Certifying Differential Invariants of Neural Networks using Abstract Duals

Chandra Kanth Nagesh

Department of Computer Science
University of Colorado Boulder

Final Project; CSCI – 7135, Fall 2025

Setup

Safety Property

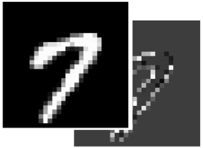
Given a neural network classifier $f : \mathbb{R}^n \to \mathbb{R}^k$, an input x_0, and a perturbation radius ϵ. Verify that the classification of x_0 is invariant within the ϵ-ball:

$$\forall x \in \mathbb{B}_\infty(x_0, \epsilon) : \quad \text{argmax}(f(x)) = \text{argmax}(f(x_0))$$

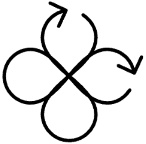
Original Image | Original Image $+\ \epsilon = 0.25$ | Original Image $+\ \epsilon = 0.1$

The Problem

Neural Networks are highly non-linear and vulnerable to *adversarial attacks*.



Original Image + tiny perturbations



Continuous input spaces

Safety-Critical Domains

Possible solution

Try computing an over-approximation of the reachable output set $\mathcal{R}_f(X_0)$ to prove safety

Why is it Interesting?

University of Colorado Boulder

The power of global bounds with simplicity of gradient analysis

1. **Simplicity beats complexity**
   - → DeepPoly[1] is strictly better than our method.
   - → However, *suprising results found!*
2. **Gradient Instability**
   - → Found 100% correlation between instability and failure at high ϵ
   - → Our method detects when the *local linearity assumption collapses*

$$f(x)_{true} - f(x)_{other} > \sup_{x \in X_0} ||\nabla f(x)||_1 \cdot \epsilon$$

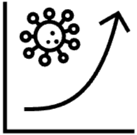
[1] Singh, G. et. al. (2019). An abstract domain for certifying neural networks. 3(POPL), pp.1–30.

Why is it Hard?

1. **Activation Non-Linearity:**
   - $\rightarrow$ **Sigmoid ($\sigma(x)$):** Non-convex. Linear relaxation introduce loose bounding boxes
   - $\rightarrow$ **ReLU ($\max(0, x)$):** Discontinuous derivative
     - $\nabla \text{ReLU}(x) \in \{0, 1\}$. At $x = 0$, the sub-gradient is the set $[0, 1]$
     - This causes the Lipschitz bound to explode: $K_{layer} = ||W|| \cdot 1$
2. **The Wrapping Effect:**
   - $\rightarrow$ Transforming a simple shape (zonotope/box) through a linear map rotates it
   - $\rightarrow$ Re-approximating it with axis-aligned bounds adds "dead space" volume exponentially with depth

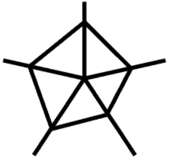
Contributions

Formalization:
Affine Arithmetic



Gradient Anomaly:
Depth < 3; Sigmoid

Instability Metric:
Identifies failures



Evaluation:
Tiny/Small/Std. ϵ

Statement: "Global gradient bounds provide a superior verification domain for shallow, smooth networks, while relational domains are essential for deep, non-smooth architectures"

Contribution Idea

Algebra $\mathbb{D} = \{a + b\eta\}$. Exact derivative propagation looks like: $f(x+\eta) = f(x) + f'(x)\eta$

- $\rightarrow$ We lift this to **Affine Forms** (Zonotopes) to represent the input region X_0: $\hat{x} = \alpha_0 + \sum_{i=1}^{n} \alpha_i \epsilon_i, \ \epsilon_i \in [-1, 1]$
- $\rightarrow$ Forward Propagation:

$$\hat{y} = f(\hat{x}) = \underbrace{\left(y_0 + \sum y_i \epsilon_i\right)}_{\text{Value Interval}} + \underbrace{\left(d_0 + \sum d_i \epsilon_i\right)}_{\text{Gradient } \nabla f} \eta$$

- $\rightarrow$ Lipschitz Bound:

$$K = \sup_{x \in X_0} ||\nabla f(x)|| = |d_0| + \sum |d_i|$$

Contribution Idea

A neuron n is **gradient-unstable** if its gradient interval $[\underline{g}, \overline{g}]$ contains 0

$$\exists x_1, x_2 \in X_0 \text{ s.t. } \operatorname{sign}\left(\frac{\partial f}{\partial n}(x_1)\right) \neq \operatorname{sign}\left(\frac{\partial f}{\partial n}(x_2)\right)$$

Implication

→ Instability $\implies$ High local curvature (non-monotonicity)

→ Linear approximations (DeepPoly) become loose when curvature is high

Evaluation

1. **Sigmoid (Smooth):**
   - $\rightarrow$ Gradient method wins in **16.1%** of Tiny nets.
   - $\rightarrow$ *Reason:* Smooth derivatives allow tight global bounds via Abstract Duals.
2. **ReLU (Non-smooth):**
   - $\rightarrow$ DeepPoly dominates (**100% win rate**).
   - $\rightarrow$ *Reason:* Discontinuous gradients ($\{0, 1\}$) cause the abstract dual interval to become $[0, 1]$ everywhere, destroying precision.
3. **Performance:** Gradient analysis is **2.4x faster** but scales poorly with depth ($K_{total} \approx \prod ||W_i||$).

OCaml code run on MNIST classifiers (Tiny: 2x10, Small: 5x20, Standard: 10x20) with perturbation $\epsilon \in \{0.01 \ldots 0.12\}$.

Thank You!

University of Colorado Boulder

- $\rightarrow$ No “Silver Bullet” domain exists for Neural Network verification
- $\rightarrow$ **Shallow + Smooth** $\rightarrow$ Use **Gradient (Abstract Duals)**
- $\rightarrow$ **Deep + Non-smooth** $\rightarrow$ Use **Relational (DeepPoly)**

Final Takeaway

”Precision in static analysis is not just about the domain’s complexity, but its alignment with the function’s smoothness.”