Recent Trends in Adversarial Machine Learning

Thanks to Ian Goodfellow, Somesh Jha, Patrick McDaniel, and Nicolas Papernot for some slides

December 4, 2018
Berkay Celik
Learning: find classifier function that minimize a cost/loss (~model error)
How it works … run-time …

Inference time: which "class" is most like the input sample
An Example …
I.I.D. Machine Learning

I: Independent
I: Identically
D: Distributed

All train and test examples drawn independently from same distribution
ML reached “human-level performance” on many IID tasks circa 2013

...recognizing objects and faces....

(Szegedy et al, 2014)

...solving CAPTCHAS and reading addresses...

(Taigmen et al, 2013)

(Goodfellow et al, 2013)

(Goodfellow et al, 2013)
Caveats to “human-level” benchmarks

Humans are not very good at some parts of the benchmark

The test data is not very diverse. ML models are fooled by natural but unusual data.
Security Requires Moving Beyond I.I.D.

- Not identical: attackers can use unusual inputs

(Eykholt et al, 2017)

- Not independent: attacker can repeatedly send a single mistake ("test set attack")
Good models make surprising mistakes in non-IID setting

“Adversarial examples”

Schoolbus + Perturbation (rescaled for visualization) = Ostrich

(Szegedy et al, 2013)
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
Attacks on the machine learning pipeline

\[ X \rightarrow \theta \rightarrow \hat{y} \]

- Training data
- Training set poisoning
- Adversarial Examples
- Test input
- Test output
- Model theft
- Recovery of sensitive training data
- Learned parameters
- Learning algorithm
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)
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
Fifty Shades of Gray Box Attacks

- Does the attacker go first, and the defender reacts?
  - This is easy, just train on the attacks, or design some preprocessing to remove them
- If the defender goes first
  - Does the attacker have full knowledge? This is “**white box**”
  - Limited knowledge: “**black box**”
    - Does the attacker know the task the model is solving (input space, output space, defender cost)?
    - Does the attacker know the machine learning algorithm being used?
    - Details of the algorithm? (Neural net architecture, etc.)
    - Learned parameters of the model?
    - Can the attacker send “probes” to see how the defender processes different test inputs?
      - Does the attacker observe just the output class? Or also the probabilities?
Roadmap

- WHITE-BOX ATTACKS
- BLACK-BOX ATTACKS
- TRANSFERABILITY
- DEFENSE TECHNIQUES
White Box Attacks

- Adversary’s problem
  - Given: $x \in X$
  - Find $\delta$
    - $\min_\delta \mu(\delta)$
    - Such that: $F(x + \delta) \in T$
      - Where: $T \subseteq Y$

- Misclassification: $T = Y \setminus \{F(x)\}$
- Targeted: $T = \{t\}$
FGSM (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 2018
Intuition
JSMA (targeted)
Carlini-Wagner (CW) (targeted)

- **Formulation**
  - \[ \min_{\delta} \ | \delta |_2 \]
  - Such that \( F(x + \delta) = t \)

- **Define**
  - \( g(x) = \max(\max_{i \neq t} Z(F)(x)[i] - Z(F)(x)[t], -\kappa) \)
  - Replace the constraint
    - \( g(x) \leq 0 \)

- **Paper**
Success of an adversarial image

Experiments excluding MNIST 1s, many of which look like 7s

<table>
<thead>
<tr>
<th>Pair</th>
<th>Diff</th>
<th>L0</th>
<th>L1</th>
<th>L2</th>
<th>L∞</th>
</tr>
</thead>
<tbody>
<tr>
<td>L0</td>
<td>![3, 7, 3]</td>
<td>63</td>
<td>35.0</td>
<td>4.86</td>
<td>1.0</td>
</tr>
<tr>
<td>L1</td>
<td>![7, 9, 7]</td>
<td>91</td>
<td>19.9</td>
<td>3.21</td>
<td>.996</td>
</tr>
<tr>
<td>L2</td>
<td>![4, 9, 9]</td>
<td>110</td>
<td>21.7</td>
<td>2.83</td>
<td>1.0</td>
</tr>
<tr>
<td>L∞</td>
<td>![4, 9, 9]</td>
<td>121</td>
<td>34.0</td>
<td>3.82</td>
<td>.76</td>
</tr>
</tbody>
</table>
Black-box Attacks

(1) The adversary queries remote ML system for labels on inputs of its choice.

“no truck sign”

(2) The adversary uses this labeled data to train a local substitute for the remote system.

“no truck sign”
Black-box Attacks

(3) The adversary selects new synthetic inputs for queries to the remote ML system based on the local substitute's output surface sensitivity to input variations.

(4) The adversary then uses the local substitute to craft adversarial examples, which are misclassified by the remote ML system because of transferability.

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

• WHITE-BOX ATTACKS
• BLACK-BOX ATTACKS
• TRANSFERABILITY
• DEFENSE TECHNIQUES
Pipeline of Defense Failures

No effect on advx

Reduces advx, but reduces clean accuracy too much

Does not affect adaptive attacker

Does not generalize over attack algos

Seems to generalize, but it’s an illusion

Does not generalize over threat models

No effect on advx

Reduces advx, but reduces clean accuracy too much
Pipeline of Defense Failures

Dropout at Train Time

- Does not generalize over threat models
- Seems to generalize, but it’s an illusion
- Does not generalize over attack algos
- Does not affect adaptive attacker
- Reduces advx, but reduces clean accuracy too much
- No effect on advx
Pipeline of Defense Failures

- No effect on advx
- Reduces advx, but reduces clean accuracy too much
- Does not affect adaptive attacker
- Does not generalize over attack algos
- Does not generalize over threat models
- Seems to generalize, but it’s an illusion
- Weight Decay
- Reduces advx, but reduces clean accuracy too much
- No effect on advx
Pipeline of Defense Failures

- Cropping / fovea mechanisms
- No effect on advx
- Reduces advx, but reduces clean accuracy too much
- Does not affect adaptive attacker
- Does not generalize over attack algos
- Seems to generalize, but it’s an illusion
- Does not generalize over threat models
- No effect on advx

Original vs Foveal images
Adversarial Training with a Weak Attack

- No effect on advx
- Reduces advx, but reduces clean accuracy too much
- Does not affect adaptive attacker
- Does not generalize over threat models
- Seems to generalize, but it’s an illusion
- Does not generalize over attack algos
- Reduces advx, but reduces clean accuracy too much
- No effect on advx
Pipeline of Defense Failures

Defensive Distillation

No effect on advx

Reduces advx, but reduces clean accuracy too much

Does not affect adaptive attacker

Does not generalize over attack algos

Does not generalize over threat models

Seems to generalize, but it’s an illusion

No effect on advx

Reduces advx, but reduces clean accuracy too much
Pipeline of Defense Failures

Adversarial Training with a Strong Attack
Current Certified / Provable Defenses

- Does not generalize over threat models
- Seems to generalize, but it’s an illusion
- Does not generalize over attack algos
- Does not affect adaptive attacker
- Reduces advx, but reduces clean accuracy too much
- No effect on advx
What's next defense?

Robust Objectives

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

Future Directions

- Common goal (AML and ML)
  - Just make the model better

- They still share this goal

- It is now clear security research must have some independent goals. For two models with the same error volume, for reasons of security we prefer:
  - The model with lower confidence on mistakes
  - The model whose mistakes are harder to find
THANKS!

December 4, 2018
Berkay Celik