really like the experimental part, where the authors try to answer the following questions:

- Does HER improve performance?
- Does HER improve performance even if there is only one goal we care about?
- How does HER interact with reward shaping? (this one I don't like actually, they show that some particular reward doesn't work for both methods. So? Not clear what should we take out of that.)
- How many goals should we replay each trajectory with and how to choose them?

We can see, that DDPG+HER destroys the baseline DDPG and DDPG + count based exploration on the three robotics tasks as well. However, it would be interesting to see how other SOA algorithms work for this benchmarks (e.g. PPO).

The last evaluation task is really cool. They pretrain the policy in the simulation and use it for real robot without pretraining!!! They also add a disclaimer saying that we need to add noise to observation and it will work out of the box.

# 5 Comments

- The authors somehow do not mention that the method should work only with learning the goal-directed behaviour. This is kinda limiting. E.g. even in such simple environment as *Hopper* there is no goal, you just need to learn how to jump.
- I like the analogy with the puck form the introduction, at the same time, I don't think this is how human usually think. I would reason "if I had shot a bit left, I would have scored". Moreover, providing a goal looks like a simplified version of reward shaping to me. The argument "it is not applicable in the situations when we don't know what admissible behaviour

[1] With blackjack?

may look like" from the intro does not work then. Or do we know the goal, but do not know the admissible behaviour? Looks weird to me.

- "In order to prevent saturation and vanishing gradients we add the square of their preactivations to the actor's cost function". Would be interesting to see whether everything fails if you remove that or not.
- Is feeding the relative target position makes it something like implicit reward shaping? Not sure here, but would be also interesting to see what happens when we feed the static global target position here.
- "It confirms that the most valuable goals for replay are the onese which are going to be achieved in the near future". Another indicator of implicit reward shaping? Maybe I'm just missing something. =(

# References


[Lillicrap et al., 2015] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

[Mnih et al., 2015] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529.

[Schaul et al., 2015] Schaul, T., Horgan, D., Gregor, K., and Silver, D. (2015). Universal value function approximators. In *International Conference on Machine Learning*, pages 1312–1320.