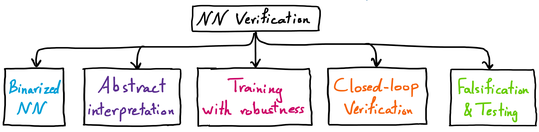
Review VAS - Reaching out towards fully verified Autonomous systems

Sunday, 25 October 2020 16:09

- A system is defined as autonomous; if it can operate in a reliable manner without requiring "frequent" human intervention.
- Safety-critical systems are the systems that are required to perform in a provably safe manner despite the uncertainties about the environment and the numerous limitations on the system's ability to sense, compute and actuate.
- Machine learning approaches for building autonomous systems use a variety of models that are trained during design.
- The key challenge: building systems with neural network components that are also guaranteed to satisfy key safety and liveness properties, even in the presence of significant uncertainties in the environment.
- Verification of Neural Networks:
  - DNNs are essentially acyclic computation graphs formed by composing activation functions, but their behavior can be complex and highly non-linear.
  - NN are used as components inside a closed-loop autonomous system.
  - → verification problem:
    1. component-wise specification; involving just the neural network.
    or 2. end-to-end approach; studies the network in composition with other parts.



- Binarized NN:
  pros: 1. Unit weights yield computational savings (efficient).
  2. Amenable to implementation as digital circuits (practical).
  - Ordered Binary Decision Diagram (OBDD) is learned locally to abstract parts of the neural network.
    Verifying binarized NN by local automaton learning [Shih'19]
  - BNN could be reduced to hardware verification problem → speed-ups in performance.
    Verification of binarized NN [Cheng'17]
- Abstract interpretation:
  was developed to systematically propagate sets of reachable states of a program.
  - abstract domains represent the reachable set of states.
  - zonotopes are used to over-approximate the non-convex set of possible outputs for each layer of the network → allows layer-by-layer analysis to compute sound over-approximations for the output of the neural network. Ai2 [Gehr'18]
  - Computing the output ranges as a union of convex polytopes.
    doesn't use SMT or MILP solvers → accurate estimates of the output range.
    manipulating polyhedra is expensive → restricted to small networks.
    Reachable set computation and safety verification for NN [Xiang'17]
  - Range computation using symbolic intervals. ReluVal [Wang'18]
  - tight ranges for individual neurons. Maximum resilience of ANN [Cheng'17]
- Training with Robustness:
  Verification to improve the learning process.
  - Mixtrain [Wang'18] uses the output set estimates computed by verification tools to incorporate robustness in the training phase
    → mean to defend against any adversarial perturbations of the input, but it is more expensive than standard approaches.
- Closed-loop verification:
  - Compute reachable set of states of the closed-loop involving an ODE and NN by: 1. computing Taylor models (polynomial + errors) as an approximate of the behavior of the NN in compact domain.
    2. Using standard reachability tools on these models.
    Reachability analysis of neural feedback systems using regressive... [Dutta'19]
  - Reachnn [Huang'19] approximates the NN controller with Bernstein polynomials to deal with activation functions that are more general than ReLU.
  - Verisig [Ivanov'19] proposes modeling activation functions using differential equations evolving over time → encode a network as an ODE itself.
    this allows the transformation of a single layer of NN into a hybrid system, and then use standard reachability analysis tools.
  - Barrier certificates is an approach to establish safety properties of dynamical systems. It can be synthesized using an SMT solver to prove properties of ODEs with neural networks as feedback.
    Reasoning about safety of learning-based components... [Tuncali'18]
  - Using SMT based approach to construct a finite state abstraction of the closed loop system using fixed set of predicates to partition the state space.
    Formal verification of NN controlled autonomous systems [Sun'19]
- Falsification and Testing:
  - The falsification problem consist of finding an execution that violates a requirement.
  - Falsification algorithms for CPS implement efficient heuristics to search for a system's input that can falsify a requirement.
  - Finding an adversarial example by minimizing the robustness function of the Signal Temporal Logic (STL) formula via gradient descent.
    Gray-box adversarial testing for control systems [Yaghoubi'19]
  - Two steps: 1. Falsify an abstraction of the NN component and CPS.
    2. Confirm the counterexample in the NN component.

- Challenges:
  1. Specifications:
     Problem: formally specifying the behavior of perception systems that classify sensor data including images and LIDAR data.
     Challenge: specifying what a valid image is in a logical formalism that is compatible with verification tools.
     Current solutions:
     1. sidestep functional specification in favor of requiring the classifier to be robust to perturbations around training examples.
     2. simplify the sensor's capabilities to make modeling easier.
     3. uses generative models that specify inputs at a high-level.
  2. Scalability:
     Problem: the current networks are 100x or 1000x larger than the most efficient verification tools available.
     Possible solutions:
     1. improving constraint solvers by specializing them to handle NNs.
     2. Using abstractions
  3. Recurrent Networks: like LSTM.
  4. Runtime Verification:
     Problem: current static / pre-deployment verification.
     Possible solutions:
     1. Using LI-Simplex architecture that switch to a lower performance but formally validated control when an impeding failure is predicted.
        Using simplicity to control complexity [Sha'01]
     2. Verification to guarantee safety is shielding; uses a supervisor (shield) to monitor the execution of autonomous system and interve to enforce temporal logic properties if a violation is imminent.
        Safe reinforcement learning via shielding [Alshiekh'18]
     3. Monitoring viability rather than safety
        to sidestep the need to reason about the controller
        instead reason over the behavior of the plant model
        Model-predictive real-time monitoring of linear systems [Chen'17]