Provable Certificates for Adversarial Examples: Fitting a Ball in the Union of Polytopes

Matt Jordan, Justin Lewis, Alex Dimakis
UT Austin
Problem Setup

**Complete Verification**: Exactly compute pointwise robustness of piecewise-linear neural nets,

\[ F(x) : \mathbb{R}^n \rightarrow [k] \]

\[ \rho(F, x_0, p) := \inf_{\{x \mid F(x) \neq F(x_0)\}} \|x - x_0\|_p \]
Hardness Results

- Decision Problem: "Does there exist a point \( y \) with 
  \[ \| y - x_0 \|_p \leq \epsilon \] such that \( F(y) \neq F(x_0) \)"
  is NP-Complete [1].
- Approximation is also computationally hard [2].

Previous Work on Complete Verification of Neural Nets

- **SMT Solvers:** ReLUPlex[1], PLANET[3]
- **Mixed Integer Programming:** Many works [4, 5]


Our Approach (Goals):

- Want algorithm that:
  - Exactly computes pointwise robustness
  - Works for arbitrary convex $\ell_p$ norms
  - Returns good lower bound if stopped early
- Game we play:
  Under a moderate time limit, provide the largest provable lower bound.
Roadmap

- Decision regions of ReLU Neural Nets form a *polyhedral complex*.
- Can compute distance to *boundary* of polyhedral complexes.
- Can “warm-start” our algorithm for pointwise robustness with a provable lower bound.
Roadmap

- Decision regions of ReLU Neural Nets form a polyhedral complex.
- Can compute distance to boundary of polyhedral complices.
- Can “warm-start” our algorithm for pointwise robustness with a provable lower bound.
Geometry (Notation)

- Polytope $\mathcal{P} := \{x \mid Ax \leq b\}$
- Facet $\mathcal{F}_i := \{x \mid (Ax \leq b) \land (a_i^T x = b_i)\}$
- Nonconvex Polytope: $\bigcup_{\mathcal{P}_j \in \mathcal{P}} \mathcal{P}_j$
Geometry (Polyhedral Complices)

**Definition:** A nonconvex polytope, $\mathcal{P}$, forms a polyhedral complex if, for every $\mathcal{P}_i, \mathcal{P}_j \in \mathcal{P}$ with nonempty intersection, $\mathcal{P}_i \cap \mathcal{P}_j$ is a face of each.
Input Space

Gibbon

Panda

\[ \{ x \mid A x \leq b \} \]
Input Space

\{x \mid Ax \leq b\}
Input Space

Gibbon

Panda

$x_0$
Geometry of ReLU Nets

- Linear Regions of ReLU Nets are polytopes.
- **Claim 1 (informal):** Any collection of linear regions of a ReLU Nets forms a polyhedral complex
- **Claim 2 (informal):** $D(x_0) := \{x \mid F(x) = F(x_0)\}$ is a polyhedral complex
Geometry of PLNN’s

Claim (informal): Any collection of linear regions of a ReLU Net forms a polyhedral complex
Roadmap

- Decision regions of ReLU Neural Nets form a polyhedral complex.
- Can compute distance to boundary of polyhedral complices.
- Can “warm-start” our algorithm for pointwise robustness with a provable lower bound.
Robustness as a Geometry Problem

Question: “What is the infimal distance to the boundary of a polyhedral complex?”
A Naive Algorithm

- Given polyhedral complex, $\mathcal{P}$, write down convex decomposition of boundary
- Compute projection distance of $x_0$ each convex component of boundary
- Return minimum amongst projections
A Better Algorithm: GeoCert
What we have so far...

**GeoCert:**
- Algorithm to exactly compute pointwise robustness
- Can be stopped early to provide valid lower bound

**Problems:**
- Runtime depends on \(# \text{ iterations}, \text{ cost/iteration}\) 
- Might take a long time to give bound better than SOTA incomplete verifiers
Roadmap

- Decision regions of ReLU Neural Nets form a polyhedral complex.
- Can compute distance to boundary of polyhedral complices.
- Can “warm-start” our algorithm for pointwise robustness with a provable lower bound.
Runtime Improvements (# iterations)

- GeoCert searches *symmetrically*
Runtime Improvements (# iterations)

- Key idea: reorder how we examine facets
Runtime Improvements (# iterations)
Experiments: Completion Time

Binary MNIST, some $\ell_1$-regularization, 70/40 hidden units

<table>
<thead>
<tr>
<th>Method</th>
<th>$\ell_p$</th>
<th>Dist.</th>
<th>Time</th>
<th>Dist.</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast-Lip</td>
<td>$\ell_2$</td>
<td>1.289</td>
<td>0.010</td>
<td>1.319</td>
<td>0.013</td>
</tr>
<tr>
<td>GeoCert</td>
<td>$\ell_\infty$</td>
<td>0.191</td>
<td>1.300</td>
<td>0.187</td>
<td>4.031</td>
</tr>
<tr>
<td>MIP</td>
<td>$\ell_\infty$</td>
<td>0.191</td>
<td>0.947</td>
<td>0.187</td>
<td>0.689</td>
</tr>
<tr>
<td>Fast-Lip</td>
<td>$\ell_2$</td>
<td>1.555</td>
<td>4.789</td>
<td>1.607</td>
<td>21.852</td>
</tr>
<tr>
<td>GeoCert</td>
<td>$\ell_\infty$</td>
<td>1.555</td>
<td>4.030</td>
<td>1.607</td>
<td>5.831</td>
</tr>
</tbody>
</table>
Experiments: Timeout-case

- Binary MNIST, weight decay, MLP [3x20]
- Reported safe bounds after 300s computation time

<table>
<thead>
<tr>
<th>Ex.</th>
<th>Fast-Lip</th>
<th>GeoCert</th>
<th>MIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.782</td>
<td>2.251</td>
<td>2.0</td>
</tr>
<tr>
<td>2</td>
<td>1.319</td>
<td>1.356</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>1.501</td>
<td>1.620</td>
<td>1.0</td>
</tr>
<tr>
<td>4</td>
<td>1.975</td>
<td>2.499</td>
<td>2.0</td>
</tr>
<tr>
<td>5</td>
<td>1.871</td>
<td>2.402</td>
<td>2.0</td>
</tr>
</tbody>
</table>
Summary

- GeoCert provides a novel approach to complete verification of piecewise linear neural networks
- Provides a sequence of provable lower bounds that eventually become tight

Questions? Poster session immediately following
Bonus Slides
Single Polytope Case

What is the infimal distance to the ‘boundary’?

\[
\max \ t \\
\text{s.t. } \sup \ A(x_0 + tv) \leq b \\
\|v\| \leq 1
\]
Graph Search Interpretation

- Maintain ‘frontier facets’
- Maintain ‘seen polytopes’

Algorithm (GeoCert):
- Consider minimal distance ‘frontier facet’
- Return if this facet is ‘boundary’
- Otherwise, look at polytope on other side of this facet. Add new polytope to ‘seen polytopes’
- Add new polytope’s facets to ‘frontier facet’ set
Experiments: # of Linear Regions

<table>
<thead>
<tr>
<th>Potential</th>
<th>Binary MNIST</th>
<th>Full MNIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi_{lip}$</td>
<td>4.2</td>
<td>15.3</td>
</tr>
<tr>
<td>$\phi_p$</td>
<td>5.1</td>
<td>25.6</td>
</tr>
</tbody>
</table>
Runtime Improvements (# iterations)

Consider $\Phi : \mathcal{F} \rightarrow \mathbb{R}$ of the form $\Phi(\mathcal{F}) := \min_{y \in \mathcal{F}} \phi(y)$

If $\phi(y)$ is monotonically increasing in every direction, correctness is maintained.

Can think of $\phi(y)$ as lower bound on distance to decision boundary passing through $y$. 
Runtime Improvements (# iterations)

Idea: Change the ordering of how we examine ‘frontier facets’

Current: $\Phi(\mathcal{F}) := \min_{y \in \mathcal{F}} \| y - x_0 \|_p$

Ideal: $\Phi(\mathcal{F}) := \min_{y \in \mathcal{F}, z \in D(x_0)^c} \| y - x_0 \|_p + \| y - z \|_p$
Runtime Improvements (# iterations)

Leveraging Lipschitz Continuity:
- Let \( f(x) : \mathbb{R}^n \rightarrow \mathbb{R}, \quad F(x) = \text{sign}(f(x)) \)
- Let \( L_q \) be such that for all \( y \in D(x_0) \),
  \[
  |f(x_0) - f(y)| \leq L_q \|x_0 - y\|_p
  \]
- Then \( \phi(y) = \|y - x_0\|_p + \frac{|f(y)|}{L_q} \) maintains correctness
  Lower bound on \( \inf_{z \in D(x_0)} \|y - z\|_p \)
Added Bonus of Lipschitz Potential

- If computed using Fast-Lip/RecurJac, ‘immediately’ outputs nontrivial lower bound
Runtime Improvements (cost/iteration)

- **Vanilla**: computes (# ReLU’s) polyhedral projections per iteration. Many redundant constraints!
- **Improvements**:
  - Domain Restrictions: Images lie in $[0, 1]^n$
  - ReLu Stability: Upper bounds help!