Safe and Robust Deep Learning

Gagandeep Singh
PhD Student
Department of Computer Science
SafeAI @ ETH Zurich (safeai.ethz.ch)

Joint work with

- Martin Vechev
- Markus Püschel
- Timon Gehr
- Matthew Mirman
- Mislav Balunovic
- Maximilian Baader
- Petar Tsankov
- Dana Drachsler

Publications:

- S&P’18: AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation
- NeurIPS’18: Fast and Effective Robustness Certification
- POPL’19: An Abstract Domain for Certifying Neural Networks
- ICLR’19: Boosting Robustness Certification of Neural Networks
- ICML’18: Differentiable Abstract Interpretation for Provably Robust Neural Networks
- ICML’19: DL2: Training and Querying Neural Network with Logic

Systems:

- ERAN: Generic neural network verifier
- DiffAI: System for training provably robust networks
- DL2: System for training and querying networks with logical constraints
Deep learning systems

- Self driving cars: https://waymo.com/tech/
- Translation: https://translate.google.com
- Voice assistant: https://www.amazon.com/Amazon-Echo-And-Alexa-Devices
Attacks on deep learning

The self-driving car incorrectly decides to turn right on Input 2 and crashes into the guardrail

The Ensemble model is fooled by the addition of an adversarial distracting sentence in blue.

Adding small noise to the input audio makes the network transcribe any arbitrary phrase

Article: Super Bowl 50
Paragraph: “Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver’s Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.”
Question: “What is the name of the quarterback who was 38 in Super Bowl XXXIII?”
Original Prediction: John Elway
Prediction under adversary: Jeff Dean

Adversarial Examples for Evaluating Reading Comprehension Systems, EMNLP’17
Audio Adversarial Examples: Targeted Attacks on Speech-to-Text, ICML 2018
Attacks based on intensity changes in images

To verify absence of attack:

\[ L_\infty \text{-norm: consider all images } I \text{ in the } \epsilon \text{-ball } B(I_0, \infty)(\epsilon) \text{ around } I_0 \]
Attacks based on geometric transformations

Consider all images $I$ obtained by applying geometric transformations to $I_0$.

To verify absence of attack:

$I = rotate(I_0, -35)$
Attacks based on intensity changes to sound

Consider all signals $s$ in the $\epsilon$-ball $B_{(s_0,\infty)}(\epsilon)$ around $s_0$
Neural network verification: problem statement

Given: Neural Network $f$, 
      Input Region $\mathcal{R}$ 
      Safety Property $\psi$

Prove: $\forall I \in \mathcal{R}$, 
       prove that $f(I)$ satisfies $\psi$

Example networks and regions:

- **Image classification network $f$**
  Region $\mathcal{R}$ based on changes to pixel intensity
  Region $\mathcal{R}$ based on geometric: e.g., rotation

- **Speech recognition network $f$**
  Region $\mathcal{R}$ based on added noise to audio signal

- **Aircraft collision avoidance network $f$**
  Region $\mathcal{R}$ based on input sensor values

Input Region $\mathcal{R}$ can contain an infinite number of inputs, thus enumeration is infeasible
Experimental vs. certified robustness

**Experimental robustness**

- Tries to find violating inputs
- Like testing, no full guarantees
- E.g. Goodfellow 2014, Carlini & Wagner 2016, Madry et al. 2017

**Certified robustness**

- Prove absence of violating inputs
- Actual verification guarantees
- E.g.: Reluplex [2017], Wong et al. 2018, AI2 [2018]

In this talk we will focus on certified robustness
General approaches to network verification

**Complete** verifiers, but suffer from scalability issues:
SMT: Reluplex [CAV’17], MILP: MIPVerify [ICLR’19],
Splitting: Neurify [NeurIPS’18],…

**Incomplete** verifiers, trade-off precision for scalability:
Box/HBox [ICML’18], SDP [ICLR’18], Wong et.al. [ICML’18], FastLin
[ICML’18], Crown [NeurIPS'18],…

Key Challenge: scalable and precise automated verifier
Network verification with ERAN

**Input region**

- Based on **Pixel Intensity** changes
- Based on **Geometric transformations**: vector fields, rotations, etc.
- Based on **Audio processing**
- Possible sensor values

**Aircraft sensors**

**Neural Network**

- Fully connected
- Convolutional
- Residual
- LSTM
- ReLU
- Sigmoid
- Tanh
- Maxpool

**Safety Property**

- Extensible to other verification tasks
- State-of-the-art **complete** and **incomplete** verification
- Sound w.r.t. floating point arithmetic

**ERAN verification framework**

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

- **Box**
- DeepZ [NeurIPS’18]
- DeepPoly [POPL’19]
- GPUPoly [submitted]
- RefineZono [ICLR’19]: MILP + DeepZ
- KPoly [submitted]: MILP + DeepPoly

**Yes**

**No**
Complete and incomplete verification with ERAN

### Faster Complete Verification

<table>
<thead>
<tr>
<th>Aircraft collision avoidance system (ACAS)</th>
<th>Reluplex</th>
<th>Neurify</th>
<th>ERAN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&gt; 32 hours</td>
<td>921 sec</td>
<td>227 sec</td>
</tr>
</tbody>
</table>

### Scalable Incomplete Verification

<table>
<thead>
<tr>
<th>CIFAR10 ResNet-34</th>
<th>$\epsilon$</th>
<th>%verified</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.03</td>
<td>66%</td>
<td>79 sec</td>
</tr>
</tbody>
</table>
Geometric and audio verification with ERAN

### Geometric Verification

Rotation between -30° and 30° on MNIST CNN with 4,804 neurons

<table>
<thead>
<tr>
<th>$\epsilon$</th>
<th>%verified</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>86</td>
<td>10 sec</td>
</tr>
</tbody>
</table>

### Audio Verification

LSTM with 64 hidden neurons

<table>
<thead>
<tr>
<th>$\epsilon$</th>
<th>%verified</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-110 dB</td>
<td>90%</td>
<td>9 sec</td>
</tr>
</tbody>
</table>
We want to prove that $x_{11} > x_{12}$ for all values of $x_1, x_2$ in the input set
Complete verification with solvers often does not scale.

\[
\begin{align*}
\min x_{11} - x_{12} \\
\text{s.t.:} & \quad x_{11} = x_9 + x_{10} + 1, \quad x_{12} = x_{10}, \\
& \quad x_9 = \max(0, x_7), \quad x_{10} = \max(0, x_8), \\
& \quad x_7 = x_5 + x_6, \quad x_8 = x_5 - x_6, \\
& \quad x_5 = \max(0, x_3), \quad x_6 = \max(0, x_4), \\
& \quad x_3 = x_1 + x_2, \quad x_4 = x_1 - x_2, \\
& \quad -1 \leq x_1 \leq 1, \quad -1 \leq x_2 \leq 1.
\end{align*}
\]

Each \( x_j = \max(0, x_i) \) corresponds to \((x_i \leq 0 \text{ and } x_j = 0)\) or \((x_i > 0 \text{ and } x_j = x_i)\).

Solver has to explore two paths per ReLU resulting in exponential number of paths.
Abstract interpretation

An elegant framework for approximating concrete behaviors

Key Concept: Abstract Domain

Abstract element: approximates set of concrete points
Concretization function $\gamma$: concretizes an abstract element to the set of points that it represents.
Abstract transformers: approximate the effect of applying concrete transformers e.g. affine, ReLU

Patrick and Radhia Cousot Inventors

Tradeoff between the precision and the scalability of an abstract domain
Network verification with ERAN: high level idea

Attacker region $L_\infty$ ball with $\epsilon = 0.1$:

\[ x_0 = [0.1, 0.3] \]
\[ x_1 = [0.4, 0.6] \]
\[ x_2 = [0.18, 0.36] \]
\[ \ldots \]
\[ x_{784} = [0.7, 0.9] \]

Output constraint $\varphi_n$:

\[ x_0 = 0 \]
\[ x_1 = 2.60 + 0.015\eta_0 + 0.023\eta_1 + 5.181\eta_2 + \ldots \]
\[ x_2 = 4.63 - 0.005\eta_0 - 0.006\eta_1 + 0.023\eta_2 + \ldots \]
\[ \ldots \]
\[ x_9 = 0.12 - 0.125\eta_0 + 0.102\eta_1 + 3.012\eta_2 + \ldots \]
\[ \forall i. \eta_i \in [0,1] \]
Verification with the Box domain fails as it cannot capture relational information
**DeepPoly approximation [POPL’19]**

**Shape:** associate a lower polyhedral $a_i^\leq$ and an upper polyhedral $a_i^\geq$ constraint with each $x_i$

$$a_i^\leq, a_i^\geq \in \{x \mapsto v + \sum_{j \in [i-1]} w_j \cdot x_j \mid v \in \mathbb{R} \cup \{-\infty, +\infty\}, w \in \mathbb{R}^{i-1}\} \text{ for } i \in [n]$$

**Concretization of abstract element $a$:**

$$\gamma_n(a) = \{x \in \mathbb{R}^n \mid \forall i \in [n]. a_i^\leq(x) \leq x_i \land a_i^\geq(x) \geq x_i\}$$

**Domain invariant:** store auxiliary concrete lower and upper bounds $l_i, u_i$ for each $x_i$

$$\gamma_n(a) \subseteq \times_{i \in [n]} [l_i, u_i]$$

- less precise than Polyhedra, restriction needed to ensure scalability
- captures affine transformation precisely unlike Octagon, TVPI
- custom transformers for ReLU, sigmoid, tanh, and maxpool activations

---

<table>
<thead>
<tr>
<th>Transformer</th>
<th>Polyhedra</th>
<th>Our domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affine</td>
<td>$O(nm^2)$</td>
<td>$O(w_{max}^2 L)$</td>
</tr>
<tr>
<td>ReLU</td>
<td>$O(\exp(n, m))$</td>
<td>$O(1)$</td>
</tr>
</tbody>
</table>
Example: analysis of a toy neural network

1. 4 constraints per neuron
2. Pointwise transformers => parallelizable.
4. Non-linear activations => approximate and minimize the area
\[ x_1 \geq -1, \quad x_3 \geq x_1 + x_2, \]
\[ x_1 \leq 1, \quad x_3 \leq x_1 + x_2, \]
\[ l_1 = -1, \quad l_3 = -2, \]
\[ u_1 = 1 \quad u_3 = 2 \]

\[ [-1, 1] \]

\[ x_1 \quad 1 \quad 0 \quad \text{max}(0, x_3) \quad 1 \quad 0 \quad \text{max}(0, x_7) \quad 1 \quad 0 \quad x_{11} \]

\[ x_2 \quad 1 \quad -1 \quad \text{max}(0, x_4) \quad 1 \quad 1 \quad \text{max}(0, x_8) \quad 1 \quad 0 \quad x_{12} \]

\[ x_2 \geq -1, \quad x_4 \geq x_1 - x_2, \]
\[ x_2 \leq 1, \quad x_4 \leq x_1 - x_2, \]
\[ l_2 = -1, \quad l_4 = -2, \]
\[ u_2 = 1 \quad u_4 = 2 \]
ReLU activation

Pointwise transformer for $x_j := \max(0, x_i)$ that uses $l_i, u_i$

- if $u_i \leq 0$, $a_j^≤ = a_j^≥ = 0, l_j = u_j = 0$,
- if $l_i \geq 0$, $a_j^≤ = a_j^≥ = x_i, l_j = l_i, u_j = u_i$,
- if $l_i < 0$ and $u_i > 0$

choose (b) or (c) depending on the area

$$\begin{align*}
\langle x_5 \geq 0, \\
x_5 &\leq 0.5 \cdot x_3 + 1, \\
l_5 &= 0, \\
u_5 &= 2 \rangle
\end{align*}$$

$$\begin{align*}
\langle x_6 \geq 0, \\
x_6 &\leq 0.5 \cdot x_4 + 1, \\
l_6 &= 0, \\
u_6 &= 2 \rangle
\end{align*}$$
Affine transformation after ReLU

\[ x_7 \geq x_5 + x_6, \]
\[ x_7 \leq x_5 + x_6, \]
\[ l_7 = 0, \]
\[ u_7 = 4 \]

Imprecise upper bound \( u_7 \) by substituting \( u_5, u_6 \) for \( x_5 \) and \( x_6 \) in \( a_7 \).
Backsubstitution

\[
\langle x_7 \geq 0, \\
x_7 \leq 0.5 \cdot x_3 + 0.5 \cdot x_4 + 2, \\
l_7 = ?, \\
u_7 = ? \rangle
\]

\[
\langle x_6 \geq 0, \\
x_6 \leq 0.5 \cdot x_4 + 1, \\
l_6 = 0, \\
u_6 = 2 \rangle
\]
\[ \langle x_1 \geq -1, \quad x_1 \leq 1, \quad l_1 = -1, \quad u_1 = 1 \rangle \]

\[ \langle x_5 \geq 0, \quad x_5 \leq 0.5 \cdot x_3 + 1, \quad l_5 = 0, \quad u_5 = 2 \rangle \]

\[ \langle x_7 \geq 0, \quad x_7 \leq x_1 + 2, \quad l_7 = 0, \quad u_7 = 3 \rangle \]

\[ \langle x_2 \geq -1, \quad x_2 \leq 1, \quad l_2 = -1, \quad u_2 = 1 \rangle \]

\[ \langle x_6 \geq 0, \quad x_6 \leq 0.5 \cdot x_4 + 1, \quad l_6 = 0, \quad u_6 = 2 \rangle \]

Affine transformation with backsubstitution is pointwise, complexity: \( O(w^2_{\text{max}} L) \)
\[
\begin{align*}
x_1 \geq -1, & \quad x_3 \geq x_1 + x_2, \quad x_5 \geq 0, \\
x_1 \leq 1, & \quad x_3 \leq x_1 + x_2, \quad x_5 \leq 0.5 \cdot x_3 + 1, \\
l_1 = -1, & \quad u_1 = 1, \quad u_3 = 2, \quad u_5 = 2 \\
u_2 = 1. & \quad u_4 = 2, \quad u_6 = 2 \\
\end{align*}
\]

\[
\begin{align*}
x_7 \geq x_5 + x_6, & \quad x_9 \geq x_7, \quad x_{11} \geq x_9 + x_{10} + 1, \\
x_7 \leq x_5 + x_6, & \quad x_9 \leq x_7, \quad x_{11} \leq x_9 + x_{10} + 1, \\
l_7 = 0, & \quad l_9 = 0, \quad l_{11} = 1, \\
u_7 = 3, & \quad u_9 = 3, \quad u_{11} = 5.5 \\
u_8 = 2. & \quad u_{10} = 2, \quad u_{12} = 2 \\
\end{align*}
\]
Checking for robustness

Prove $x_{11} - x_{12} > 0$ for all inputs in $[-1,1] \times [-1,1]$

\[
\begin{align*}
\langle x_{11} \geq x_9 + x_{10} + 1, \\
x_{11} \leq x_9 + x_{10} + 1, \\
l_{11} = 1, \\
u_{11} = 5.5 \rangle \\
\langle x_{12} \geq x_{10}, \\
x_{12} \leq x_{10}, \\
l_{12} = 0, \\
u_{12} = 0 \rangle
\end{align*}
\]

Computing lower bound for $x_{11} - x_{12}$ using $l_{11}, u_{12}$ gives -1 which is an imprecise result.

With backsubstitution, one gets 1 as the lower bound for $x_{11} - x_{12}$, proving robustness.
Abstract interpretation + solvers

Key Idea: refine abstract interpretation results by calling the solver

- Refine neuron bounds before ReLU transformer is applied => less area

\[ l'_8 := \min x_8 \]

\[ s.t.: x_8 = x_5 - x_6, \]

\[ x_5 = \max(0, x_3), \quad x_6 = \max(0, x_4), \]

\[ x_3 = x_1 + x_2, \quad x_4 = x_1 - x_2, \]

\[ -1 \leq x_1 \leq 1, \quad -1 \leq x_2 \leq 1. \]
Verification against geometric attacks

\[ I_0 \] Rotate \( I_0 \) between \(-5^\circ\) and \(+5^\circ\)

\[ I_0 \] Rotate \( I_0 \) between \(-5^\circ\) and \(0^\circ\)

\[ I_0 \] Rotate \( I_0 \) between \(0^\circ\) and \(+5^\circ\)

Sampling + Lipschitz optimization

\[ P(\mathcal{R}) \]

ERAN

ERAN

ERAN
## Medium sized benchmarks

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>Type</th>
<th>#Neurons</th>
<th>#Layers</th>
<th>Defense</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST</td>
<td>6 × 100</td>
<td>feedforward</td>
<td>610</td>
<td>6</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>6 × 200</td>
<td>feedforward</td>
<td>1,210</td>
<td>6</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>9 × 200</td>
<td>feedforward</td>
<td>1,810</td>
<td>9</td>
<td>None</td>
</tr>
<tr>
<td>ConvSmall</td>
<td></td>
<td>convolutional</td>
<td>3,604</td>
<td>3</td>
<td>DiffAI</td>
</tr>
<tr>
<td>ConvBig</td>
<td></td>
<td>convolutional</td>
<td>34,688</td>
<td>6</td>
<td>DiffAI</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>ConvSmall</td>
<td>convolutional</td>
<td>4,852</td>
<td>3</td>
<td>Wong et al.</td>
</tr>
<tr>
<td></td>
<td>ConvBig</td>
<td>convolutional</td>
<td>62,464</td>
<td>6</td>
<td>PGD</td>
</tr>
</tbody>
</table>
Results on medium benchmarks (100 test images)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>#correct</th>
<th>$\epsilon$</th>
<th>DeepPoly</th>
<th>kPoly</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>%</td>
<td>time(s)</td>
</tr>
<tr>
<td>MNIST</td>
<td>6 × 100</td>
<td>99</td>
<td>0.026</td>
<td>21</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>6 × 200</td>
<td>99</td>
<td>0.015</td>
<td>32</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>9 × 200</td>
<td>97</td>
<td>0.015</td>
<td>29</td>
<td>0.9</td>
</tr>
<tr>
<td>ConvSmall</td>
<td>100</td>
<td></td>
<td>0.12</td>
<td>13</td>
<td>6.0</td>
</tr>
<tr>
<td>ConvBig</td>
<td>100</td>
<td></td>
<td>0.3</td>
<td>93</td>
<td>12.3</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>ConvSmall</td>
<td>38</td>
<td>0.03</td>
<td>35</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>ConvBig</td>
<td>65</td>
<td>0.008</td>
<td>39</td>
<td>49</td>
</tr>
</tbody>
</table>
# Large benchmarks

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>Type</th>
<th>#Neurons</th>
<th>#Layers</th>
<th>Defense</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR10</td>
<td>ResNetTiny</td>
<td>residual</td>
<td>311K</td>
<td>12</td>
<td>PGD</td>
</tr>
<tr>
<td></td>
<td>ResNet18</td>
<td>residual</td>
<td>558K</td>
<td>18</td>
<td>PGD</td>
</tr>
<tr>
<td></td>
<td>ResNetTiny</td>
<td>residual</td>
<td>311K</td>
<td>12</td>
<td>DiffAI</td>
</tr>
<tr>
<td></td>
<td>SkipNet18</td>
<td>residual</td>
<td>558K</td>
<td>18</td>
<td>DiffAI</td>
</tr>
<tr>
<td></td>
<td>ResNet18</td>
<td>residual</td>
<td>558K</td>
<td>18</td>
<td>DiffAI</td>
</tr>
<tr>
<td></td>
<td>ResNet34</td>
<td>residual</td>
<td>967K</td>
<td>34</td>
<td>DiffAI</td>
</tr>
</tbody>
</table>
## Results on large benchmarks (500 test images)

<table>
<thead>
<tr>
<th>Model</th>
<th>Training</th>
<th>#correct</th>
<th>$\epsilon$</th>
<th>Hbox[ICML’18]</th>
<th>GPUPoly</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>%✅</td>
<td>time(s)</td>
</tr>
<tr>
<td>ResNetTiny</td>
<td>PGD</td>
<td>391</td>
<td>0.002</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>ResNet18</td>
<td>PGD</td>
<td>419</td>
<td>0.002</td>
<td>0</td>
<td>6.8</td>
</tr>
<tr>
<td>ResNetTiny</td>
<td>DiffAl</td>
<td>184</td>
<td>0.03</td>
<td>118</td>
<td>0.3</td>
</tr>
<tr>
<td>SkipNet18</td>
<td>DiffAl</td>
<td>168</td>
<td>0.03</td>
<td>130</td>
<td>6.1</td>
</tr>
<tr>
<td>ResNet18</td>
<td>DiffAl</td>
<td>193</td>
<td>0.03</td>
<td>129</td>
<td>6.3</td>
</tr>
<tr>
<td>ResNet34</td>
<td>DiffAl</td>
<td>174</td>
<td>0.03</td>
<td>103</td>
<td>16</td>
</tr>
</tbody>
</table>
Network verification with ERAN

**Input region**
- Based on **Pixel Intensity** changes
- Based on **Geometric** transformations: vector fields, rotations, etc.
- Based on Audio processing
- Possible sensor values

**Neural Network**
- Fully connected
- Convolutional
- Residual
- LSTM
- ReLU
- Sigmoid
- Tanh
- Maxpool

**Safety Property**

**ERAN verification framework**
https://github.com/eth-sri/eran

- Box
- DeepZ [NeurIPS’18]
- DeepPoly [POPL’19]
- GPUPoly [submitted]
- RefineZono [ICLR’19]: MILP + DeepZ
- K-Poly [submitted]: MILP + DeepPoly

**Extensible to other verification tasks**
- State-of-the-art **complete** and **incomplete** verification
- Sound w.r.t. floating point arithmetic

**Aircraft sensors**

Yes

No
In-progress work in verification/training (sample)

Verification Precision: More precise convex relaxations by considering multiple ReLUs

Verification Scalability: GPU-based custom abstract domains for handling large nets

Theory: Proof on Existence of Accurate and Provable Networks with Box

Provable Training: Procedure for training Provable and Accurate Networks

Applications: e.g., reinforcement learning, geometric, audio, sensors
Attacks on Deep Learning

The self-driving car incorrectly decides to turn right on Input 2 and crashes into the guardrail.

The Ensemble model is fooled by the addition of an adversarial distracting sentence in blue.

Adding small noise to the input audio makes the network transcribe any arbitrary phrase.


Adversarial Examples for Evaluating Reading Comprehension Systems, EMNLP’17

Audio Adversarial Examples: Targeted Attacks on Speech-to-Text, ICML 2018

Neural Network Verification: Problem statement

Given: Neural Network $f$, Input Region $\mathcal{R}$
Safety Property $\psi$

Prove: $\forall I \in \mathcal{R}$, prove that $f(I)$ satisfies $\psi$

Example networks and regions:
- Image classification network $f$
- Region $\mathcal{R}$ based on changes to pixel intensity
- Region $\mathcal{R}$ based on geometric e.g. rotation
- Speech recognition network $f$
- Region $\mathcal{R}$ based on added noise to audio signal
- Aircraft collision avoidance network $f$
- Region $\mathcal{R}$ based on input sensor values

Input Region $\mathcal{R}$ can contain an infinite number of inputs, thus enumeration is infeasible

Network Verification with ERAN

Input region
- Based on Pixel Intensity changes
- Based on Geometric transformations: vector fields, rotations, etc.
- Based on Audio processing
- Possible sensor values

Neural Network
- Fully connected
- Convolutional
- Residual
- LSTM
- ReLU
- Sigmoid
- Tanh
- Maxpool

Safety Property

ERAN verification framework
https://github.com/eth-sri/eran

Box
DeepZ [NeurIPS’18]
DeepPoly [POPL’19]
GPUPoly [submitted]
RefineZooe [ICLR’19]: MILP + DeepZ
K-Poly [submitted]: MILP + DeepPoly

Extensible to other verification tasks
State-of-the-art complete and incomplete verification
Sound w.r.t. floating point arithmetic

Complete and Incomplete Verification with ERAN

Faster Complete Verification
- Aircraft collision avoidance system (ACAS)
- Reluplex
- Neurify
- ERAN
- $> 32$ hours
- 921 sec
- 227 sec

Scalable Incomplete Verification
- CIFAR10 ResNet-34
- $\epsilon$
- $\%$ verified
- Time (s)
- 0.03
- 66%
- 79 sec
Using AI to Train Robust Deep Learning

Idea: define abstract loss to include AI result, apply automatic differentiation on AI

<table>
<thead>
<tr>
<th>Training Method</th>
<th>Accuracy %</th>
<th>Certified %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>98.4</td>
<td>2.8</td>
</tr>
<tr>
<td>Madry et al.</td>
<td>98.8</td>
<td>11.2</td>
</tr>
<tr>
<td>DiffAI (our method)</td>
<td>99.0</td>
<td>96.4</td>
</tr>
</tbody>
</table>

Convolutional Network with 124,000 neurons, $L_\infty$ with $\epsilon = 0.1$

Differentiable Abstract Interpretation for Provably Robust Neural Networks
ICML 2018
(Matthew Mirman, Timon Gehr, Martin Vechev)
Released Frameworks

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

Framework for verification of deep neural networks, supports various numerical domains, floating-point sound, different perturbations, largest dataset to date: 50+ networks. Currently the most scalable and precise verifier.

http://github.com/eth-sri/diffai

Framework for training deep neural nets to be more robust using symbolic analysis. Different defenses and attacks (PGD, PGD + DiffAI). Currently the most scalable framework.
Challenges and Open Problems

**Specification**
- Typically, some norm: $L_0$, $L_1$, $L_\infty$
- How about geometric changes? Distributions?
- Guarantees: unbounded number of images?

**Verification**
- What is a good abstraction?
- How do we leverage testing results?
- How to battle approximation loss downstream?
- Creative combinations with complete methods?

**Networks**
- Classification? Reinforcement Learning?
- Regression? Recurrent?
- Combinations of models?

**Trade-offs**
- Accuracy vs. Robustness?
- Provability vs. Accuracy?
Input region $L_\infty(I_0, \epsilon)$

All images $I$ where the intensity at each pixel differs from the intensity at the corresponding pixel in $I_0$ by $\leq \epsilon$
Input regions

\[ I_0 \]

\[ I \in L_\infty(I_0, \varepsilon) \]

\[ I \in \text{Rotate}(I_0, \varepsilon, \alpha, \beta) \]
Input region $\text{Rotate}(I_0, \epsilon, \alpha, \beta)$

All images $I$ which are obtained by rotation each image in $L_\infty(I_0, \epsilon)$ by an angle between $\alpha$ and $\beta$ using bilinear interpolation.