Imitating Deep Reinforcement Learning Policies with Interpretable Decision Trees

1 Aim of the Project

Recent advances in deep learning allowed the community to learn very good policies for the game of Go, Atari video games, or robot control in simulation. In these applications, policies are implemented with deep neural networks and a reinforcement learning algorithm is used to learn their parameters. Deep neural network policies however, are rarely used to make real-life decisions because they are not interpretable. Indeed, neural networks are black-box making it hard to make sense of action decisions computed by a trained neural network policy.

It is possible to interpret such deep neural network policies a posteriori with imitation learning. VIPER [1] algorithm uses the CART algorithm to fit a decision tree to the neural network policy. However many decision tree learning algorithms have been developed recently and are know to return better trees than CART. The goal of the is project is thus to try to imitate deep policies with those new algorithms.

This project aims at benchmarking different decision tree learning algorithms to control robots in simulation. For that, the student will use the VIPER algorithm [1] to imitate deep reinforcement learning policies (Figure 1). The student will answer the question: "what trees can be imitated from a deep policy with exisitng decision tree algorithms ?". Quantitatively, the student will compare the results of the VIPER algorithm when used with Quant-BnB, CART, and MFOCT.

2 Guidelines and Supervision

The student should be familiar with reinforcement learning `https://en.wikipedia.org/wiki/Reinforcement_learning` ,`pytorch`, and `scikit-learn`. All algorithms mentioned above are open-sourced and easy to run. The student should start by collecting training data from deep reinforcement learning policies available online. For that, the student will get familiar with the `stable-baselines3` library. Then the student should fit decision trees to those expert data using the three algorithms mentioned above. Finally, the student will evaluate the performances of the fitted decision tree policies on the `Mujoco` benchmarks. Great attention will be given to the rigor of the experimental process, coding quality, and results presentation with plotting tools such as `matplotlib`. All sources and codes will be given to the students.

The main supervisor of this project will be Hector Kohler, a 2nd year PhD student in the internationally recognised Scool team. As such, he will have plenty of time to supervise and guide the student through in-person meetings and `discord` discussions.

References

[1] Osbert Bastani, Yewen Pu, and Armando Solar-Lezama. "Verifiable reinforcement learning via policy extraction". In: *Advances in neural information processing systems* 31 (2018).

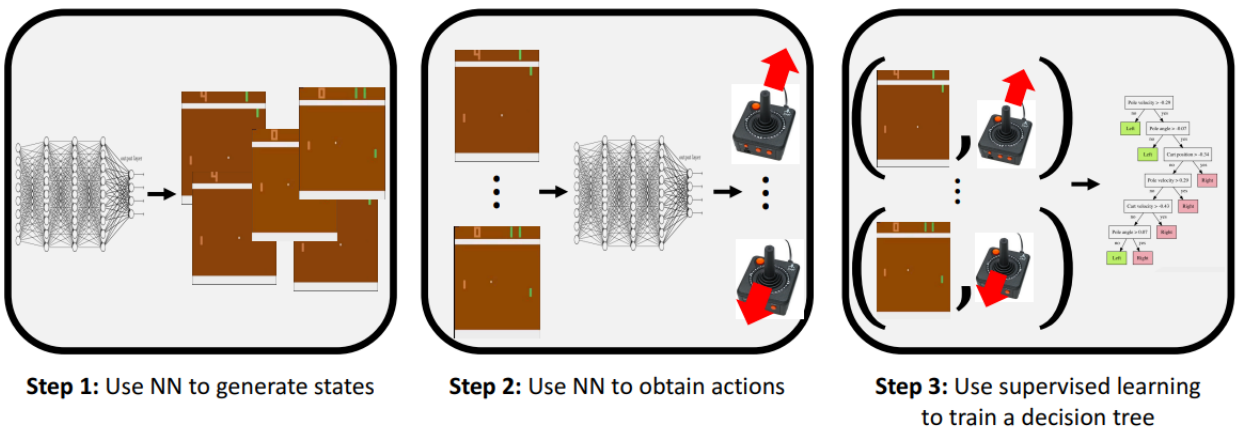

Figure 1: Imitation Learning