1 Abstract

One of the outstanding problems in data-trained neural networks (NNs) is the design of the NN’s architecture: that is the number of neurons and their connections. Current state-of-the-art practices typically choose a NN architecture either according to heuristics or else via a computationally expensive iteration schemes that involves adapting the architecture iteratively and re-training the NN. Besides being computationally taxing, neither of these provide any assurances that the resultant architecture is sufficient to permit adequate performance of the final trained NN. Indeed, the absence of such a guarantee on the architecture necessarily precludes such a guarantee on the trained network.

In Ferlez and Shoukry [2019], we addressed the problem of automatically designing a regression NN architecture to generate (control) actions for a linear dynamical system under specified performance objectives. Specifically, we proposed AREN, an algorithm that generates assured Rectified Linear Unit (ReLU) NN architectures: given a linear dynamical system, AREN designs a ReLU NN architecture with the assurance that there exist network weights that exactly implement a Model Predictive Control (MPC) expert controller. AREN thus offers new insight into the design of ReLU NN architectures for the control of linear systems and Deep Reinforcement Learning, where MPC experts are commonly used. Instead of the computationally intensive or heuristic-driven methods described above, AREN can provide an adequate NN architecture before training begins.

AREN achieves this using two novel features.

- First, AREN employs a novel ReLU architecture based on the Two-Level Lattice (TLL) representation of CPWL functions in Tarela and Martínez [1999]; this architecture comes from a careful interpretation of the recent results in Arora et al. [2016] and Shuning Wang and Xusheng Sun [2005] in the context of the TLL representation. Crucially, this architecture can implement any CPWL function as a ReLU NN with an architecture determined wholly by the number of linear regions in the function.
- Second, AREN uses a computationally efficient scheme to over-approximate the number of linear regions in the MPC controller without obtaining the MPC controller explicitly. This involves converting the state-region specification equation for the optimal controller into a single, state-independent collection of linear-inequality feasibility problems – at the expense of over-counting the number of affine regions that might be present in the optimal MPC controller. This requires an algorithmic solution rather than a closed-form one, but
our algorithm executes quickly even on problems for which finding the MPC controller is prohibitive.

Together, these results allow AReN to design an assured ReLU NN architecture for the MPC control problem without explicitly solving for the optimal MPC controller.

References


