Secure Learning in Adversarial Deep Neural Networks

Bo Li
UC, Berkeley
Machine Learning in Physical World

Autonomous Driving

Healthcare

Smart City

Malware Classification

Fraud Detection

Biometrics Recognition
Security & Privacy Problems

Syrian hackers claim AP hack that tipped stock market by $136 billion. Is it terrorism?

By Max Fisher  April 23, 2013

Breaking: Two Explosions in the White House and Barack Obama is injured

This chart shows the Dow Jones Industrial Average during Tuesday afternoon's drop, caused by a fake

Biometric recognition at airport border raises privacy concerns, says expert

Plan would involve 90% of passengers being processed through Australian immigration without human involvement
We Are in Adversarial Environments
While cybersecurity R&D needs are addressed in greater detail in the NITRD Cybersecurity R&D Strategic Plan, some cybersecurity risks are specific to AI systems. **One key research area is “adversarial machine learning”**, that explores the degree to which AI systems can be compromised by “contaminating” training data, by modifying algorithms, or by making subtle changes to an object that prevent it from being correctly identified....

  - National Science and Technology Council 2016
Perils of Stationary Assumption

Traditional machine learning approaches assume

\[ \text{Training Data} \approx \text{Testing Data} \]
Adversarial Examples

\[ \eta = \epsilon \text{sign}(\nabla_x J(\theta, x, y)) \]


Optimization Based Attack

\[
\begin{align*}
\text{minimize} & \quad D(x, x + \delta) \\
\text{such that} & \quad C(x + \delta) = t \\
x + \delta & \in [0, 1]^n
\end{align*}
\]

\[
\begin{align*}
\text{minimize} & \quad D(x, x + \delta) + c \cdot f(x + \delta) \\
\text{such that} & \quad x + \delta \in [0, 1]^n
\end{align*}
\]

<table>
<thead>
<tr>
<th></th>
<th>MNIST</th>
<th>CIFAR</th>
<th>MNIST</th>
<th>CIFAR</th>
<th>MNIST</th>
<th>CIFAR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>prob</td>
<td>mean</td>
<td>prob</td>
<td>mean</td>
<td>prob</td>
</tr>
<tr>
<td>Our $L_0$</td>
<td>8.5</td>
<td>100%</td>
<td>5.9</td>
<td>100%</td>
<td>16</td>
<td>100%</td>
</tr>
<tr>
<td>JSMA-Z</td>
<td>20</td>
<td>100%</td>
<td>20</td>
<td>100%</td>
<td>56</td>
<td>100%</td>
</tr>
<tr>
<td>JSMA-F</td>
<td>17</td>
<td>100%</td>
<td>25</td>
<td>100%</td>
<td>45</td>
<td>100%</td>
</tr>
<tr>
<td>Our $L_2$</td>
<td>1.36</td>
<td>100%</td>
<td>0.17</td>
<td>100%</td>
<td>1.76</td>
<td>100%</td>
</tr>
<tr>
<td>Deepfool</td>
<td>2.11</td>
<td>100%</td>
<td>0.85</td>
<td>100%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Our $L_\infty$</td>
<td>0.13</td>
<td>100%</td>
<td>0.0092</td>
<td>100%</td>
<td>0.16</td>
<td>100%</td>
</tr>
<tr>
<td>Fast Gradient Sign</td>
<td>0.22</td>
<td>100%</td>
<td>0.015</td>
<td>99%</td>
<td>0.26</td>
<td>42%</td>
</tr>
<tr>
<td>Iterative Gradient Sign</td>
<td>0.14</td>
<td>100%</td>
<td>0.0078</td>
<td>100%</td>
<td>0.19</td>
<td>100%</td>
</tr>
</tbody>
</table>

Autonomous Driving is the Trend...
However, What We Can See Everyday...
The Physical World Is... Messy

Varying Physical Conditions (Angle, Distance, Lighting, ...)  Physical Limits on Imperceptibility

Fabrication/Perception Error (Color Reproduction, etc.)

Background Modifications*

[Evitmov, Eykholt, Fernandes, Kohno, Li, Prakash, Rahmati, and Song, 2017]
An Optimization Approach To Creating Robust Physical Adversarial Examples

\[
\begin{align*}
\text{argmin}_{\delta} & \quad \lambda \|\delta\|_p + J(f_\theta(x + \delta), y^*) \\
p \text{ norm (L-0, L-1, L-2, ...)} & \quad \text{Loss Function} \\
\text{Adversarial Target Label} & \\
\text{Perturbation/Noise Matrix} &
\end{align*}
\]
Optimizing Spatial Constraints (Handling Limits on Imperceptibility)

$$\arg\min_{\delta} \lambda \|M_x \cdot \delta\|_p + \frac{1}{k} \sum_{i=1}^{k} J(f_\theta(x_i + M_x \cdot \delta), y^*)$$

Subtle Poster
Camouflage Sticker

Mimic vandalism
“Hide in the human psyche”
Handling Fabrication/Perception Errors

\[ \arg\min_{\delta} \lambda \|M_x \cdot \delta\|_p + \frac{1}{k} \sum_{i=1}^{k} J(f_{\theta}(x_i + M_x \cdot \delta), y^*) + NPS(M_x \cdot \delta) \]

\[ NPS(\delta) = \sum_{\hat{p} \in \delta} \prod_{p' \in P} |\hat{p} - p'| \]

NPS based on Sharif et al., “Accessorize to a crime,” CCS 2016
How Can We Realistically Evaluate Attacks?

**Lab Test (Stationary)**

- Angles = 0°, 15°, 30°, ...
- Road Sign (Top View)
- 5'
- 10'
- 40'
- Camera

**Field Test (Drive-By)**

- ~ 250 feet, 0 to 20 mph
- Record video
- Sample frames every k frames
- Run sampled frames through DNN
Lab Test Summary (Stationary)

Target Class: Speed Limit 45
Art Perturbation
Subtle Perturbation
Physical Attacks Against Detectors
Physical Attacks Against Detectors
Adversarial Examples in Physical World

Adversarial perturbations are possible in physical world under different conditions and viewpoints, including the distances and angles.
Different approaches to optimize the objective

• Fast approaches
  – Fast gradient sign ($d = ||\cdot||_{\infty}$): $x^* = x + B \text{sgn}(\nabla_x \ell(f_\theta(x), y))$

  – Fast gradient ($d = ||\cdot||_2$): $x^* = x + B \left( \frac{\nabla_x \ell(f_\theta(x), y)}{||\nabla_x \ell(f_\theta(x), y)||_2} \right)$

• Iterative approaches
  – E.g., use a SGD optimizer, such as Adam, to optimize

\[
\max_{x^*} \ell(f_\theta(x^*), y) + \lambda d(x, x^*)
\]

• Optimization
\[
\argmin_{\delta} \lambda ||\delta||_p + J(f_\theta(x + \delta), y^*)
\]

• Need to know model $f_\theta$
A General Framework for Black-box attacks

- **Zero-Query Attack**
  - Random perturbation
  - Difference of means
  - *Transferability-based attack*
    - Practical Black-Box Attacks against Machine Learning
    - Ensemble transferability-based attack

- **Query Based Attack**
  - Finite difference gradient estimation
  - Query reduced gradient estimation
  - Results: similar effectiveness to whitebox attack
  - A general active query game model
## Transferability

<table>
<thead>
<tr>
<th>Machine Learning Technique</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN</td>
<td>97.72</td>
<td>97.91</td>
<td>97.91</td>
<td>97.6</td>
<td>97.62</td>
</tr>
<tr>
<td>LR</td>
<td>82.57</td>
<td>83.45</td>
<td>84.07</td>
<td>83.16</td>
<td>82.98</td>
</tr>
<tr>
<td>SVM</td>
<td>88.9</td>
<td>89.07</td>
<td>89.29</td>
<td>88.84</td>
<td>88.9</td>
</tr>
<tr>
<td>DT</td>
<td>80.64</td>
<td>81.57</td>
<td>80.94</td>
<td>81.78</td>
<td>81.55</td>
</tr>
<tr>
<td>kNN</td>
<td>94.42</td>
<td>94.92</td>
<td>94.83</td>
<td>94.91</td>
<td>94.44</td>
</tr>
</tbody>
</table>

**MNIST**

**PDF Malware**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>0.84</td>
<td>0.84</td>
<td>0.84</td>
<td>0.84</td>
<td>0.84</td>
</tr>
<tr>
<td>NB</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
</tr>
<tr>
<td>MLP</td>
<td>0.85</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
<td>0.87</td>
</tr>
<tr>
<td>RF</td>
<td>0.74</td>
<td>0.73</td>
<td>0.73</td>
<td>0.73</td>
<td>0.73</td>
</tr>
<tr>
<td>DT</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.86</td>
<td>0.88</td>
</tr>
<tr>
<td>kNN</td>
<td>0.87</td>
<td>0.87</td>
<td>0.86</td>
<td>0.86</td>
<td>0.86</td>
</tr>
<tr>
<td>LR</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
</tr>
</tbody>
</table>

Papernot, McDaniel, Goodfellow, Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. 2016
Xiao, Li, Malware Evasion Attacks Based on Generative Adversarial Networks (GANs), 2017.
Targeted vs Non-targeted

• Non-targeted adversarial examples
  • The goal is to mislead the classifier to predict any labels other than the ground truth
  • Most existing work deals with this goal

• Targeted adversarial examples
  • The goal is to mislead the classifier to predict a target label for an image
  • Harder!
Ground truth: running shoe

<table>
<thead>
<tr>
<th>Model</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG16</td>
<td>Military uniform</td>
</tr>
<tr>
<td>ResNet50</td>
<td>Jigsaw puzzle</td>
</tr>
<tr>
<td>ResNet101</td>
<td>Motor scooter</td>
</tr>
<tr>
<td>ResNet152</td>
<td>Mask</td>
</tr>
<tr>
<td>GoogLeNet</td>
<td>Chainsaw</td>
</tr>
</tbody>
</table>
Targeted Adversarial Example’s Transferability Among Two Models is Poor!

<table>
<thead>
<tr>
<th></th>
<th>ResNet152</th>
<th>ResNet101</th>
<th>ResNet50</th>
<th>VGG16</th>
<th>GoogLeNet</th>
<th>Incept-v3</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet152</td>
<td>100%</td>
<td>2%</td>
<td>1%</td>
<td>1%</td>
<td>1%</td>
<td>0%</td>
</tr>
<tr>
<td>ResNet101</td>
<td>3%</td>
<td>100%</td>
<td>3%</td>
<td>2%</td>
<td>1%</td>
<td>1%</td>
</tr>
<tr>
<td>ResNet50</td>
<td>4%</td>
<td>2%</td>
<td>100%</td>
<td>1%</td>
<td>1%</td>
<td>0%</td>
</tr>
<tr>
<td>VGG16</td>
<td>2%</td>
<td>1%</td>
<td>2%</td>
<td>100%</td>
<td>1%</td>
<td>0%</td>
</tr>
<tr>
<td>GoogLeNet</td>
<td>1%</td>
<td>1%</td>
<td>0%</td>
<td>1%</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>Incept-v3</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Only 2% of the adversarial images generated for VGG16 (row) can be predicted as the targeted label by ResNet50 (column)
Black-box Attacks Based On Transferability

Adversary

White-Box Model

Adversarial Examples

Transfer to

Black-Box System
Ensemble Targeted Black-box Attacks Based On Transferability

Liu, Chen, Liu, Song. Delving into Transferable Adversarial Examples and Black-box Attacks, ICLR 2017
Ground truth from ImageNet: broom

jacamar
Adversarial Example on Clarifai.com

- Ground truth: *broom*
- Target label: *jacamar*
Ground truth on ImageNet: Waterbuffalo
Adversarial Example on Clarifai.com

- Ground truth: water buffalo
- Target label: rugby ball
Ground truth from ImageNet: rosehip

stupa

LCL-S17. Delving into Transferable Adversarial Examples and Black-box Attacks, ICLR 2017
Adversarial Example on Clarifai.com

- Ground truth: rosehip
- Target label: stupā

LCLS17. Delving into Transferable Adversarial Examples and Black-box Attacks, ICLR 2017
Black-box attacks

• Zero-Query Attack (Previous methods)
  – Random perturbation
  – Difference of means
  – Transferability based attack

• Query Based Attack (Our methods)
  – Finite difference gradient estimation
  – Query reduced gradient estimation

The zero-query attack can be viewed as a special case for the query based attack, where the number of queries made is zero

[Bhagoji, Li, He, Dawn, 2017]
Query Based attacks

- Finite difference gradient estimation
  - Given $d$-dimensional vector $x$, we can make $2d$ queries to estimate the gradient as below
    \[
    \text{FD}_x(g(x), \delta) = \begin{bmatrix}
    \frac{g(x+\delta e_1) - g(x-\delta e_1)}{2\delta} \\
    \vdots \\
    \frac{g(x+\delta e_d) - g(x-\delta e_d)}{2\delta}
    \end{bmatrix}
    \]
  - An example of approximate FGS with finite difference
    \[
    x_{adv} = x + \epsilon \cdot \text{sign} \left( \text{FD}_x(\ell_f(x, y), \delta) \right)
    \]
- Query reduced gradient estimation
  - Random grouping
  - PCA

Similarly, we can also approximate for logit-based loss by making $2d$ queries
Effectiveness of various single step black-box attacks on MNIST. The $y$-axis represents the variation in adversarial success as $\epsilon$ increases.

Finite Differences method outperform other black-box attacks and achieves similar attach success rate with the white-box attack.
Effectiveness of various single step black-box attacks on CIFAR-10. The $\epsilon$-axis represents the variation in adversarial success as $\epsilon$ increases.

Finite Differences method outperform other black-box attacks and achieves similar attack success rate with the white-box attack.
Adversarial success rates for Gradient Estimation attacks with query reduction on Model A (MNIST) and Resnet-32 (CIFAR-10).

Finite Differences method with query reduction perform approximately similar with the gradient estimation black-box attack.
Black-box Attack Clarifai

Original image, classified as “drug” with a confidence of 0.99

Adversarial example, classified as “safe” with a confidence of 0.96

The Gradient Estimation black-box attack on Clarifai’s Content Moderation Model
Black-box Attacks

Black-box attacks are possible on deep neural networks with **query access**.
The number of queries needed can be reduced.
Generating Adversarial Examples with Adversarial Networks

\[ \mathcal{L}_{GAN} = \mathbb{E}_{x \sim \mathcal{P}_{data}} \log \mathcal{D}(x) + \mathbb{E}_{x \sim \mathcal{P}_{data}} \log(1 - \mathcal{D}(x + \mathcal{G}(x))) \]

\[ \mathcal{L} = \mathcal{L}^{f}_{adv} + \alpha \mathcal{L}_{GAN} + \beta \mathcal{L}_{hinge} \]

The GAN loss here tries to ensure the diversity of adversarial examples

[Chaowei Xiao, Bo Li, Jun-yan Zhu, Warren He, Mingyan Liu, Dawn Song, 2017]
The perturbed images are very close to the original ones. The original images lie on the diagonal.
The perturbed images are very close to the original ones. The original images lie on the diagonal.
Poodle  Ambulance  Basketball  Electric guitar
## Attack Effectiveness Under Defenses

<table>
<thead>
<tr>
<th>Data</th>
<th>Model</th>
<th>Defense</th>
<th>FGSM</th>
<th>Opt.</th>
<th>AdvGAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST</td>
<td>A</td>
<td>Adv.</td>
<td>4.3%</td>
<td>4.6%</td>
<td>8.0%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ensemble</td>
<td>1.6%</td>
<td>4.2%</td>
<td>6.3%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Iter. Adv.</td>
<td>4.4%</td>
<td>2.96%</td>
<td>5.6%</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>Adv.</td>
<td>6.0%</td>
<td>4.5%</td>
<td>7.2%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ensemble</td>
<td>2.7%</td>
<td>3.18%</td>
<td>5.8%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Iter. Adv.</td>
<td>9.0%</td>
<td>3.0%</td>
<td>6.6%</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>Adv.</td>
<td>2.7%</td>
<td>2.95%</td>
<td>18.7%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ensemble</td>
<td>1.6%</td>
<td>2.2%</td>
<td>13.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Iter. Adv.</td>
<td>1.6%</td>
<td>1.9%</td>
<td>12.6%</td>
</tr>
<tr>
<td>CIFAR</td>
<td>ResNet</td>
<td>Adv.</td>
<td>13.10%</td>
<td>11.9%</td>
<td>16.03%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ensemble.</td>
<td>10.00%</td>
<td>10.3%</td>
<td>14.32%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Iter. Adv</td>
<td>22.8%</td>
<td>21.4%</td>
<td>29.47%</td>
</tr>
<tr>
<td></td>
<td>Wide ResNet</td>
<td>Adv.</td>
<td>5.04%</td>
<td>7.61%</td>
<td>14.26%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ensemble</td>
<td>4.65%</td>
<td>8.43%</td>
<td>13.94%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Iter. Adv.</td>
<td>14.9%</td>
<td>13.90%</td>
<td>20.75%</td>
</tr>
</tbody>
</table>

Attack success rate of adversarial examples generated by AdvGAN in semi-whitebox setting under defenses on MNIST and CIFAR-10.
## Attack Effectiveness Under Defenses

### Black-Box Leaderboard (Original Challenge)

<table>
<thead>
<tr>
<th>Attack</th>
<th>Submitted by</th>
<th>Accuracy</th>
<th>Submission Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdvGAN from &quot;Generating Adversarial Examples with Adversarial Networks&quot;</td>
<td>AdvGAN</td>
<td>92.76%</td>
<td>Sep 25, 2017</td>
</tr>
<tr>
<td>PGD against three independently and adversarially trained copies of the network</td>
<td>Florian Tramèr</td>
<td>93.54%</td>
<td>Jul 5, 2017</td>
</tr>
<tr>
<td>FGSM on the CW loss for model B from &quot;Ensemble Adversarial Training [...]&quot;</td>
<td>Florian Tramèr</td>
<td>94.36%</td>
<td>Jun 29, 2017</td>
</tr>
<tr>
<td>FGSM on the CW loss for the naturally trained public network</td>
<td>(initial entry)</td>
<td>96.08%</td>
<td>Jun 28, 2017</td>
</tr>
<tr>
<td>PGD on the cross-entropy loss for the naturally trained public network</td>
<td>(initial entry)</td>
<td>96.81%</td>
<td>Jun 28, 2017</td>
</tr>
<tr>
<td>Attack using Gaussian Filter for selected pixels on the adversarially trained public network</td>
<td>Anonymous</td>
<td>97.33%</td>
<td>Aug 27, 2017</td>
</tr>
<tr>
<td>FGSM on the cross-entropy loss for the adversarially trained public network</td>
<td>(initial entry)</td>
<td>97.66%</td>
<td>Jun 28, 2017</td>
</tr>
<tr>
<td>PGD on the cross-entropy loss for the adversarially trained public network</td>
<td>(initial entry)</td>
<td>97.79%</td>
<td>Jun 28, 2017</td>
</tr>
</tbody>
</table>
Spatially Transformed Adversarial Examples

\[ f^* = \arg\min_f \mathcal{L}_{adv}(x, f) + \tau \mathcal{L}_{flow}(f), \]

[Xiao, Zhu, Li, He, Liu, Song. ICLR 2018]
Examples generated by stAdv

Flow visualization on MNIST. The digit “0” is misclassified as “2”.

Adversarial examples generated by stAdv on MNIST
The ground truth images are shown in the diagonal
### Attack Effectiveness Under Defenses

<table>
<thead>
<tr>
<th>Model</th>
<th>Def.</th>
<th>FGSM</th>
<th>C&amp;W.</th>
<th>stAdv</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Adv.</td>
<td>4.3%</td>
<td>4.6%</td>
<td><strong>32.62%</strong></td>
</tr>
<tr>
<td></td>
<td>Ens.</td>
<td>1.6%</td>
<td>4.2%</td>
<td><strong>48.07%</strong></td>
</tr>
<tr>
<td></td>
<td>PGD</td>
<td>4.4%</td>
<td>2.96%</td>
<td><strong>48.38%</strong></td>
</tr>
<tr>
<td>B</td>
<td>Adv.</td>
<td>6.0%</td>
<td>4.5%</td>
<td><strong>50.17%</strong></td>
</tr>
<tr>
<td></td>
<td>Ens.</td>
<td>2.7%</td>
<td>3.18%</td>
<td><strong>46.14%</strong></td>
</tr>
<tr>
<td></td>
<td>PGD</td>
<td>9.0%</td>
<td>3.0%</td>
<td><strong>49.82%</strong></td>
</tr>
<tr>
<td>C</td>
<td>Adv.</td>
<td>3.22%</td>
<td>0.86%</td>
<td><strong>30.44%</strong></td>
</tr>
<tr>
<td></td>
<td>Ens.</td>
<td>1.45%</td>
<td>0.98%</td>
<td><strong>28.82%</strong></td>
</tr>
<tr>
<td></td>
<td>PGD</td>
<td>2.1%</td>
<td>0.98%</td>
<td><strong>28.13%</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Def.</th>
<th>FGSM</th>
<th>C&amp;W.</th>
<th>stAdv</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet32</td>
<td>Adv.</td>
<td>13.10%</td>
<td>11.9%</td>
<td><strong>43.36%</strong></td>
</tr>
<tr>
<td></td>
<td>Ens.</td>
<td>10.00%</td>
<td>10.3%</td>
<td><strong>36.89%</strong></td>
</tr>
<tr>
<td></td>
<td>PGD</td>
<td>22.8%</td>
<td>21.4%</td>
<td><strong>49.19%</strong></td>
</tr>
<tr>
<td>wide ResNet34</td>
<td>Adv.</td>
<td>5.04%</td>
<td>7.61%</td>
<td><strong>31.66%</strong></td>
</tr>
<tr>
<td></td>
<td>Ens.</td>
<td>4.65%</td>
<td>8.43%</td>
<td><strong>29.56%</strong></td>
</tr>
<tr>
<td></td>
<td>PGD</td>
<td>14.9%</td>
<td>13.90%</td>
<td><strong>31.6%</strong></td>
</tr>
</tbody>
</table>

Attack success rate of adversarial examples generated by stAdv against different models under standard defense on MNIST and CIFAR-10.
Attention of network

CAM attention visualization for ImageNet inception_v3 model. (a) the original image and (b)-(d) are stAdv adversarial examples targeting different classes. Row 2 shows the attention visualization for the corresponding images above.
CAM attention visualization for ImageNet inception_v3 model. Column 1 shows the CAM map corresponding to the original image. Column 2-4 show the adversarial examples generated by different methods. (a) and (e)-(g) are labeled as the ground truth “cinema”, while (b)-(d) and (h) are labeled as the adversarial target “missile.”
Adversarial Examples Prevalent in Deep Learning Systems

• Most existing work on adversarial examples:
  – Image classification task
  – Target model is known

• Our investigation on adversarial examples:

  - Generative Models
  - Deep Reinforcement Learning
  - VisualQA/Image-to-code
  - Other tasks and model classes

  - New Attack Methods
  - Provide more diversity of attacks

  - Blackbox Attacks
    Weaker Threat Models (Target model is unknown)
Generative models

- VAE-like models (VAE, VAE-GAN) use an intermediate latent representation.
- An **encoder**: maps a high-dimensional input into a lower-dimensional latent representation $z$.
- A **decoder**: maps the latent representation back to a high-dimensional reconstruction.
Adversarial Examples in Generative Models

- An example attack scenario:
  - Generative model used as a compression scheme

- Attacker’s goal: for the decompressor to reconstruct a different image from the one that the compressor sees.
Adversarial Examples for VAE-GAN in MNIST

Jernej Kos, Ian Fischer, Dawn Song: Adversarial Examples for Generative Models
Adversarial Examples for VAE-GAN in SVHN

Original images
Reconstruction of original images
Adversarial examples
Reconstruction of adversarial examples

Target Image

Jernej Kos, Ian Fischer, Dawn Song: Adversarial Examples for Generative Models
Adversarial Examples for VAE-GAN in SVHN

Original images

Reconstruction of original images

Adversarial examples

Reconstruction of adversarial examples

Target Image

Jernej Kos, Ian Fischer, Dawn Song: Adversarial Examples for Generative Models
Takeaways

VAE-like generative models are vulnerable to adversarial examples
Visual Question & Answer (VQA)

Q: Where is the plane?

Fooling VQA

Target: Sky

**Benign image**

Answer: Runway

**VQA Model**

**Adversarial example**

**VQA Model**

Answer: Sky
Q: How many cats are there?

Fooling VQA

Target: 2

Answer:

1

2
A3C: A Deep Policy on Pong

Reinforcement learning algorithms:

• Actor – policy network to predict the action based on each frame

• Critics – value function to predict the value of each frame, and the action is chosen to maximize the expected value

• Actor-critics (A3C) – combine value function into the policy network to make prediction
Agent in Action: attack the policy network

Original Frames

Adversarial perturbation injected into every frame

Jernej Kos and Dawn Song: Delving into adversarial attacks on deep policies. ICLR Workshop 2017
Agent in Action: attack the value function

Original Frames

Adversarial perturbation injected into every other 10 frames

Song et al.: Delving into adversarial attacks on deep policies. ICLR Workshop 2017
Takeaways

**Reinforcement learning** systems (e.g., robotics, self-driving systems) are also **vulnerable** to adversarial examples.

To attack a reinforcement learning system, **adversarial perturbations need not be injected to every frame**.
Coffee Break!

Amazon Machine Learning
aws.amazon.com/machine-learning

TensorFlow

torch
Facebook AI Research

Spark

mxnet

H2O.ai

Coffee Break!
Numerous Defenses Proposed

- Ensemble
- Normalization
- Distributional detection
- PCA detection
- Secondary classification
- Stochastic
- Generative
- Training process
- Architecture
- Retrain
- Pre-process input

Detection

Prevention
Towards Deep Learning Models Resistant to Adversarial Attacks

\[ \min_{\theta} \rho(\theta), \quad \text{where} \quad \rho(\theta) = \mathbb{E}_{(x,y) \sim D} \left[ \max_{\delta \in S} L(\theta, x + \delta, y) \right] \]

- Use a natural saddle point (min-max) formulation to capture the notion of security against adversarial attacks in a principled manner.
- The formulation casts both attacks and defenses into a common theoretical framework.
- Motivate projected gradient descent (PGD) as a universal “first-order adversary”.

Model Capacity
Towards Deep Learning Models Resistant to Adversarial Attacks

MNIST

Accuracy vs Capacity scale

Average loss vs Capacity scale

Natural
FGSM
PGD
Decision Boundary Analysis of Adversarial Examples

He, Li, Song, Decision Boundary Analysis of Adversarial Examples, ICLR 2017.
<table>
<thead>
<tr>
<th>Training attack</th>
<th>False pos.</th>
<th>False neg.</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Benign</td>
<td>OPTBrittle</td>
<td>OPTMargin</td>
</tr>
<tr>
<td>----------------</td>
<td>------------</td>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td><strong>MNIST, normal training</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPTBrittle</td>
<td>1.0%</td>
<td>1.0%</td>
<td>74.1%</td>
</tr>
<tr>
<td>OPTMargin</td>
<td>9.6%</td>
<td>0.6%</td>
<td>7.2%</td>
</tr>
<tr>
<td><strong>MNIST, PGD adversarial training</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPTBrittle</td>
<td>2.6%</td>
<td>2.0%</td>
<td>39.8%</td>
</tr>
<tr>
<td>OPTMargin</td>
<td>10.3%</td>
<td>0.4%</td>
<td>14.5%</td>
</tr>
<tr>
<td><strong>CIFAR-10, normal training</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPTBrittle</td>
<td>5.3%</td>
<td>3.2%</td>
<td>56.8%</td>
</tr>
<tr>
<td>OPTMargin</td>
<td>8.4%</td>
<td>7.4%</td>
<td>5.3%</td>
</tr>
<tr>
<td><strong>CIFAR-10, PGD adversarial training</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPTBrittle</td>
<td>0.0%</td>
<td>2.4%</td>
<td>51.8%</td>
</tr>
<tr>
<td>OPTMargin</td>
<td><strong>3.6%</strong></td>
<td>0.0%</td>
<td>1.2%</td>
</tr>
</tbody>
</table>
Adversarial Examples Detection

![Diagram with scatter plots and labels indicating normal, adversarial (adv), and random (rand) examples.](image)

- **Top-left graph:** Shows scatter plots with markers for normal, adversarial, and random examples. KM=0.19, KD=0.92, LID=1.53.
- **Top-right graph:** Similar to the top-left, KM=0.19, KD=0.92, LID=4.36.

**Bottom graphs:**
- **Left:** 100 random CIFAR samples with k-mean distance plotted.
- **Right:** 100 random CIFAR samples with k-mean distance (PCA) plotted.
Adversarial Examples Detection via Local Intrinsic Dimensionality (LID)
# Adversarial Examples Detection

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Feature</th>
<th>FGM</th>
<th>BIM-a</th>
<th>BIM-b</th>
<th>JSMA</th>
<th>Opt</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MNIST</strong></td>
<td>KD</td>
<td>78.12%</td>
<td>99.14%</td>
<td>98.61%</td>
<td>68.77%</td>
<td>95.15%</td>
</tr>
<tr>
<td></td>
<td>BU</td>
<td>32.37%</td>
<td>91.55%</td>
<td>25.46%</td>
<td>88.74%</td>
<td>71.29%</td>
</tr>
<tr>
<td></td>
<td>KD+BU</td>
<td>82.43%</td>
<td>99.20%</td>
<td>98.81%</td>
<td>90.12%</td>
<td>95.35%</td>
</tr>
<tr>
<td></td>
<td>LID</td>
<td><strong>96.89%</strong></td>
<td><strong>99.60%</strong></td>
<td><strong>99.83%</strong></td>
<td><strong>92.24%</strong></td>
<td><strong>99.24%</strong></td>
</tr>
<tr>
<td><strong>CIFAR-10</strong></td>
<td>KD</td>
<td>64.92%</td>
<td>68.38%</td>
<td>98.70%</td>
<td>85.77%</td>
<td>91.35%</td>
</tr>
<tr>
<td></td>
<td>BU</td>
<td>70.53%</td>
<td>81.60%</td>
<td>97.32%</td>
<td>87.36%</td>
<td>91.39%</td>
</tr>
<tr>
<td></td>
<td>KD+BU</td>
<td>70.40%</td>
<td>81.33%</td>
<td>98.90%</td>
<td>88.91%</td>
<td>93.77%</td>
</tr>
<tr>
<td></td>
<td>LID</td>
<td><strong>82.38%</strong></td>
<td><strong>82.51%</strong></td>
<td><strong>99.78%</strong></td>
<td><strong>95.87%</strong></td>
<td><strong>98.93%</strong></td>
</tr>
<tr>
<td><strong>SVHN</strong></td>
<td>KD</td>
<td>70.39%</td>
<td>77.18%</td>
<td>99.57%</td>
<td>86.46%</td>
<td>87.41%</td>
</tr>
<tr>
<td></td>
<td>BU</td>
<td>86.78%</td>
<td>84.07%</td>
<td>86.93%</td>
<td>91.33%</td>
<td>87.13%</td>
</tr>
<tr>
<td></td>
<td>KD+BU</td>
<td>86.86%</td>
<td>83.63%</td>
<td>99.52%</td>
<td>93.19%</td>
<td>90.66%</td>
</tr>
<tr>
<td></td>
<td>LID</td>
<td><strong>97.61%</strong></td>
<td><strong>87.55%</strong></td>
<td><strong>99.72%</strong></td>
<td><strong>95.07%</strong></td>
<td><strong>97.60%</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attack Failure Rate</th>
<th>MNIST</th>
<th>CIFAR-10</th>
<th>SVHN</th>
</tr>
</thead>
<tbody>
<tr>
<td>(one-layer)</td>
<td>100%</td>
<td>95.7%</td>
<td>97.2%</td>
</tr>
<tr>
<td>(all-layer)</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>
There Is Still A Long Way For Defense

- Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods [Carlini, Wagner]
- Better threat model
- Better understanding of neural networks
Poisoning Attacks

Traditional machine learning approaches assume

Training Data

≈

Testing Data
Machine Learning

Training

Training Data

Machine Learning Training

Inference

Test Data

Machine Learning model

Prediction
Training Data Poisoning

Training is under control of the defender

Inference

Test data is not tampered
Data Poisoning Attacks for Factorization Based Collaborative Filtering

• Problem
  – Poisoning attack within learning systems
  – Recommendation systems
• Nearest neighbor methods
• Low-rank Matrix Completion

![Ratings Matrix]

- Task: Complete ratings matrix
- Applications: recommendation systems, PCA with missing entries

Data Poisoning Attacks for Factorization Based Collaborative Filtering

• Preliminaries
  – Low rank matrix completion
    \[
    \min_{X \in \mathbb{R}^{m \times n}} \| \mathcal{R}_\Omega (M - X) \|_F^2, \quad \text{s.t. } \text{rank}(X) \leq k
    \]
    where \( \| A \|_F^2 = \sum_{i,j} A_{ij}^2 \)
  – Alternating minimization
    \[
    \min_{U \in \mathbb{R}^{m \times k}, V \in \mathbb{R}^{n \times k}} \left\{ \| \mathcal{R}_\Omega (M - UV^T) \|_F^2 \right. \\
    \left. + 2\lambda_U \| U \|_F^2 + 2\lambda_V \| V \|_F^2 \right\}
    \]
  – Nuclear norm minimization
    \[
    \min_{X \in \mathbb{R}^{m \times n}} \| \mathcal{R}_\Omega (M - X) \|_F^2 + 2\lambda \| X \|_*
    \]
Data Poisoning Attacks for Factorization Based Collaborative Filtering

• Threat model
  – Assume attack malicious inject $\alpha m$ rows $\tilde{M}$
    $$\Theta_{\lambda}(\tilde{M}; M) = \arg\min_{U, \tilde{U}, V} \left\{ \| R_{\Omega}(M - UV^T) \|_F^2 + \| R_{\tilde{\Omega}}(\tilde{M}) \|_F^2 \right\}$$
    $$= \tilde{U}V^T + 2\lambda_U (\| U \|_F^2 + \| \tilde{U} \|_F^2) + 2\lambda_V \| V \|_F^2$$
  – Availability attack
    $$R^{av}(\tilde{M}, M) = \| R_{\Omega_c}(\tilde{M} - M) \|_F^2$$
  – Integrity attack
    $$R^{in}_{j_0, w}(\tilde{M}, M) = \sum_{i=1}^{m} \sum_{j \in J_0} w(j) \tilde{M}_{ij}$$
  – Hybrid attack
    $$R^{hybrid}_{j_0, w, \mu}(\tilde{M}, M) = \mu_1 R^{av}_{j_0, w}(\tilde{M}, M) + \mu_2 R^{in}(\tilde{M}, M)$$

Goal: $\tilde{M}^* \in \arg\max_{\tilde{M} \in M} R(\tilde{M}, M)$
Data Poisoning Attacks for Factorization Based Collaborative Filtering

• Mimic normal user behaviors
  – Normal users do not pick items uniformly at random

Malicious users that pick rated uniformly at random can be easily identified by running a t-test against a known database consisting of only normal users

– Stochastic gradient Langevin Dynamics (SGLD)

\[
p(\widehat{M}|M) = \frac{p_0(\widehat{M})p(M|\widehat{M})}{p(M)}
\propto \exp\left( - \sum_{i=1}^{m'} \sum_{j=1}^{n} \frac{(\widehat{M}_{ij} - \xi_j)^2}{2\sigma_j^2} + \beta R(\widehat{M}, M) \right)
\]

\[
p_0(\widehat{M}) = \prod_{i=1}^{m'} \prod_{j=1}^{n} \mathcal{N}(\widehat{M}_{ij}; \xi_j, \sigma_j^2)
\]

\[
p(\widehat{M}|M) = \frac{1}{Z} \exp \left( \beta \cdot R(\widehat{M}, M) \right)
\]

[Welling and Teh 2011.]
Data Poisoning Attacks for Factorization Based Collaborative Filtering

- Normal users usually do not rate items uniformly at random

---

**Algorithm 2 Optimizing $\widetilde{M}$ via SGLD**

1: **Input:** Original partially observed $m \times n$ data matrix $M$, algorithm regularization parameter $\lambda$, attack budget parameters $\alpha$, $B$ and $\Lambda$, attacker’s utility function $R$, step size $\{s_t\}_{t=1}^{\infty}$, tuning parameter $\beta$, number of SGLD iterations $T$.

2: **Prior setup:** compute $\xi_j = \frac{1}{m} \sum_{i=1}^{m} M_{ij}$ and $\sigma_j^2 = \frac{1}{m} \sum_{i=1}^{m} (M_{ij} - \xi_j)^2$ for every $j \in [n]$.

3: **Initialization:** sample $\widetilde{M}_{ij}^{(0)} \sim \mathcal{N}(\xi_j, \sigma_j^2)$ for $i \in [m']$ and $j \in [n]$.

4: **for** $t = 0$ to $T$ **do**

5: Compute the optimal solution $\Theta_{\lambda}(\widetilde{M}^{(t)}; M)$.

6: $\widetilde{M}^{(t+1)} = \widetilde{M}^{(t)} + \frac{s_t}{2} \left( \nabla_{\widetilde{M}} \log p(M|\widetilde{M}) \right) + \varepsilon_t$.

7: Update $M^{(t+1)}$ according to Eq. (18).

8: **end for**

9: **Projection:** find $\widetilde{M}^* \in \arg\min_{\widetilde{M} \in M} \| \widetilde{M} - \widetilde{M}^{(t)} \|^2_F$.

Details in the main text.

10: **Output:** $m' \times n$ malicious matrix $\widetilde{M}^*$. 

---
Data Poisoning Attacks for Factorization Based Collaborative Filtering

- Experiments
  - MovieLens dataset: 27,000 movies with 138,000 users
  - P value and RMSE/Average ratings for ALM with different $\beta$
    (a) $\mu_1 = 1, \mu_2 = 0$  (b) $\mu_1 = 0, \mu_2 = 1$
Data Poisoning Attacks for Factorization Based Collaborative Filtering

• Experiments
  – RMSE for ALM with different percentage of malicious profiles
    
    (a) $\mu_1 = 1, \mu_2 = 0$  
    (b) $\mu_1 = 1, \mu_2 = -1$
Poisoning Attack Against SVM

• To maximize the hinge loss on a validation set

\[
\max_{x_c} L(x_c) = \sum_{k=1}^{m} (1 - y_k f_{x_c}(x_k))^+ = \sum_{k=1}^{m} (-g_k)^+
\]

• Gradient ascent

\[
x'_c = x_c + t \cdot \nabla L(x_c)
\]

\[
\frac{dg_k}{dx_c} = \sum_j (Q_{kj} \frac{d\alpha_j}{dx_c}) + y_k \frac{db}{dx_c} + \frac{dQ_{kc}}{dx_c} \alpha_c, \text{ where } Q = yy^T \odot K
\]

How does the SVM solution change during a single update of \( x_c \)
Data Poisoning on Multi-Task Learning (AAAI’18)

Single-task learning (STL)

Multi-task learning (MTL)

Data poisoning on STL:

Data poisoning on MTL:
Computing Optimal Attacks

Formulation: Bilevel Program

\[
\begin{align*}
\text{max} \quad & \sum_{i \in T_{att}} \mathcal{L}(D_i, w^i), \\
\text{s.t.} \quad & \text{Constraints on } \{\hat{D}_i| T_i \in T_{att}\}, \\
\text{min} \quad & \sum_{i' = 1}^{m} \frac{1}{n_{i'} + \hat{n}_{i'}} \mathcal{L}(D_{i'} \cup \hat{D}_{i'}, w_{i'}) \\
& + \frac{\lambda_1}{2} \text{tr}(WW^T) + \frac{\lambda_2}{2} \text{tr}(\Omega^{-1}W^T), \\
\text{s.t.} \quad & \Omega \succeq 0, \text{tr}(\Omega) = 1.
\end{align*}
\]

Maximize loss of targeted tasks

Multi-task learning with poisoned data

Solver: Stochastic Projected Gradient Descent

\[
(\hat{x}_j^i)^{t} \leftarrow \text{Proj}_X((\hat{x}_j^i)^{t-1} + \eta \nabla_{(\hat{x}_j^i)^{t-1}} l((w_{t-1}^p)^\top x_q^p, y_q^p))
\]
**Experimental Results**

**DIRECT attack:**
- Attacker poisons: A
- Target task: A

**INDIRECT attack:**
- Attacker poisons: A
- Target task: B

**Results:**
- Direct attacks are more effective than indirect attacks
- Both Direct attacks and Indirect attacks are more effective than random attacks
Poisoning Attacks for Face Recognition

Adding 5 poisoning samples into the training set is sufficient to mislead the model to predict the poisoning labels for the back-door images.

[Chen, Liu, Li, Song, 2017]
Poisoning Attacks for Face Recognition
Poisoning Attacks for Face Recognition

small  medium  large
Poisoning Attacks for Face Recognition

Deep neural networks are easy to be poisoning attacked.
Training Data Poisoning

Training is under control of the defender

Inference

Test data is not tampered

What the best can the defender do to defend against training data poisoning?
Robust Logistic Regression and Classification

The estimated logistic regression curve (red solid) is far away from the correct one (blue dashed) due to the existence of just one outlier (red circle)

Feng et al. Robust Logistic Regression and Classification, 2013
**Definition 1** (Sub-Gaussian design). We say that a random matrix \( X = [x_1, \ldots, x_n] \in \mathbb{R}^{p \times n} \) is sub-Gaussian with parameter \( \left( \frac{1}{n} \Sigma_x, \frac{1}{n} \sigma_x^2 \right) \) if: (1) each column \( x_i \in \mathbb{R}^p \) is sampled independently from a zero-mean distribution with covariance \( \frac{1}{n} \Sigma_x \), and (2) for any unit vector \( u \in \mathbb{R}^p \), the random variable \( u^\top x_i \) is sub-Gaussian with parameter \( \frac{1}{\sqrt{n}} \sigma_x \).

**Algorithm 1** RoLR

**Input:** Contaminated training samples \( \{(x_1, y_1), \ldots, (x_{n+n_1}, y_{n+n_1})\} \), an upper bound on the number of outliers \( n_1 \), number of inliers \( n \) and sample dimension \( p \).

**Initialization:** Set \( T = 4 \sqrt{\log p/n} + \log n/n \).

**Preprocessing:** Remove samples \( (x_i, y_i) \) whose magnitude satisfies \( \|x_i\| \geq T \).

Solve the following linear programming problem (see Eqn. (3)):

\[
\hat{\beta} = \arg \max_{\beta \in B_2^p} \sum_{i=1}^n [y(\beta, x)](i).
\]

**Output:** \( \hat{\beta} \).
Robust Linear Regression Against Data Poisoning Attack

Main Ideas: a two-phase solution

• Phase 1: Rely on dimension reduction (PCA) to prune non-principal noise in the training data

• Phase 2: In the low-dimensional space, learn a linear model (i.e., PCR)

Liu, Li, Vorobeychik, Oprea, Robust Linear Regression Against Training Data Poisoning. Aisec, 2017.
Main Challenges

- Both of the **two phases** can be the **target** of the training data poisoning **adversary**

- Have **no assumption** on the ground truth distribution
  - ... except assuming they lie in a low-dimensional manifold
What Can Be Achieved

• Prove a **sufficient and necessary** condition on the **exact subspace recovery** problem
  – Provides a criteria that the PCA process cannot be poisoned

• A **bound** on the **expected test error** when the training data is poisoned up to **γ poisoning rate**
  – i.e., inject up to γN poisoning samples into the pristine training data of N samples
Which line fits the data better?
Answer: democracy!
What about now?
Observation 1: When $\gamma \geq 1$, it is **impossible** to distinguish the poisoning samples from the pristine ones.
What is the mean of the data distribution?
How can a data poisoning adversary efficiently fool the mean estimator?
Answer: leveraging the pristine data!
Answer: leveraging the pristine data
Answer: leveraging the pristine data
Observation 2: the data poisoning adversary can fool a machine learning algorithm if and only if there is a portion of the pristine data that he can leverage.
Sub-space Recovery Problem

• Problem Definition 1 (Subspace Recovery). Design an algorithm $L_{recovery}$, which takes as input $X$, and returns a set of vectors $B$ that form the basis of $X_\star$

• Notation:
  – $X$: observed (poisoned) feature matrix
  – $X_\star$: the pristine feature matrix
  – $X_0$: the pristine feature matrix with noise
    • $X_0 = X_\star + noise$
Noise residual and sub-matrix residual

- Noise residual $NR(X_0)$ optimizes
  $$\min_{X'} \| X_0 - X' \|$$
  s. t. $\text{rank}(X') \leq k$

- Sub-matrix residual $SR(X_0)$ optimizes
  $$\min_{I, B, U} \| X_0^I - U\bar{B} \|$$
  s. t. $\text{rank}(\bar{B}) = k, \bar{B}\bar{B}^T = I_k, X_\star \bar{B}^T\bar{B} \neq X_\star$
  $$I \subseteq \{1,2,\ldots,n\}, |I| = (1 - \gamma)N$$
Sufficient and necessary condition

• Theorem. If $SR(X_0) \leq NR(X_0)$, then no algorithm solves problem 1 with a probability greater than $1/2$.

• If $SR(X_0) > NR(X_0)$, then Algorithm 2 solves problem 1.

---

**Algorithm 2** Exact recovery algorithm for Problem 1

Solve the following optimization problem and get $\mathcal{I}$.

\[
\min_{\mathcal{I}, L} ||X^\mathcal{I} - L|| \\
\text{s.t. rank}(L) \leq k, \mathcal{I} \subseteq \{1, \ldots, n + n_1\}, |\mathcal{I}| = n
\]  

\text{return} a basis of $X^\mathcal{I}$.  

\[\text{(3)}\]
Trimmed Principal Component Regression

- TPCR Lemma. Algorithm 3 returns $\hat{\beta}$, such that for any real value $h > 1$, with at least probability of $1 - ch^{-2}$ for some constant $c$, we have

$$E \left[ \left( x(\hat{\beta} - \beta^*) \right)^2 \right] \leq 4\sigma^2 \left( 1 + \sqrt{\frac{1}{1 - \gamma}} \right)^2 \log c$$

\begin{algorithm}
\textbf{Algorithm 3 Trimmed Principal Component Regression}
\vspace{.5em}
\textbf{Input:} $X, y$
\vspace{.5em}
\begin{enumerate}
  \item Use Algorithm 2 to compute a basis from $X$, and orthogonalize it to get $B$.
  \item Project $X$ onto the span space of $B$ and get $U \leftarrow XB^T$.
  \item Solve the following minimization problem to get $\hat{\beta}_U$
  \begin{equation}
  \min_{\beta_U} \sum_{j=1}^{n} \{(y_i - u_i \beta_U)^2 \text{ for } i = 1, \ldots, n + n_1}\}
  \end{equation}
  where $z(j)$ denotes the $j$-th smallest element in sequence $z$.
\end{enumerate}
\textbf{return} $\hat{\beta} \leftarrow B\hat{\beta}_U$.
\end{algorithm}
Efficient Algorithm using Alternative Minimization

• Problem: minimize the objective in the following form:

\[
\min_{\theta} \sum_{j=1}^{n} \{ l(y_i, f_{\theta}(x_i)) \mid i = 1, \ldots, n + n_1 \}_{(j)}
\]

• Strategy: iteratively do the following two steps until convergence
  – Find the subset of \( \{j\} \) of size \( n \) that minimizes \( l \left( y_j, f_{\theta}(x_j) \right) \)
  – Minimize the total loss with respect to \( \theta \)
Sub-space recover experiments (synthetic data)
Robust regression experiments

Real malicious domain dataset
Takeaways

Message 1. The poisoning attacker can leverage pristine data distribution to construct strong attacks.

Message 2. When the poisoning ratio is not sufficiently large, we can bound the loss on the computed estimator.
Adversarial Machine Learning

• Adversarial machine learning:
  – Learning in the presence of adversaries

• Inference time: adversarial example fools learning system
  – Evasion attacks
    • Evade malware detection; fraud detection

• Training time:
  – Attacker poisons training dataset (e.g., poison labels) to fool learning system to learn wrong model
    • Poisoning attacks: e.g., Microsoft’s Tay twitter chatbot
  – Attacker selectively shows learner training data points (even with correct labels) to fool learning system to learn wrong model
  – Data poisoning is particularly challenging with crowd-sourcing & insider attack
  – Difficult to detect when the model has been poisoned

• Adversarial machine learning particularly important for security critical system
Collaborators
Robust Smart Home

Privacy-Preserving Data Analysis

Topic of Workflow Analysis

Game Theoretic Auditing System for EMR

Large-Scale Auditing Game With Human In the Loop

Privacy Protected Mobile Healthcare

Robust Face Recognition Against Poisoning Attack

Thank You!
Bo Li
crystalboli@berkeley.edu

http://www.crystal-boli.com/