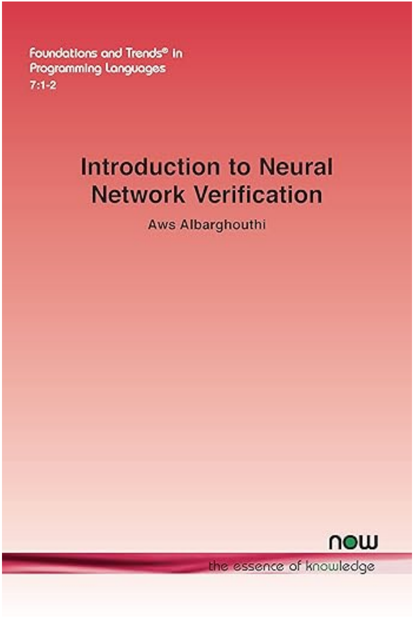
What is Neural Network Verification?

Yulia Alexandr

Book



Introduction to Neural Network Verification
by Aws Albarghouthi

A foundational text covering:

- Formal definitions of verification
- Methods and tools
- Applications to safety-critical systems

Available on arXiv.

Motivation

- Neural networks power critical applications:
  - Autonomous driving
  - Medical diagnosis
  - Defense and aerospace
- But they are black boxes:
  - Why did the model make a decision?
  - Can we guarantee it will not fail?

Definition and Scope

Neural network verification is the problem of proving that a neural network satisfies certain desirable properties, such as:

- Correctness: Does the output match the intended specification?
- Safety: Does it avoid unsafe behaviors under all valid inputs?
- Robustness: Do small input perturbations leave outputs unchanged?

Example Property

Image classifier for road signs

- Input: Image of a stop sign
- Property: Even with noise (graffiti, lighting, stickers), the output should still be "Stop"

Key Trade-offs in Neural Network Verification

Neural network verification systems need to balance three aspects:

- Soundness: Can we trust that the answer is correct?
- Completeness: Can an answer be provided in many cases?
- Scalability: Can the solver handle large networks efficiently?

Approaches to Neural Network Verification

- Constraint-based techniques (complete verification)
  - Represent the neural network and desired properties as a system of constraints
  - Solve the constraints to prove or disprove that the network satisfies the properties
- Abstraction-based techniques (approximate verification)
  - Instead of executing the network on a single input, operate on an infinite set of inputs
  - Show that all inputs in this set satisfy the desired properties
  - More scalable but may produce conservative (approximate) guarantees

Complete Verification

Encoding Correctness Properties

Idea: Reduce the verification problem to checking satisfiability of a logical formula.

Formal structure:

$$\{\text{precondition}\}, \quad r \leftarrow f(\text{x}), \quad \{\text{postcondition}\}$$

Example: Binary image classifier

- Network: $f_G : \mathbb{R}^n \to \mathbb{R}^2$
  Input: grayscale image $x \in [0, 1]^n$
- Goal: small perturbations do not change prediction
  Precondition: $|x - c| \leq 0.1$ (c = original cat image)
  Postcondition: $r_1 > r_2$ (probability of cat > probability of dog)

DPLL Algorithm

Question: How do we check whether a logical formula is satisfiable?

DPLL (Davis–Putnam–Logemann–Loveland) algorithm:

- Developed decades ago for checking satisfiability of Boolean formulas
- Can be extended to handle first-order formulas over theories
- Forms the basis of modern SAT and SMT solvers:
  - SAT solvers: check satisfiability of propositional logic formulas
  - SMT solvers: check satisfiability modulo theories (e.g., numbers, arrays, bit-vectors)

Simplex Algorithm

The simplex algorithm is used for solving conjunctions of linear inequalities (literals) over real numbers.

Steps in neural network verification:

- Initially handles linear constraints directly using the simplex method
- ReLUs introduce piecewise-linear behavior
- ReLUs are encoded as disjunctions and solved using a SAT solver
- Instead, we can extend the solver to handle ReLUs natively:
  - directly reason about ReLU activations as part of its computations, instead of converting them into another form that the solver can understand (like disjunctions for a SAT solver)

This combination allows verification of neural networks with piecewise-linear activations more effectively than treating ReLUs purely as Boolean constraints.

Approximate verification

Motivation:

- Complete verification encodes the network and correctness property as a formula in first-order logic
- Leads to NP-complete problems (e.g., satisfiability modulo linear real arithmetic / mixed-integer linear programming)
- Scaling to large neural networks is challenging

Approximate verification:

- Uses overapproximation (abstraction) of the network semantics
- Can prove correctness properties, but failure does not imply violation
- Based on abstract interpretation (Cousot & Cousot, 1977)
- Provides a scalable, pragmatic way to verify large neural networks