Informed Exploration in Atari Games via Next State Prediction

Tudor-Victor Armegioiu[1] and Andreea-Alexandra Musat[1]

[1]Department of Informatics, Technical University of Munich, Munich 85748

Abstract

In the last years, deep learning based computer vision algorithms have reached super-human performance in applications such as classification [1], [2], object detection [3] and many others. Reinforcement learning (RL) is a domain that has greatly benefited from these advances. In the model-free form, where the agent directly learns how to act, the behaviour can be modelled, for example, using a policy network [4] or a state-action value network [5], both of them commonly being convolutional neural networks (CNNs). Similarly, in model-based RL, the agent indirectly learns the optimal behaviour by making use of some model of the world, which can be used when a simulator is not always available, such as in autonomous driving [6], for creating an intrinsic reward signal when real rewards are sparse [7] and others.

The aim of this project is implementing an Atari agent endowed with with such a world model and using it to derive an informed exploration signal. More precisely, the agent should learn to predict the consequences of its actions and based on these predictions, it should choose the action which leads to the least visited state. So, for each action, the agent 'imagines' the future (next frame prediction component). Then, each of these possible futures is compared to a set of recent frames and the action that produces the most dissimilar frame is chosen (novelty detection component). Thus, the discovery of potentially better states is encouraged. A data pipeline for the project is presented in Figure 1.

A similar idea was explored in [8], where better scores on several Atari games were obtained using this exploration method instead of random exploration. For the next frame prediction component, two CNN-based architectures were proposed: one which stacks the sequence of consecutive frames and one which uses an RNN to embed them before feeding them to a CNN. The models are trained to minimize the Mean Squared Error (MSE) over multiple time steps, because small errors compound when predicting frames in a more distant future. A simple Gaussian kernel was used for measuring the similarity between the imagined frame and a trajectory memory.

For our implementation, we plan to use a U-Net architecture [9] as a baseline for the next frame prediction model. Because the next frame depends on the action, the model should be conditioned on it. We will condition on the action by injecting it using multiplicative interactions in the middle embedding layer, as shown in Figure 1.

Then, we will use generative adversarial training for another model. Similar to the Pix2Pix architecture [10], we will implement a generator $G : \{s_t, a, z\} \mapsto s_{t+1}$ conditioned on the previous frames s_t, as well as on the action a, injecting it as before. The discriminator $D : \{s_t, a, s_{t+1}\}$ will also be conditioned on both of them. In order to discourage blurry results, an L1 loss term will be added to the classical GAN loss.

For the simple U-Net baseline, the novelty detection module can be implemented by noting the following: the loss of the next state predictor will be higher for pairs of states and actions that it hasn't seen together too often. Thus, we can directly use the loss of the next state prediction model as a novelty signal.

We then plan to expand the novelty detection pipeline using an approach presented in [11], which was originally used for modelling an intrinsic reward. The crux of the approach relies one using two neural networks having the current state and an action as input. One of the networks is a fixed and randomly initialized, called the target network. The other network, called the predictor, is tasked on producing the same output as the target network and is trained to minimize the L2 loss between their outputs. The purpose of this approach is to take advantage of the fact that the target network, since it is fixed by design, will always produce the same output for any given s_t - hence, the predictor network would in time learn to distill the target one. The novelty will be defined as the loss of this predictor network.

We have split the project into 4 stages of 2 weeks each, as follows:
1. Set up the Pong and Breakout Atari environments. Implement a random agent, as well as a DQN agent with an ε-greedy policy and benchmark them.
2. Implement an action-conditional U-Net video prediction architecture and a basic image similarity module that uses the U-Net loss for measuring novelty.
3. Experiment with a video prediction architecture based on Pix2Pix and an image similarity model using random network distillation.
4. Presentation preparation, final code cleaning and buffer for unexpected problems.

For a more detailed timeline, as well as future code, you can check our github repo here.

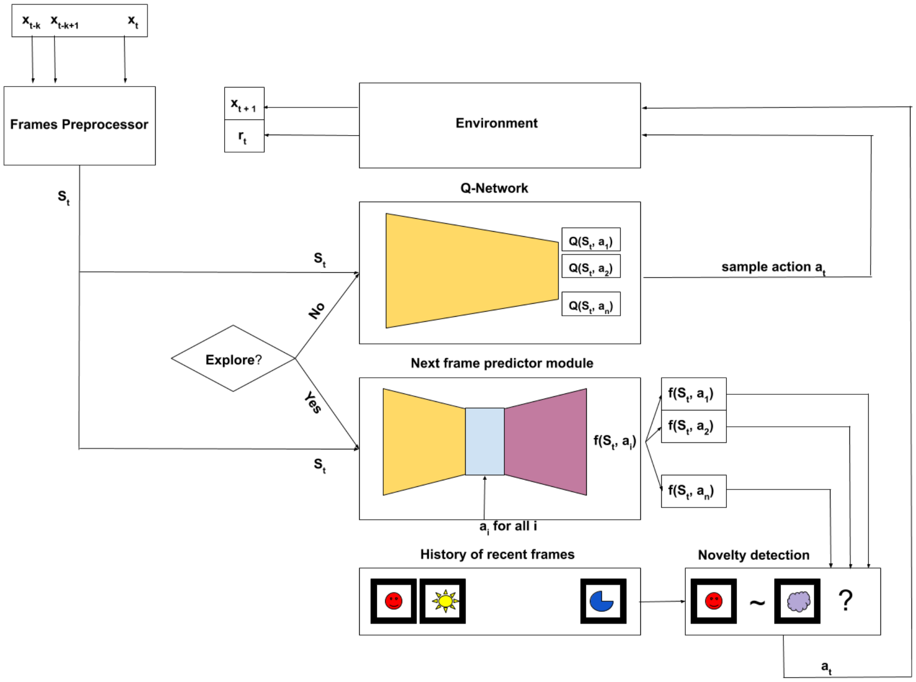

Figure 1: Information flow pipeline: orange blocks are convolutional layers, purple blocks are transposed convolutions. The environment outputs raw frames x_t which are stacked and processed as $s_t = P(x[t-k:t])$. From this, if exploring, the next frame predictor module outputs an imagined frame $f(s_t, a_i)$ for each possible action a_i and the novelty detection module outputs the action a_t which produces the most dissimilar frame compared to what has been seen.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[3] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *arXiv preprint arXiv:1905.05055*, 2019.

[4] S. Levine and V. Koltun, "Guided policy search," in *International Conference on Machine Learning*, pp. 1–9, 2013.

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[6] E. Santana and G. Hotz, "Learning a driving simulator," *arXiv preprint arXiv:1608.01230*, 2016.

[7] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 16–17, 2017.

[8] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh, "Action-conditional video prediction using deep networks in atari games," in *Advances in neural information processing systems*, pp. 2863–2871, 2015.

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[10] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

[11] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, "Exploration by random network distillation," *arXiv preprint arXiv:1810.12894*, 2018.