Learning to Perceive in Deep Model-Free Reinforcement Learning

Extended Abstract

Gonçalo Querido
INESC-ID & Instituto Superior Técnico
Lisbon, Portugal
goncalo.querido@tecnico.ulisboa.pt

Alberto Sardinha
INESC-ID & Instituto Superior Técnico & PUC-Rio
Lisbon, Portugal
jose.alberto.sardinha@tecnico.ulisboa.pt

Francisco S. Melo
INESC-ID & Instituto Superior Técnico
Lisbon, Portugal
fmelo@inesc-id.pt

ABSTRACT

This work proposes a novel model-free Reinforcement Learning (RL) agent that is able to learn how to complete an unknown task by having access to only a part of the input observation. We extend the *recurrent attention model* (RAM) and combine it with the *proximal policy optimization* (PPO) algorithm. Despite the visual limitation, we show that our model matches the performance of PPO+LSTM in two of the three games tested.

KEYWORDS

Reinforcement Learning; Model-Free; Attention Mechanism; Hard Attention; Active Perception; Visual Attention

ACM Reference Format:
Gonçalo Querido, Alberto Sardinha, and Francisco S. Melo. 2023. Learning to Perceive in Deep Model-Free Reinforcement Learning: Extended Abstract. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 3 pages.

1 INTRODUCTION

In this paper, we present the first RL architecture that implements an attention mechanism similar to the one humans have. Applying such a mechanism allows the model to only process the pixels it perceives as the most useful, which makes it much more computationally efficient.

The closest work to our approach is the model proposed by Mnih et al., called RAM [2], which implements the same attention mechanism in the context of image classification. The authors introduce the crucial concept of a *glimpse*, a retina-like representation of a portion of an image centered around a location l. The region of the image around l has high resolution; regions further away from l have increasingly lower resolution.

In this paper, we show that it is possible to attain state-of-the-art performance in some complex control tasks with limited (but active) perception. Our work proposes a novel architecture that combines a glimpse-based attention mechanism with a model-free RL algorithm (PPO). Our results show our model can match the performance of PPO+LSTM in two of the three games tested while processing a significantly smaller number of pixels from the input images. A full version of this paper is available in [3].

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

2 GLIMPSE-BASED ACTOR-CRITIC (GBAC)

In this section, we introduce our model called Glimpse-Based Actor-Critic (GBAC), which combines a hard attention mechanism with PPO. Compared to other RL models, GBAC processes much fewer pixels, and its training parameters do not depend directly on the size of the input, making it more efficient.

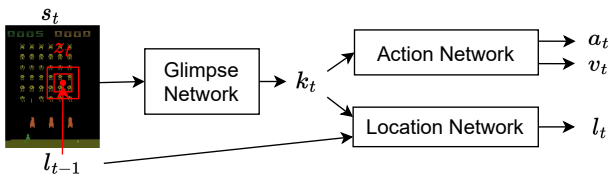

Figure 1: Overview of the architecture of GBAC

Figure 1 presents an overview of the GBAC architecture. The Glimpse Network is the module responsible for extracting the region the agent focuses its attention from the game frame. The Action Network selects the game action a_t the agent should perform in each timestep and evaluates whether the agent is performing well. The Location Network is the module behind the hard attention mechanism and is responsible for choosing the image coordinates l_t where the agent should look in the next timestep. The coordinates of l_t are sampled from a truncated normal distribution where this network gives the mean, and the standard deviation is a fixed value.

The agent receives a game frame s_t and the set of coordinates l_{t-1} from the previous timestep. From the frame s_t, the Glimpse Network takes as input the glimpse z_t, centered at l_{t-1}, and extracts the features k_t. After that, the vector k_t is used as the input of the Action Network. This network outputs not only the agent's next action a_t but also an estimate v_t of the value function. The vector k_t is also used by the Location Network, which merges it with the features extracted from the location l_{t-1} to select the next location coordinates l_t.

The training process of our model is very similar to the one presented by Schulman et al. in the PPO paper [4]. We follow their suggestions; the only difference is that we have two policy losses instead of just one. Therefore, the objective's formula is the following:

$$L_t(\theta) = \hat{\mathbb{E}}_t \left[L_t^{CLIP_a}(\theta) + L_t^{CLIP_g}(\theta) - \alpha L_t^{VF}(\theta) + \beta B[\pi_a](s_t) \right] \quad (1)$$

where α and β are coefficients, B is an entropy bonus that promotes the exploration of the environment, and L_t^{VF} is the squared-error loss of the value function.

Table 1: Comparison between the training and testing performances of all the models, in the three games tested. We present the results of the best glimpse configuration for each number of patches.

		PongNoFrameskip-v4		SpaceInvadersNoFrameskip-v4		CarRacing-v0	
Model	**Glimpse**	**Max. Train Avg.**	**Test Avg.**	**Max. Train Avg.**	**Test Avg.**	**Max. Train Avg.**	**Test Avg.**
PPO	full img.	20.91 ± 0.11	20.83 ± 0.13	2261.90 ± 295.87	2221.62 ± 201.31	867.16 ± 6.64	824.31 ± 8.04
PPO + LSTM	full img.	20.03 ± 0.23	19.85 ± 0.39	900.20 ± 79.16	812.58 ± 111.51	783.62 ± 11.58	659.73 ± 24.42
GBAC	2 patches	7.15 ± 20.80	6.64 ± 21.14	444.18 ± 40.78	341.47 ± 25.71	694.50 ± 107.94	641.11 ± 57.42
GBAC	3 patches	20.06 ± 0.44	19.82 ± 0.88	596.50 ± 182.35	544.43 ± 166.79	676.82 ± 84.89	564.00 ± 56.42
GBAC	4 patches	-14.86 ± 5.39	-15.77 ± 4.82	439.72 ± 26.27	378.00 ± 28.70	-	-
PPO Random	best conf.	-19.87 ± 0.05	-20.16 ± 0.11	516.62 ± 60.59	467.38 ± 50.53	622.41 ± 17.89	589.87 ± 13.19

3 EXPERIMENTAL EVALUATION

In this section, we explain our evaluation process and present the results. We decided to test our agent in three games: Pong and SpaceInvaders from the Atari 2600, and CarRacing from OpenAI Gym [1].

We chose to compare GBAC against three different versions of PPO. The first one is the original PPO agent presented by Schulman et al. [4] and the second is the PPO+LSTM variant. The third version of PPO is a modification of the PPO+LSTM algorithm where the restriction of only using a small portion of the game frame was imposed. But, different from GBAC, the coordinates where the agent looks are chosen randomly. This allows us to test if the perception mechanism implemented in our model is better than one that makes its choices randomly.

To evaluate the RL models, during training and testing, we average the episodic return each agent achieved over the last 100 episodes. Then, for each configuration tested, the results are always the average over three runs and their respective standard deviations are also presented.

3.1 Performance Analysis

In this section, we study not only the impact that different sizes of glimpses and different numbers of patches have on the performance of our agent but also how the best configuration performs against the three versions of PPO.

In our architecture, each glimpse can have two or more patches. Since we stipulated that each new patch has double the size of the previous, increasing the number of patches results in glimpses with a smaller focus region. The higher the number of patches, the larger the "peripheral vision" of our model. Nonetheless, this increase in information comes with the price of it not being as detailed as the portions of the image closer to the focal point.

Table 1 shows the performance of the multiple models tested in our study. When analyzing the multiple configurations of GBAC, we can conclude that the performance of our model does not increase linearly with the number of patches. It reaches a point where the information lost with the reduction of the glimpse size is more significant than the information gained with the addition of another patch. The optimal number of patches is three for the Atari games and two for CarRacing. Those numbers of patches proved to be the right balance between having a patch size that discarded the irrelevant information, allowing the model to just focus on the most important, and not being too small such that after rescaling the larger patches, it was still possible to understand what was present in the agent's "peripheral vision". Finally, we saw that the largest glimpse size possible for each number of patches is the one that produces the best results. Both in the two Atari games (3 patches) and in CarRacing (2 patches), the best scores were achieved using glimpses of size 40x40.

When comparing GBAC with the PPO+LSTM, we see that our model is capable of matching its performance in Pong (3 patches) and CarRacing (2 patches), while just using a portion of the game frame. Despite, in SpaceInvaders, it not being able to match the performance of PPO+LSTM, we still consider it an interesting result, considering the viewing restrictions of our problem. While processing 86% fewer pixels than PPO+LSTM (4.800 vs. 33.600) in each timestep, GBAC only had a performance drop of 33%.

In relation to the PPO variant with random glimpses, we discovered that when the environment has many places where the agent could focus to successfully play the game (the case of SpaceInvaders and CarRacing), the importance of choosing a good glimpse location decreases because the performance difference between having a perception mechanism or choosing a location randomly is smaller. In Pong, since the agent needs to keep track of the location of each paddle and the ball, taking glimpses at random locations makes the agent perform poorly.

4 CONCLUSION

This work proposed a solution for the problem of an agent that has limited vision. Hence, the agent must not only decide its action but also choose which part of the environment it should look at. To solve this problem, we present GBAC, a model that combines a glimpse-based hard attention mechanism with PPO. We proved that, for games like Pong and CarRacing, our model is already capable of achieving similar performance to the PPO+LSTM version. On other games like SpaceInvaders, a drop in performance is verified, but we believe there still is room for improvement.

ACKNOWLEDGMENTS

This work was partially supported by Portuguese national funds through Fundação para a Ciência e a Tecnologia (FCT) under projects UIDB/50021/2020 (INESC-ID multi-annual funding), PTDC/CCI-COM/5060/2021 (RELEvaNT), PTDC/CCI-COM/7203/2020 (HOTSPOT). In addition, this research was partially supported by the Air Force Office of Scientific Research under award number FA9550-22-1-0475 and an EU Horizon 2020 project (TAILOR) under GA No. 952215.

REFERENCES

[1] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym.

[2] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. 2014. Recurrent Models of Visual Attention. In *Proceedings of the 27th International Conference on Neural Information Processing Systems* (Montreal, Canada) *(NIPS'14, Vol. 2)*. MIT Press, Cambridge, MA, USA, 2204–2212.

[3] Gonçalo Querido, Alberto Sardinha, and Francisco S. Melo. 2023. Learning to Perceive in Deep Model-Free Reinforcement Learning. *arXiv preprint arXiv:2301.03730* (1 2023). https://doi.org/10.48550/arxiv.2301.03730

[4] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov Openai. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347* (7 2017). https://doi.org/10.48550/arxiv.1707.06347