Learning Neuro-Augmented Domain-Specific Languages

Abstract

Program synthesis offers a powerful approach to solving machine learning problems by allowing users to define domain-specific languages (DSLs) that encode important inductive biases. A well-designed language balances expressiveness and conciseness, which allows for sample-efficient learning. However, designing such languages is a challenging engineering task. A language that is "too expressive" may result in hypothesis spaces that are too large to be practical, while one that is "too concise" may fail to include effective solutions. In this paper, we introduce Neuro-Augmented Language (NAL), an approach for learning DSLs that achieve this balance. NAL begins with a minimalist DSL and uses a program synthesizer to generate solutions to target problems. Parts of the synthesized program are then replaced with neural networks trained to perform specific sub-tasks. These learned networks are incorporated back into the DSL as new primitives, expanding its expressiveness. This iterative process refines the DSL by leveraging the expressive power of neural networks, thus producing a neurosymbolic solution to the original task and an augmented DSL that can be reused in downstream tasks. We evaluate NAL in both reinforcement learning and supervised learning domains. In reinforcement learning, NAL learns to play a real-time strategy game. In supervised learning, NAL generates regression models for time series prediction. Our results show that NAL can learn effective DSLs that can be reused in downstream tasks, thus reducing the engineering hurdle of designing effective DSLs.