Safe and Robust Deep Learning

Gagandeep Singh

PhD Student

Department of Computer Science





SafeAI @ ETH Zurich (safeai.ethz.ch)

Joint work with





Martin Vechev

Timon Püschel Gehr



Matthew Mirman



Balunovic





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:

Baader

ERAN: Generic neural network verifier

DiffAl: System for training provably robust networks

DL2: System for training and querying networks with logical constraints

Deep learning systems



https://waymo.com/tech/

Translation

 \equiv **Google** Translate

☆ _A Text	Documents
---------------------	-----------



https://translate.google.com

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



(a) Input 1



(b) Input 2 (darker version of 1)

DeepXplore:Automated Whitebox Testing of Deep Learning Systems, SOSP'17

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

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

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



"it was the best of times. it was the worst of times"





"it is a truth universally acknowledged that a single"

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

Attacks based on intensity changes in images



To verify absence of attack:

 L_{∞} -norm: consider all images I in the ϵ -ball $\mathcal{B}_{(I_0,\infty)}(\epsilon)$ around I_0

Attacks based on geometric transformations



To verify absence of attack:

Consider all images I obtained by applying geometric transformations to I_0

Attacks based on intensity changes to sound



To verify absence of attack:

Consider all signals s in the ϵ -ball $\mathcal{B}_{(s_0,\infty)}(\epsilon)$ around s_0

Neural network verification: problem statement

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

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

Example networks and regions:

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

Speech recognition network fRegion \mathcal{R} based on added noise to audio signal

Aircraft collision avoidance network fRegion \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, Al2 [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



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



Geometric and audio verification with ERAN

Geometric Verification

Rotation betweer CNN wit	n -30° and 30° or th 4,804 neurons	n MNIST
ϵ	%verified	Time(s)
0.001	86	10 sec

Audio Verification

LSTM with 64 hidden neurons				
ϵ	%verified	Time (s)		
-110 dB	90%	9 sec		

Example: analysis of a toy neural network



We want to prove that $x_{11} > x_{12}$ for all values of x_1, x_2 in the input set



 $min \ x_{11} - x_{12}$

 $\begin{array}{l} s.t.: \ x_{11} = x_9 + x_{10} + 1, \ x_{12} = x_{10}, \\ x_9 = \max(0, x_7), \ x_{10} = \max(0, x_8), \\ x_7 = x_5 + x_6, \ x_8 = x_5 - x_6, \\ x_5 = \max(0, x_3), \ x_6 = \max(0, x_4), \\ x_3 = x_1 + x_2, \ x_4 = x_1 - x_2, \\ -1 \leq x_1 \leq 1, \ -1 \leq x_2 \leq 1. \end{array}$

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

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

Complete verification with solvers often does not scale

Abstract interpretation



Patrick and Radhia Cousot Inventors An elegant framework for approximating concrete behaviors

Key Concept: Abstract Domain

Abstract element: approximates set of concrete points Concretization function γ : 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

Tradeoff between the precision and the scalability of an abstract domain

Network verification with ERAN: high level idea



Box approximation (scalable but imprecise)



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

n: #neurons, m: #constraints

 w_{max} : max #neurons in a layer, L:# layers

Transformer	Polyhedra	Our domain
Affine	$0(nm^2)$	$O(w_{max}^2L)$
ReLU	$O(\exp(n,m))$	0(1)

Example: analysis of a toy neural network



- I.4 constraints per neuron
- 2. Pointwise transformers => parallelizable.
- 3. Backsubstitution => helps precision.
- 4. Non-linear activations => approximate and minimize the area

20



ReLU activation

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

$$if \ u_i \le 0, a_j^{\le} = a_j^{\ge} = 0, l_j = u_j = 0, \\ if \ l_i \ge 0, a_j^{\le} = a_j^{\ge} = x_i, l_j = l_i, u_j = u_i, \\ if \ l_i < 0 \ and \ u_i > 0$$

$$egin{aligned} &\langle x_5 \geq 0, \ &x_5 \leq 0.5 \cdot x_3 + 1, \ &l_5 = 0, \ &u_5 = 2
angle \end{aligned}$$







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

Constant runtime

Affine transformation after ReLU



Imprecise upper bound u_7 by substituting u_5 , u_6 for x_5 and x_6 in a_7^2 and a_8^2 a

Backsubstitution





Affine transformation with backsubstitution is pointwise, complexity: $O(w_{max}^2 L)^{25}$



Checking for robustness

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

$$egin{aligned} &\langle x_{11} \geq x_9 + x_{10} + 1, & \langle x_{12} \geq x_{10}, \ &x_{11} \leq x_9 + x_{10} + 1, & x_{12} \leq x_{10}, \ &l_{11} = 1, & l_{12} = 0, \ &u_{11} = 5.5
angle & u_{12} = 0
angle \end{aligned}$$

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



Verification against geometric attacks



Medium sized benchmarks

Dataset	Model	Туре	#Neurons	#Layers	Defense
MNIST	6×100	feedforward	610	6	None
	6 × 200	feedforward	1,210	6	None
	9 × 200	feedforward	1,810	9	None
	ConvSmall	convolutional	3,604	3	DiffAl
	ConvBig	convolutional	34,688	6	DiffAl
CIFAR I 0	ConvSmall	convolutional	4,852	3	Wong et al.
	ConvBig	convolutional	62,464	6	PGD

Results on medium benchmarks (100 test images)

Dataset	Model	#correct	ϵ	Dee	pPoly	k	Poly
				%√	time(s)	%√	time(s)
MNIST	6×100	99	0.026	21	0.3	44	151
	6 × 200	99	0.015	32	0.5	56	387
	9 × 200	97	0.015	29	0.9	54	1040
	ConvSmall	100	0.12	13	6.0	28	1018
	ConvBig	100	0.3	93	12.3	93	286
CIFAR I 0	ConvSmall	38	0.03	35	0.4	35	1.4
	ConvBig	65	0.008	39	49	40	2882

Large benchmarks

Dataset	Model	Туре	#Neurons	#Layers	Defense
CIFAR 10	ResNetTiny	residual	311K	12	PGD
	ResNet18	residual	558K	18	PGD
	ResNetTiny	residual	311K	12	DiffAl
	SkipNet18	residual	558K	18	DiffAl
	ResNet18	residual	558K	18	DiffAl
	ResNet34	residual	967K	34	DiffAl

Results on large benchmarks (500 test images)

Model	Training	#correct	E	Hbox[IC	CML'18]	GP	UPoly
				% 🗸	time(s)	%√	time(s)
ResNetTiny	PGD	391	0.002	0	0.3	322	30
ResNet18	PGD	419	0.002	0	6.8	324	I 400
ResNetTiny	DiffAl	184	0.03	118	0.3	127	7.6
SkipNet18	DiffAl	168	0.03	130	6.1	140	57
ResNet18	DiffAl	193	0.03	129	6.3	139	37
ResNet34	DiffAl	174	0.03	103	16	114	79

Network verification with ERAN



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



Neural Network Verification: Problem statement

Given: Neural Network f, Input Region \mathcal{R} Safety Property ψ Prove: $\forall I \in \mathcal{R}$.

prove that f(I) satisfies ψ

Example networks and regions:

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

Speech recognition network fRegion \mathcal{R} based on added noise to audio signal

Aircraft collision avoidance network fRegion $\mathcal R$ based on input sensor values

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

Network Verification with ERAN



Complete and Incomplete Verification with ERAN

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

Yes

No

Scalable Incomplete Verification

	CIFAR10 ResNet-	34
ε	%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

Training Method	Accuracy %	Certified %
Baseline	98.4	2.8
Madry et al.	98.8	11.2
DiffAI (our method)	99.0	96.4

Convolutional Network with 124,000 neurons, L_{∞} 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



Verification



Networks



Trade-offs

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

Classification? Reinforcement Learning? Regression? Recurrent? Combinations of models?

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



Input region $Rotate(I_0, \epsilon, \alpha, \beta)$

All images I which are obtained by rotation each image in $L_{\infty}(I_0, \epsilon)$ by an angle between α and β using bilinear interpolation

Original

