Efficient Formal Safety Analysis of Neural Networks

Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, Suman Jana

Adversarial Examples: Existing violations of NN safety

1. Symbolic Linear Relaxation

How to formally guarantee the absence of violations within bounded input ranges?

Output nonconvexity ⇒ need tight approximation

Tighter interval analysis for ReLU propagations
Partial input dependencies are preserved
Used to identify crucial overestimated nodes
Overestimated nodes: the node performs nonlinearity

2. Directed Constraint Relaxation

Locate influential overestimated nodes
Split each nonlinear ReLU into two linear cases
Solve each case with linear solver

How Neurify Solves This Problem?

Input intervals
Symbolic linear relaxation
DNN
Constraints
Linear solver
Safety property
Concrete sample
Check for violation

Split target node
Refine overest. node

Terminate:
(1) Proved safe
(2) Proved unsafe with counterexamples and
(3) Timeout

Interval & Linear solver

Given:
(1) Input ranges
(2) Targeted network and
(3) predefined safety property

Neurify:
(1) Locate overestimated nodes with symbolic intervals and
(2) Iteratively refine approximated output ranges with linear solver

Results

ACAS Xu: 5000 times faster than Reluplex and 20 times faster than ReluVal

DAVE: First system to scale to network over 10,000 ReLUs.
Various safety properties (e.g., L_1, L_∞, lightning, contrast) can be formally analyzed.

Code available at https://github.com/tcwangshiqi-columbia/Neurify