Verifiable and Interpretable Reinforcement Learning through Program Synthesis

Abhinav Verma
Rice University
6100 Main Street
Houston, Texas 77005

Abstract

We study the problem of generating interpretable and verifiable policies for Reinforcement Learning (RL). Unlike the popular Deep Reinforcement Learning (DRL) paradigm, in which the policy is represented by a neural network, the aim of this work is to find policies that can be represented in high-level programming languages. Such programmatic policies have several benefits, including being more easily interpreted than neural networks, and being amenable to verification by scalable symbolic methods. The generation methods for programmatic policies also provide a mechanism for systematically using domain knowledge for guiding the policy search. The interpretability and verifiability of these policies provides the opportunity to deploy RL based solutions in safety critical environments. This thesis draws on, and extends, work from both the machine learning and formal methods communities.

Introduction

Many recent advances in Reinforcement Learning have been through models that rely on a Deep Neural Network (DNN) (Mnih et al. 2015). However, DNNs have been called "black-box" models due to a fundamental drawback, these models are difficult to interpret or to be checked for consistency for some desired properties. Consequently, there is a growing consensus that further advancements in AI research will require models that combine DNNs with other approaches and methods. The primary contribution of this thesis will be to explore and exploit the connections between automatic program synthesis and deep reinforcement learning.

We propose a learning framework, called Programmatically Interpretable Reinforcement Learning (PIRL) (Verma et al. 2018), that is based on the idea of learning policies that are represented in a Domain Specific Language (DSL). An example of this approach, is to synthesize a program that drives a car around a track, by controlling the car's acceleration and steering. The following example shows the kind of high-level program our method finds for acceleration:

$$\textbf{if } -0.001 < \textbf{hd}(\texttt{TrackPos}) \textbf{ and } \textbf{hd}(\texttt{TrackPos}) < 0.001$$
$$\textbf{then } PID_\theta(\texttt{Target}_1)$$
$$\textbf{else } \ PID_\theta(\texttt{Target}_2)$$

In contrast, the DNN that represents a similar policy has three hidden layers with 600 nodes each.

The intuition behind this work is that structured programs in a high level DSL have three key benefits. First, the DSL can be designed to be human-readable and is hence more interpretable than a DNN. Second, the language can be used to implicitly encode the learner's inductive bias, which is useful for agent generalization. Finally, it allows us to use symbolic program verification techniques to formally reason about the learned policies and check consistency with correctness properties.

There have been efforts in deep learning that aim to make DNNs more interpretable (Montavon, Samek, and Müller 2017), and to formally verify DNNs directly (Katz et al. 2017). The work in this thesis differs from these approaches in that our framework generates high-level program source code as output, which is used in place of the policy represented by the DNN. Efforts have also been made to use neural networks for learning programs in the growing field of neural program synthesis and induction, (Murali et al. 2018) is one such example. In these methods a DNN is typically trained to guide the program search. Our approach differs from these efforts in that we use DNNs trained on the RL environment's task directly.

Approach

We formalized the problem of performing synthesis for reinforcement learning policies in the PIRL framework (Verma et al. 2018). In summary, we model a reinforcement learning setting as a Partially Observable Markov Decision Process, and we define a DSL which places a syntactic restriction on the program search space. Then our goal is to find a program with optimal reward: $e^* = \arg\max_{e \in [\![\mathcal{S}]\!]} R(e)$. Here $[\![\mathcal{S}]\!]$ denotes the set of programs permitted by a DSL $\mathcal{S}$, and $R(e)$ is the agent's expected aggregate reward under the policy represented by e.

The use of a DSL to provide syntactic constraints is inspired from work in the programming languages community, where this approach has been formalized in a framework called Syntax-Guided Synthesis (Alur et al. 2015). The DSL also provides a principled mechanism to systematically include domain knowledge into the policy search. This mechanism modifies the learner's inductive bias. We note that while DRL algorithms excel in end-to-end learning, to

the best of our knowledge they currently lack methods for specifying an inductive bias for the learner.

A key technical challenge in PIRL is that the space of policies, despite syntactic constraints, is typically vast and nonsmooth. This makes direct policy search extremely difficult. To address this, we proposed a new algorithm called Neurally Directed Program Synthesis (NDPS). NDPS first uses DRL to compute a neural policy that has high performance. This network is then used to direct a local search over programmatic policies. This strategy, inspired by imitation learning (Ross, Gordon, and Bagnell 2011), allows us to perform direct policy search in a highly nonsmooth space. However, one key difference is that NDPS uses the expert trajectories to only guide the local program search, unlike the imitation learning setting where the goal is to match the expert demonstrations perfectly.

We evaluate our approach in the task of learning to drive a simulated car in the The Open Racing Car Simulator (TORCS) environment. Our experiments demonstrate that NDPS is able to find interpretable policies that, pass some significant performance bars.

Ongoing and Future Work

There are many interesting directions to explore in the area of programmatic policies for RL. I have identified two broad categories for immediate attention.

Synthesis and Verification. Our policy synthesis approach is uniquely positioned to benefit from improvements in both DRL and program synthesis techniques. New methods in either of these fields can be adopted in the NDPS algorithm. I am also developing a new algorithm, that co-evolves the neural and programmatic policies by interleaving the training of a DRL policy with the synthesis of a programmatic policy. Using more powerful verification systems to prove more complex and useful properties is another direction I am exploring. Relatedly, I am developing methods to use verification specifications to prune the search space during the synthesis of programmatic policies.

Complex Applications There are many complex, 'real-world' applications that we hope to explore as part of this thesis. For example, new adaptive drug therapies have been discovered via DRL for some diseases. These therapies are unlikely to get regulatory approval due to the black-box nature of DNNs. I am working with collaborators, to generate interpretable versions of the existing neural policies via the PIRL framework. This is an example of a situation where the explainability of the policy to a human expert is fundamental to the adoption of the RL based solution. Members of my lab are applying the PIRL framework to the problem of path and task planning for quadcopters. These policies need to have strong safety guarantees, as the cost of any failure is catastrophic.

Contributions and Impact

For many domains DRL models are the current state of the art method for finding RL policies. Therefore, methods that address their drawbacks are likely to have a significant impact on the field. This thesis aims to address two fundamental problems with DRL models, namely interpretability and verifiability. In already published work (Verma et al. 2018) we have formalized a new learning paradigm and shown promising results with a method that tackles both of these drawbacks.

In work currently under development, we propose a new algorithm which interleaves the training of neural and programmatic agents, thus significantly improving the performance and generalizability of the current best RL policy finding methods. Going forward we will explore generating results with stronger verification guarantees and applications of the PIRL framework to safety critical cyber-physical systems.

Successful completion of this research program is likely to create new avenues for researching connections between the Machine Learning and Formal Methods literature. Furthermore, by addressing some of the major concerns about the current RL methods, this work is likely to advance the applicability and adoption of RL based solutions to many real world problems. It is also a significant possibility that the contributions of this thesis will be applicable to other paradigms of machine learning, like supervised learning, that are currently dominated by deep neural networks.

References

Alur, R.; Bodík, R.; Dallal, E.; Fisman, D.; Garg, P.; Juniwal, G.; Kress-Gazit, H.; Madhusudan, P.; Martin, M. M. K.; Raghothaman, M.; Saha, S.; Seshia, S. A.; Singh, R.; Solar-Lezama, A.; Torlak, E.; and Udupa, A. 2015. Syntax-guided synthesis. In *Dependable Software Systems Engineering*.

Katz, G.; Barrett, C. W.; Dill, D. L.; Julian, K.; and Kochenderfer, M. J. 2017. Reluplex: An efficient smt solver for verifying deep neural networks. In *Computer Aided Verification*, 97–117.

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.

Montavon, G.; Samek, W.; and Müller, K. 2017. Methods for interpreting and understanding deep neural networks. *CoRR* abs/1706.07979.

Murali, V.; Qi, L.; Chaudhuri, S.; and Jermaine, C. 2018. Neural sketch learning for conditional program generation. In *International Conference on Learning Representations*.

Ross, S.; Gordon, G. J.; and Bagnell, D. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011*, 627–635.

Verma, A.; Murali, V.; Singh, R.; Kohli, P.; and Chaudhuri, S. 2018. Programmatically interpretable reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 5045–5054. Stockholmsmässan, Stockholm Sweden: PMLR.