Adversarial Machine Learning (AML): Research Trends and Directions

Somesh Jha

June 26 (Wed), 2019

CSF, New Jersey

Thanks to Nicolas Papernot, Ian Goodfellow and Jerry Zhu

for some slides.

Joint work with Tommaso Dreossi and Sanjit Seshia (Berkeley)
My First CSF..

- 14th Computer Security Foundations Workshop (CSFW), 2001
  - Cape Breton, Nova Scotia, Canada, 11-13 June 2001
Plan

• Part I [Adversarial ML]
  • Different types of attacks
  • Test-time attacks
  • Defenses
  • Theoretical explorations

• Part II [Opportunities in FM]
  • Opportunities for formal researcher
  • Focus on lot of work by Tommaso and Sanjit
Announcements/Caveats

- Please ask questions during the talk
  - If we don’t finish, fine😊

- More slides than I can cover
  - Lot of skipping will be going on

- Fast moving area
  - Apologies if I don’t mention your paper
  - ~450 submissions on this topic at ICML 2019
Machine learning brings social disruption at scale

Healthcare
Source: Peng and Gulshan (2017)

Energy
Source: Deepmind

Transportation
Source: Google

Education
Source: Gradescope
ML reached “human-level performance” on many IID tasks circa 2013

...recognizing objects and faces....

(Szegedy et al, 2014)

(Taigmen et al, 2013)

...solving CAPTCHAS and reading addresses...

(Goodfellow et al, 2013)

(Goodfellow et al, 2013)
ML beating doctors😊

• NOVEMBER 15, 2017
  • Stanford algorithm can diagnose pneumonia better than radiologists

• April 14, 2017
  • Self-taught artificial intelligence beats doctors at predicting heart attacks

• ....
Machine learning is deployed in adversarial settings

Microsoft’s Tay chatbot

*Training* data poisoning

YouTube filtering

Content evades detection at *inference*
ML in CPS

Artificial Intelligence based systems for automotive

Many Safety-Critical Systems
I.I.D. Machine Learning

Train | Test

I: Independent
I: Identically
D: Distributed

All train and test examples drawn independently from same distribution
Security Requires Moving Beyond I.I.D.

- Not identical: attackers can use unusual inputs

  (Eykholt et al, CVPR 2017)

- Not independent: attacker can repeatedly send a single mistake (“test set attack”)

Adversarial Learning is not new!!

• **Lowd:** I spent the summer of 2004 at Microsoft Research working with Chris Meek on the problem of spam.
  • We looked at a common technique spammers use to defeat filters: adding "good words" to their emails.
  • We developed techniques for evaluating the robustness of spam filters, as well as a theoretical framework for the general problem of learning to defeat a classifier (*Lowd and Meek, 2005*)

• But...
  • New resurgence in ML and hence new problems
  • Lot of new theoretical techniques being developed
    • High dimensional robust statistics, robust optimization, ...
Attacks on the machine learning pipeline

- Training data poisoning
- Training set
- Test input
- Adversarial Examples
- Test output
- Model theft
- Parameter Tampering Attack
- Learned Parameters
- Learning algorithm
ML (Basics, Supervised Learning)

• Entities
  • *(Sample Space)* $Z = X \times Y$
    • (data, label) $(x, y)$
  
  • *(Distribution over Z)* $D$
  
  • *(Hypothesis Space)* $H$
  
  • *(loss function)* $l: (H \times Z) \rightarrow R$
ML (Basics)

• **Learner’s problem**
  - Find \( w \in H \) that minimizes
    - \( E_{z \sim D} l(w, z) + \lambda R(w) \)
    - \( \frac{1}{m} \sum_{i=1}^{m} l(w, (x_i, y_i)) + \lambda R(w) \)

• **Sample set** \( S = \{(x_1, y_1), ..., (x_m, y_m)\} \)

• **Stochastic Gradient Descent (SGD)**
  - (iteration) \( w[t + 1] = w[t] - \eta_t l'(w[t], (x_{i_t}, y_{i_t})) \)
  - (learning rate) \( \eta_t \)
  - ...

ML (Basics)

• After Training
  • $F_w: X \rightarrow Y$

  • $F_w(x) = \arg\max_{y \in Y} s(F_w)(x)$

  • (softmax layer) $s(F_w)$

  • Sometimes we will write $F_w$ simply as $F$
    • $w$ will be implicit
Training Time Attack
Attacks on the machine learning pipeline

- **Training data**
- **Training set poisoning**
- **Test input**
- **Adversarial Examples**
- **Test output**
- **Model theft**

- **Learning algorithm**
- **Parameter Tampering Attack**
- **Learned Parameters**
Lake Mendota Ice Days
Poisoning Attacks

![Graph showing the effect of poisoning attacks on ice days, with original and norm attack models.](image)
Formalization

• Alice picks a data set $S$ of size $m$
• Alice gives the data set to Bob
• Bob picks
  • $\epsilon m$ points $S^B$
  • Gives the data set $S \cup S^B$ back to Alice
  • Or could replace some points in $S$
• Goal of Bob
  • Maximize the error for Alice
• Goal of Alice
  • Get close to learning from clean data
Representative Papers

• *Being Robust (in High Dimensions) Can be Practical*  
  I. Diakonikolas, G. Kamath, D. Kane, J. Li, A. Moitra, A. Stewart  
  ICML 2017

• *Certified Defenses for Data Poisoning Attacks*. Jacob Steinhardt, Pang Wei Koh, Percy Liang. NIPS 2017

• *Explicit defense actions against test-set attacks*. S. Alfeld, X. Zhu, and P. Barford, AAAI 2017

• *Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks*, NIPS 18

• ...
Model Extraction/Theft Attack
Attacks on the machine learning pipeline

- Training data poisoning
- Adversarial Examples
- Parameter Tampering Attack
- Model theft
- Test input output
Model Theft

• **Model theft:** extract model parameters by queries (intellectual property theft)
  • Given a classifier $F$
  • Query $F$ on $q_1, \ldots, q_n$ and learn a classifier $G$
  • $F \approx G$

• Goals: leverage active learning literature to develop new attacks and preventive techniques

• **Paper:** *Stealing Machine Learning Models using Prediction APIs*, Tramer et al., Usenix Security 2016
Fake-News Attacks
Fake News Attacks

Abusive use of machine learning:

Using GANs to generate **fake content** (a.k.a. deep fakes)

*Strong societal implications:*

- elections,
- automated trolling,
- court evidence ...

**Generative media:**

- Video of Obama saying things he never said, ...
- Automated reviews, tweets, comments, indistinguishable from human-generated content
Test-time Attacks
Attacks on the machine learning pipeline

- **Training data**
- **Training set poisoning**
- **Test input**
- **Adversarial Examples**
- **Test output**
- **Model theft**

**Key Points:**
- **Learning algorithm**
- **Learned Parameters**
- **Parameter Tampering Attack**

Diagram showing the flow from input $X$ to output $y$ with various attack points.
Definition

“Adversarial examples are inputs to machine learning models that an attacker has intentionally designed to cause the model to make a mistake”

(Goodfellow et al 2017)
What if the adversary systematically found these inputs?

$x$

- “panda” 57.7% confidence
- “nematode” 8.2% confidence
- “gibbon” 99.3% confidence

$+0.007 \times \text{sign}(\nabla_x J(\theta, x, y))$

Biggio et al., Szegedy et al., Goodfellow et al., Papernot et al.
Good models make surprising mistakes in non-IID setting

“Adversarial examples”

Schoolbus + Perturbation = Ostrich

(rescaled for visualization)

(Szegedy et al, 2013)
Adversarial Examples

88% tabby cat

99% guacamole
Adversarial examples...

... beyond deep learning

Logistic Regression

Nearest Neighbors

Support Vector Machines

Decision Trees

... beyond computer vision

P[X=Malware] = 0.90
P[X=Benign] = 0.10

P[X*=Malware] = 0.10
P[X*=Benign] = 0.90
Formal Definition (Local Robustness)

• Let $O \subseteq X \times X$ be an equivalence relation
  • $(x, x') \in O$ (examples $x$ and $x'$ are “perceived” same)

• Targeted local robustness $TR^O (x, F, t)$
  • $\forall x' : O(x, x') \Rightarrow \neg(F(x') = t)$

• Global targeted robustness predicate/metric $GTR^O (F, t)$
  • $\forall x : TR^O (x, F, t)$
  • $E_{x \sim D} (TR^O (x, F, t))$

• Observation
  • Targeted adversarial examples are counterexamples to $GTR^O (F, t)$
Global Robustness

• **Local robustness predicate** $R^O (x, F)$
  • $\forall x' : O(x, x') \Rightarrow (F(x) = F(x'))$

• **Global robustness predicate/metric** $GR^0 (F)$
  • $\forall x R^0 (x, F')$
  • $E_{\{x \sim D\}} (R^0 (x, F))$

• **Observation**
  • adversarial examples are counterexamples to $GR^0 (F)$
Instantiating the Oracle

• Ideal
  • $O(x, x') = 1$ iff a human perceives $x$ and $x'$ as same images
  • Difficulty:
    • We don’t completely how human perception works 😞

• What researchers actually use
  • $O(x, x') = 1$ iff $x$ and $x'$ are close under some norm
    • $L_\infty$
    • $L_1$
    • $L_p$ ($p \geq 2$)
Threat Model

• White Box
  • Complete access to the classifier $F$

• Black Box
  • Oracle access to the classifier $F$
  • for a data $x$ receive $F(x)$

• Grey Box
  • Black-Box + “some other information”
  • Example: structure of the defense
FGSM (white box, misclassification)

• Take a step in the
  • direction of the gradient of the loss function
  • $\delta = \epsilon \text{sign}(\Delta_x l(w, x, F(x)))$
  • Essentially opposite of what SGD step is doing

• Paper
  • Goodfellow, Shlens, Szegedy. Explaining and harnessing adversarial examples. ICLR 2015
PGD (white box, misclassification)

- \( \text{Proj}_{B(x,\epsilon)}(y) \)
  - Project \( y \) to the ball \( B(x,\epsilon) \)

- Iterate the following step
  - \( x_{k+1} = \text{Proj}_{B(x,\epsilon)}(x_k + \epsilon \text{sign} \Delta_x l(w,x,F(x))) \)

- Intuition:
  - Take a FGSM step, and
  - Project it down to the ball
JSMA (white-box, targeted)
Other Attacks (White-box, targeted)

- Carlini-Wagner (CW)
  - Use optimization engines (i.e. Adam) in a black-box manner

- Athalye-Carlini-Wagner
  - More on this later....
  - Builds on CW
Attacking remotely hosted black-box models

Practical Black-Box Attacks against Machine Learning [AsiaCCS 2017]
Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z.Berkay Celik, and Ananthram Swami
Abstract Algorithm

• Choose $S$ (substitute network)

• Interact with the classifier $F$ in a black-box manner

• Train the substitute network $S$

• Run white-box attack on $S$
Current Black-Box Attacks

• **Black-box Adversarial Attacks with Limited Queries and Information**, Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin, *ICML 2018*


• These are very powerful black-box learner

• **Problem: Use these in verification**
Defense
Robust Defense Has Proved Elusive

• Quote
  • *In a case study, examining noncertified white-box-secure defenses at ICLR 2018, we find obfuscated gradients are a common occurrence, with 7 of 8 defenses relying on obfuscated gradients. Our new attacks successfully circumvent 6 completely and 1 partially.*

• Paper
  • Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples.
  • Anish Athalye, Nicholas Carlini, and David Wagner, *ICML 2018*
Robust Objectives

• Use the following objective
  \[ \min_w E_z \left[ \max_{\{z' \in B(z, \epsilon)\}} l(w, z') \right] \]
  • Outer minimization use SGD
  • Inner maximization use PGD

• A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. ICLR 2018
• A. Sinha, H. Namkoong, and J. Duchi. Certifying Some Distributional Robustness with Principled Adversarial Training. ICLR 2018
Intuition

• $\min_w E_z \left[ \max_{\{z' \in B(z, \epsilon)\}} l(w, z') \right]$

• For simple models (e.g. logistic regression) one can write a formula for the inner max

• But what about DNNs?

• Let $A$ be an algorithm to solve $\max_{\{z' \in B(z, \epsilon)\}} l(w, z')$
  • **Intuition:** An attacker is trying to solve this problem, so use an existing algorithm (e.g. PGD)
Adversarial Training

1. Train the model naturally (the procedure I described first). Let $A$ be the best attack algorithm in the literature (e.g. PGD)

2. Adversarial training for each element $x_i$
   1. $z_i = A(x_i)$ (adversarial example)
   2. Use natural training on $z_i$

*Note:* Using attack technique to make the model more robust

*Analogy:* Counterexample guided re-training (refinement?)
Theoretical Explanations
Some Directions (Representative Papers)

• Are robustness and accuracy at odds?

• Can more data lead to robustness?
  • Analyzing the Robustness of Nearest Neighbors to Adversarial Examples, Yizhen Wang, Somesh Jha, Kamalika Chaudhuri, ICML 2018
  • Adversarially Robust Generalization Requires More Data. Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, Aleksander Mądry, ICLR 2018
    • We show that already in a simple natural data model, the sample complexity of robust learning can be significantly larger than that of "standard" learning.
Some Directions (Contd)

• What if the learner is given exponential time?
  • Adversarial examples from computational constraints. Sébastien Bubeck, Eric Price, Ilya Razenshteyn
  • Computational Limitations in Robust Classification and Win-Win Results, A. Degwekar, P. Nakkiran, V. Vaikuntanathan

• Can limiting the adversary to PPTA help?
  • Adversarially Robust Learning Could Leverage Computational Hardness. S. Garg, S. Jha, S. Mahloujifar, and M. Mahmoody

• Jury is Still Out!!
Verification, Analysis, Testing
Formal Definition

• Let $O \subseteq X \times X$ be a binary oracle
  • $O(x, x') = 1$ (examples $x$ and $x'$ “perceived” same)
  • Otherwise 0 (Examples are “perceived” different)

• Local robustness predicate $R^O(x, F)$
  • $\forall x' : O(x, x') \Rightarrow (F(x) = F(x'))$

• Global robustness predicate $GR^O(F)$
  • $\forall x \ GR^O(F)$

• Observation
  • adversarial examples are counterexamples to $GR^O(F)$
Decision Procedures for Local Robustness

• Decision procedures for verifying local robustness at a point
  • Safety Verification of DNNs, CAV 2017
  • ReLUplex: An Efficient SMT Solver for Verifying DNNs, CAV 2017
  • …

• Great work, but
  • Scalability (see earlier slide)
  • Not coupled with some of the ML techniques being developed

• Problem
  • Can these decision procedures help in adversarial training?
Decision Procedures for Local Robustness
Getting Better….

• Towards Fast Computation of Certified Robustness for ReLU Networks
  • Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Duane Boning, Inderjit S. Dhillon, Luca Daniel, ICML 2018
  • Activation function limited to: \( f(x) = x^+ = \max(0, x) \)
  • Reluplex
  • Based on spectral techniques

• Lot of work in this space….
Analysis/Testing

• DeepXplore, SOSP 17

• Formal Symbolic Analysis of Neural Networks using Symbolic Intervals, Usenix Security 2018

• AI2: Abstract Interpretation of Neural Networks, Oakland 2018

• Problem
  • *Can these techniques help in adversarial training?*
Glaring Omission from AML

• Specification of the system that is using ML
  • Control loop for flying a drone

• Problem
  • *Can we do better if we are more “application aware”?*

• Evidence
    • Tutorial at ICML 2018 by Sanjeev Arora
  • *Towards Verified Artificial Intelligence*, Sanjit A. Seshia, Dorsa Sadigh, S. Shankar Sastry
Automatic Emergency Braking System

• Goal: Brake whenever an obstacle is detected

Theme 1

• We allowed only one kind of transformation
  • Add a vector $\delta$

• Allow richer transformations
  • Relevant to the application
  • Translation, cloudy background, ..... 
  • Paper
    • A Rotation and Translation Suffice: Fooling CNNs with Simple Transformations

• Problem:
  • *Construct adversarial examples given a specification of transformations?*
Semantic Adversarial Analysis and Training

DNN analysis must be more *semantic*

- Semantic modification
- System-level specification
- Semantic (re-)training
- Confidence-based analysis

Non-semantic perturbation (i.e., noise)

Semantic perturbation (i.e., translation)
Theme 2

• Problem:
  • Construct adversarial examples that actually lead to system-level failures?

• We can then use these examples for adversarial training

• More on this later...
Semantic Adversarial Analysis and Training

DNN analysis must be more *semantic*

Example: AEBS Counterexamples?

- Semantic modification
- System-level specification
  - Semantic (re-)training
  - Confidence-based analysis

<table>
<thead>
<tr>
<th>Perception-level spec: “detect cars”</th>
<th>✗</th>
<th>✗</th>
</tr>
</thead>
<tbody>
<tr>
<td>System-level spec: “do not crash”</td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>

Does not affect the system
Semantic Adversarial Analysis and Training

DNN analysis must be more semantic

Example: AEBS
Spec: “do not crash”

- Semantic modification
- System-level specification
- Semantic (re-)training
- Confidence-based analysis

Semantic augmentation

Original Training set + Original Training set VS

Experimental Results

- Augmentation methods comparison

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>0.61</td>
<td>0.74</td>
</tr>
<tr>
<td>Standard augmentation</td>
<td>0.69</td>
<td>0.80</td>
</tr>
<tr>
<td>Random</td>
<td>0.76</td>
<td>0.87</td>
</tr>
<tr>
<td>Halton</td>
<td>0.79</td>
<td>0.87</td>
</tr>
<tr>
<td>Distance constraint</td>
<td>0.75</td>
<td>0.86</td>
</tr>
</tbody>
</table>

Theme 3

• Problem:
  • *Can we use ML in a white-box manner to synthesize more resilient controllers?*

• Some evidence that using confidence measure (i.e. output of softmax layer) can help
  • *Reinforcing Adversarial Robustness using Model Confidence Induced by Adversarial Training*, Xi Wu, Uyeong Jang, Jiefeng Chen, Lingjiao Chen, Somesh Jha, ICML 2018
Semantic Adversarial Analysis and Training

DNN analysis must be more *semantic*

- Semantic modification
- System-level specification
- Semantic (re-)training
- Confidence-based analysis

Example: AEBS
Spec: “*do not crash*”

No car
*Keep going*

VS

Maybe car…
*Better slow down*

Prediction: car 49 %
Theme 3

• Problem:
  • *Can we generate adversarial examples that matter (i.e. cause system-level failure)?*
Compositional Falsification

Statement

given a formal specification \( \varphi \) (say in a formalism such as signal temporal logic) and a CPS+ML model \( M \),

find an input for which \( M \) does not satisfy \( \varphi \).

Problem:

*How do handle the ML component?*
Obvious Strategies

• Treat ML component as any other component and
  • Let “abstraction refinement” handle it
  • Will it work?
    • DNN models are constantly getting bigger (>= 20 million parameters)
    • Some folks are talking about a billion parameters

• Use adversarial example generator as a “black box”
  • Will it work?
    • Will generate lot of examples that won’t falsify the system
    • Density of “spurious” adversarial examples is too large
      • This is a conjecture!!!
Our Approach: Use a **System-Level Specification**

- “Verify the Deep Neural Network Object Detector”
- “Verify the System containing the Deep Neural Network”

Formally Specify the **End-to-End Behavior** of the System

Temporal Logic: $G \left( \text{dist}(\text{ego vehicle}, \text{env object}) > \Delta \right)$
Compositional Falsification

• Challenge: Very High Dimensionality of Input Space!
• Standard solution: Use Compositional (Modular) Verification

• However: *no formal spec.* for neural network component!
• Compositional Verification *without* Compositional Specification?!!
Compositional Approach: Combine Temporal Logic CPS Falsifier with ML Analyzer

- CPS Falsifier uses abstraction of ML component
  - Optimistic analysis: assume ML classifier is always correct
  - Pessimistic analysis: assume classifier is always wrong
- Difference is the region of uncertainty where output of the ML component “matters”
Identifying Region of Uncertainty (ROU) for Automatic Emergency Braking System

Green $\rightarrow$ environments where the property is satisfied

ML always correct

ML always wrong

Potentially unsafe region depending on ML component (yellow)
Sample Result

Inception-v3 Neural Network (pre-trained on ImageNet using TensorFlow)

This misclassification may not be of concern
Sample Result

Inception-v3 Neural Network (pre-trained on ImageNet using TensorFlow)

Misclassifications

Corner case Image

But this one is a real hazard!
Theme 4 (*)

• Problem:
  • Can we use the specification to modify the loss function?

• Intuition
  • Steer the ML model towards correcting mis-classifications that cause system-level failure?
  • Initial results, but inconclusive!
    • Trained with hinge loss
    • Does reduce the impact of the collision
Future
Exciting Area

• Several problems mentioned during the talk

• Get involved
  • Several workshops coming up
  • Don’t ignore the email invitations 😊

• Release benchmarks!
  • https://www.robust-ml.org/
  • https://github.com/tensorflow/cleverhans
Get involved!

https://github.com/tensorflow/cleverhans