Scalable and Precise Certification of Deep Neural Networks

Martin Vechev

ETH Zurich and LatticeFlow.ai
What we work on

Create **accurate** and **safe** deep learning models

More at: [http://safeai.ethz.ch](http://safeai.ethz.ch)
[certification, provable training, background priors, fairness, etc.]

Today: certification and a bit on certified training
Verification: Prove Absence of Attack

Input image

Rotations
Flipping
Brightening
Noise

All possible perturbations

Verified
All attacks classify to label 8

Not guaranteed
Some attacks may not classify to 8
Verification: Prove Absence of Attack

Challenges:
- Impossible to brute-force all possible perturbations
- Standard solvers (MILP, SMT) do not scale due to non-linearities
Key Technical Insight: AI for AI

Deep neural networks
Affine transforms + specific non-linearities

Abstract Interpretation
Scalable and precise numerical domains
Certifying AI using Abstract Interpretation

AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation, Gehr et al., 2018

Inputs:
$x_0 = 0$
$x_1 = 0.975 + 0.025\epsilon_1$
$x_2 = 0.125$
$...$
$x_{783} = 0.938 + 0.062\epsilon_{783}$
$\forall i. \epsilon_i \in [-1,1]$

Abstract layer transformer

Captures all vectors after first layer

Output constraint $\varphi_n$
$x_0 = 0$
$x_1 = 2.60 + 0.015\epsilon_0 + 0.023\epsilon_1 + 5.181\epsilon_2 + ...$
$x_2 = 4.63 - 0.005\epsilon_0 - 0.006\epsilon_1 + 0.023\epsilon_2 + ...$
$...$
$x_9 = 0.12 - 0.125\epsilon_0 + 0.102\epsilon_1 + 3.012\epsilon_2 + ...$
$\forall i. \epsilon_i \in [-1,1]$

Captures all possible perturbations

Captures all possible outputs

Label $i$ is possible iff: $\varphi_n \cap \{\forall j. x_i \geq x_j\} \neq \bot$
Abstraction
(convex shape)

\[ x_1 \geq z_1 + z_2 \]
\[ x_1 \leq z_1 + z_3 \]

\[ x_2 \geq z_1 - z_2 \]
\[ x_2 \leq z_1 - z_3 \]

\[ y_1 = \max(0, x_1) \]
\[ y_2 = \max(0, x_2) \]

ReLU layer

\[ y_1 \]
\[ y_2 \]
Single Neuron Abstract Transformer

**Abstraction** (convex shape)

- $x_1 \geq z_1 + z_2$
- $x_1 \leq z_1 + z_3$
- $x_2 \geq z_1 - z_2$
- $x_2 \leq z_1 - z_3$

**ReLU layer**

- $y_1 = \max(0, x_1)$
- $y_2 = \max(0, x_2)$

**Constraints**

- $y_1 \geq f_1(x_1)$
- $y_1 \leq g_1(x_1)$
- $x_1 \geq z_1 + z_2$
- $x_1 \leq z_1 + z_3$
- $x_2 \geq z_1 - z_2$
- $x_2 \leq z_1 - z_3$

**Computed only based on $x_1$**
Single Neuron Abstract Transformer

Abstraction
(convex shape)

\[
x_1 \geq z_1 + z_2
\]
\[
x_1 \leq z_1 + z_3
\]

\[
\cdots
\]

\[
x_1
\]

\[
y_1 = \max(0, x_1)
\]

\[
\cdots
\]

\[
y_1
\]

Abstraction $\psi$
(convex shape)

\[
y_1 \geq f_1(x_1)
\]
\[
y_1 \leq g_1(x_1)
\]

\[
\cdots
\]

\[
\text{computed only based on } x_1
\]

ReLU layer

\[
x_2 \geq z_1 - z_2
\]
\[
x_2 \leq z_1 - z_3
\]

\[
\cdots
\]

\[
x_2
\]

\[
y_2 = \max(0, x_2)
\]

\[
\cdots
\]

\[
y_2
\]

\[
y_2 \geq f_2(x_2)
\]
\[
y_2 \leq g_2(x_2)
\]

\[
\cdots
\]

\[
\text{computed only based on } x_2
\]
Single Neuron Abstract Transformer

**Abstraction** (convex shape)

\[
x_1 \geq z_1 + z_2 \\
x_1 \leq z_1 + z_3 \\
\vdots
\]

\[x_1\]

\[y_1 = \max(0, x_1)\]

\[y_1 \geq f_1(x_1)\]

\[y_1 \leq g_1(x_1)\]

**ReLU layer**

\[x_2 \geq z_1 - z_2\]

\[x_2 \leq z_1 - z_3\]

\[y_2 = \max(0, x_2)\]

**Abstraction \(\psi\) (convex shape)**

\[y_2 \geq f_2(x_2)\]

\[y_2 \leq g_2(x_2)\]

\[\vdots\]

\[y_1\]

\[y_2\]

\[\text{computed only based on } x_1\]

\[\text{computed only based on } x_2\]
The single-neuron abstract effect of ReLU: $y = \max(0, x)$
The single-neuron abstract effect of ReLU: $y = \max(0, x)$

Optimal: triangle

[R. Ehlers, ATVA’17]
The single-neuron abstract effect of ReLU: $y = \max(0, x)$

**Optimal: triangle**
[R. Ehlers, ATVA’17]

**Box**

**DeepZono**
[Singh et al., NeurIPS’18]
[Wong et al., ICML’18]
[Weng et al., ICML’18]

**DeepPoly/Crown**
[Zhang et al., NeurIPS’18]
[Singh et al., POPL’19]
The single-neuron abstract effect of ReLU: \( y = \max(0, x) \)

Optimal: triangle  
[R. Ehlers, ATVA’17]

Box

DeepZono  
[Singh et al., NeurIPS’18]  
[Wong et al., ICML’18]  
[Weng et al., ICML’18]

DeepPoly/Crown  
[Zhang et al., NeurIPS’18]  
[Singh et al., POPL’19]

DeepPoly/Crown  
[Zhang et al., NeurIPS’18]  
[Singh et al., POPL’19]

A Convex Relaxation Barrier to Tight Robustness Verification of Neural Networks  
Salman et al., NeurIPS’19
Beyond Single Neuron Abstract Transformers

K-ReLU: Beyond the Single Neuron Convex Barrier for Neural Network Certification, NeurIPS’19, Singh et al.

Neurons $x_1$ and $x_2$ are related

Optimal single-neuron result for processing $x_1$ and $x_2$ separately.

Optimal result for processing $x_1$ and $x_2$ jointly.
ERAN: Neural Network Verification Framework

Neural Network

Pre-condition

Post-condition (e.g., robustness)

ERAN verification framework

https://github.com/eth-sri/eran

- Box
- DeepZono [NeurIPS’18]
- DeepPoly [POPL’19]
- GPUPoly
- RefineZono [ICLR’19]: MILP + AI
- k-RELU [NeurIPS’19]

State-of-the-art complete and incomplete verification

Yes

No

Leading verifier in the recent neural network verification competition

https://sites.google.com/view/vnn20/vnncomp
Geometric Verification: example of a pre-condition
Geometric Verification: example of a pre-condition

Rotation by a concrete angle:

\[ R_{30^\circ}(3) \rightarrow 3 \]
Geometric Verification: example of a pre-condition

Rotation by a **concrete angle**:

\[ R_{30^\circ}( \text{3} ) \rightarrow \text{3} \]

Rotation by a **set of angles**:

\[ R_{[0, 30^\circ]}( \text{3} ) \rightarrow \text{3} \]

Over-Approximation of all possible rotations between \([0^\circ, 30^\circ]\)

Can be non-trivial to compute the region

More transformations, compositions of these, in:
Certifying Geometric Robustness of Neural Networks, NeurIPS'19, Balunovic et al.
Thoughts on Neural Network Verification

Specification

So far: norms (e.g., $l_\infty$), only recently geometric and audio (where there is preprocessing and more semantic meaning). We need more semantic specifications.
# Thoughts on Neural Network Verification

<table>
<thead>
<tr>
<th>Specification</th>
<th>So far: norms (e.g., $l_\infty$), only recently geometric and audio (where there is preprocessing and more semantic meaning). We need more semantic specifications.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relaxations</td>
<td>Some good single-neuron convex relaxations, but more custom abstractions are possible (k-ReLU style) to battle precision loss downstream. Need better combinations with MILP solvers and testing methods. Need custom non-convex/non-linear domains.</td>
</tr>
<tr>
<td>Specification</td>
<td>So far: norms (e.g., $l_{\infty}$), only recently geometric and audio (where there is preprocessing and more semantic meaning). We need more semantic specifications.</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Relaxations</td>
<td>Some good single-neuron convex relaxations, but more custom abstractions are possible (k-ReLU style) to battle precision loss downstream. Need better combinations with MILP solvers and testing methods. Need custom non-convex/non-linear domains.</td>
</tr>
<tr>
<td>Network Types</td>
<td>So far: mostly classification. Need more on generative models, object detection, recurrent networks, image segmentation, financial predictions, more semantic tasks.</td>
</tr>
</tbody>
</table>
# Thoughts on Neural Network Verification

<table>
<thead>
<tr>
<th>Specification</th>
<th>So far: norms (e.g., $l_\infty$), only recently geometric and audio (where there is preprocessing and more semantic meaning). We need more semantic specifications.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relaxations</td>
<td>Some good single-neuron convex relaxations, but more custom abstractions are possible (k-ReLU style) to battle precision loss downstream. Need better combinations with MILP solvers and testing methods. Need custom non-convex/non-linear domains.</td>
</tr>
<tr>
<td>Network Types</td>
<td>So far: mostly classification. Need more on generative models, object detection, recurrent networks, image segmentation, financial predictions, more semantic tasks.</td>
</tr>
<tr>
<td>Network Sizes</td>
<td>Scalability of verification is improving, but still works mostly for small-to-medium networks (the deeper the network is, the more precision loss we have). To scale verification further, we need to consider deeper interactions with training and the network architecture itself. This will lead to networks with better accuracy and provable robustness, which is a fundamental problem now.</td>
</tr>
</tbody>
</table>
Using convex relations for certified training
Certified Defenses: AI for provably robust models
Differentiable Abstract Interpretation for Provably Robust Neural Networks, Mirman et al., ICML 2018
http://github.com/eth-sri/diffai

Idea: automatic differentiation on AI

\[ \theta \]

\[ \minimize \quad \rho(\theta) \]

where

\[ \rho(\theta) = \mathbb{E}_{(x,y) \sim D} \left[ \max_{z \in M^b(S(x))} L(\theta, z, y) \right] \]

Find output \( z \) that achieves high loss under abstraction
Certified Defenses: AI for provably robust models
Differentiable Abstract Interpretation for Provably Robust Neural Networks, Mirman et al., ICML 2018
http://github.com/eth-sri/diffai

Idea: automatic differentiation on AI

Various improvements since the original ICML’18 paper, mostly following same recipe

Scalable Verified Training for Provably Robust Image Classification, Gowal et al, ICCV’2019
Improvements: annealing on input, zonotope for last layer, use of standard cross-entropy loss, etc.
A Provable Defense for Deep Residual Networks, Mirman et al., arXiv 2019
Provides a DSL to specify the training regime, can instantiate various other methods
Adversarial Training and Provable Defenses: Bridging the Gap
Balunovic et al., ICLR’20 (oral)
https://github.com/eth-sri/colt

Key challenge:
find point $x_1 \in S_1$ such that loss $L$ in final layer is maximized

Need projection

$\mathbf{L} x_3 = h_3(h_2(x_1))$

Input

$h_1$

$h_2$

$h_3$

$x_1 \in S_1$

$x_2 \in S_2$

$x_3 \in S_3$

$L(x_3, y_{true})$
Current Best result on CIFAR-10 with $l_\infty$ region = 2/255

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Certified Robustness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before:</td>
<td>70.1%</td>
<td>53.9%</td>
</tr>
<tr>
<td>COLT:</td>
<td>78.8%</td>
<td>58.1%</td>
</tr>
</tbody>
</table>

Accuracy is still low
Predict or Certify: Boosting Certified Robustness of Deep Networks via a Compositional Architecture (ICLR’21)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$\epsilon_\infty$</th>
<th>Method</th>
<th>Top-1 Model Accuracy [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>2/255</td>
<td>ACE</td>
<td>91.6 80.0 22.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COLT$^1$</td>
<td>78.4 68.3 60.5</td>
</tr>
<tr>
<td>ImageNet200</td>
<td>1/255</td>
<td>ACE</td>
<td>70.0 60.5 3.1</td>
</tr>
<tr>
<td>TinyImageNet</td>
<td></td>
<td>LiRPA$^2$</td>
<td>27.8 20.5 15.9</td>
</tr>
</tbody>
</table>

ETH Zurich course: Reliable and Interpretable AI

https://www.sri.inf.ethz.ch/teaching/riai2020

lectures, exercises, videos are public

Core course for last 4 years: how to create safe, secure, robust AI systems.

Mostly research from last 2 years:
- adversarial attacks, convex relaxations, certification, adversarial and certified defenses,
- background priors, fairness, smoothing, etc.

200+ ETH MSc/PhD students from CS, math, physics, robotics, statistics.

More here: http://safeai.ethz.ch