Hierarchical Abstraction for Verifiable Learning

Shashank Pathak, Luca Pulina, Armando Tacchella, Giorgio Metta

We argue that hierarchical learning and related abstractions should not only be seen with scalability in mind, but also succinctness with which learned policies can be represented. Such succinct representations should then allow for high-level temporal reasonings over the policy, such as checking for safety properties.

Reinforcement Learning(RL) is an adaptive learning paradigm where the agent learns through enaction over environment. In practical applications, the learned policies are stochastic, and probability of choosing unsafe actions could not be diminished beyond a point. Though the rewarding schema could be tuned to make learning aware of unsafe states, the overall probability bound on such risk could not be diminished arbitrarily. Hence, verification of learned policies through formal methods could be beneficial.

Assuming some discretized abstraction of state-space, a fairly general and model-free verification of policy could be performed. In our experience with the humanoid iCub, this abstraction is done by manually crafting low-level motion control policy, inspired by human arm movements. Any RL under such state-action space, combined with its learned policy, can be modeled as discrete-time Markov chain. Using Probabilistic Computational Tree Logic (PCTL), safety criteria can be specified in most general way. Thereafter, probabilistic model checking tools can be used to provide quantitative measure of inherent risk in policies. Using the tool COMICS, also a counter-example (a countable set of paths that lead to unsafe states) can be generated. Under mild assumptions, e.g., no-loops in the DTMC,such counter-example can be leveraged to make policy safer.

However, the approach outlined above works only if the abstracted state-action space is discrete, i.e., the system allows a two level hierarchy where the top level is discrete. The naive approach of discretizing state space to a very fine resolution does not work, due to exponential blow-up in complexity. Hierarchical or incremental learning is often viewed as a cure for this. Here we provide another argument in favor of such incremental way of learning on abstract state-space, viz., cognition. In fact, there is a fine balance between complexity of the domain in which the agent lives and succinctness of the policy it learns. Often the simplest domains are the ones with precise representation of state space evolution, such as discrete state dynamic programming. Whereas, domains that are too complex to have a-priori discretization, use generalizing functions (e.g., RBFs), where the power to reason on temporal properties of the policy is lost. These methods are nevertheless quite efficient in explored state-space; they only need finite observations to generalize over a continuous state-space.

RL based on U-trees strikes a fine balance. State-space is not partitioned a-priori, but based on statistical observation of experiences. Such an abstraction is known to preserve the good properties of discrete-state abstractions, but it is more directly amenable to formal verification than the use of generalizing functions. Other hierarchical RL paradigms may prove equally worthwhile in this direction.