Security and Privacy in Machine Learning

Nicolas Papernot

Work done at the Pennsylvania State University and Google Brain

July 2018 - MSR AI summer school
Machine learning brings social disruption at scale

Healthcare
Source: Peng and Gulshan (2017)

Energy
Source: Deepmind

Transportation
Source: Google

Education
Source: Gradescope
Machine learning is not magic (training time)
Machine learning is not magic (inference time)
Machine learning is deployed in adversarial settings

Tay chatbot

Training data poisoning

YouTube filtering

Content evades detection at inference
Machine learning does not always generalize well (1/2)

Training data

Test data
What if the adversary systematically poisoned the data?

A small perturbation to one training example:

Can change multiple test predictions:

Orig (confidence): Dog (97%)  
New (confidence): Fish (97%)  

Dog (98%)  
Fish (93%)  
Dog (98%)  
Fish (87%)  
Dog (99%)  
Fish (63%)  
Dog (98%)  
Fish (52%)

(Understanding Black-box Predictions via Influence Functions, Koh and Liang)
What if the adversary systematically evaded at inference time?

\[ x \times \text{"panda"} \]

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

\[ \text{"nematode"} \]

8.2% confidence

\[ x + \varepsilon \text{sign}(\nabla_x J(\theta, x, y)) \]

\[ \text{"gibbon"} \]

99.3% confidence

(Goodfellow et al., 2014)

Biggio et al., Szegedy et al., Goodfellow et al., Papernot et al., ...
Machine learning does not always generalize well (2/2)

Training data  

Test data  

Membership inference attack (Shokri et al.)
What is the threat model?

Adversarial goal

Training
- Data integrity
- Model integrity
- Data confidentiality
- Data privacy

Inference
- Model integrity
- Data confidentiality
- Model confidentiality
- Data Privacy

Attack / defense example

- Data poisoning
  (Koh and Liang, 2017)
- Backdoor
  (Gu et al., 2017)
- Federated learning
  (McMahan, 2017)
- RAPPOR
  (Erlingsson, 2014)
- Adversarial examples
  (Szegedy et al., 2013)
- CryptoNets
  (Dowlin et al., 2016)
- Model extraction
  (Tramer et al., 2016)
- Membership inference
  (Shokri et al., 2017)

Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman
Attacking Machine Learning Integrity with Adversarial Examples
The threat model

Attacker may see the model: attacker needs to know details of the machine learning model to do an attack --- aka a **white-box attacker**

Attacker may not see the model: attacker who knows very little (e.g. only gets to ask a few questions) --- aka a **black-box attacker**
Jacobian-based Saliency Map Approach (JSMA)

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami
Adversarial examples...  

... beyond deep learning

Logistic Regression  
Nearest Neighbors  
Support Vector Machines  
Decision Trees

Useful to think about definitions and threat model

Adversarial Attacks on Neural Network Policies [arXiv preprint]  
Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, Pieter Abbeel

Adversarial Perturbations Against Deep Neural Networks for Malware Classification [ESORICS 2017]  
Kathrin Grosse, Nicolas Papernot, Praveen Manoharan, Michael Backes, Patrick McDaniel
The threat model

Attacker may see the model: attacker needs to know details of the machine learning model to do an attack --- aka a white-box attacker

Attacker may not see the model: attacker who knows very little (e.g. only gets to ask a few questions) --- aka a black-box attacker
Attacking remotely hosted black-box models

(1) The adversary queries remote ML system for labels on inputs of its choice.
Attacking remotely hosted black-box models

(2) The adversary uses this labeled data to train a local substitute for the remote system.
(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.
### Cross-technique transferability

<table>
<thead>
<tr>
<th>Source Machine Learning Technique</th>
<th>DNN</th>
<th>LR</th>
<th>SVM</th>
<th>DT</th>
<th>kNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN</td>
<td>38.27</td>
<td>23.02</td>
<td>64.32</td>
<td>79.31</td>
<td>8.36</td>
</tr>
<tr>
<td>LR</td>
<td>6.31</td>
<td>91.64</td>
<td>91.43</td>
<td>87.42</td>
<td>11.29</td>
</tr>
<tr>
<td>SVM</td>
<td>2.51</td>
<td>36.56</td>
<td>100.0</td>
<td>80.03</td>
<td>5.19</td>
</tr>
<tr>
<td>DT</td>
<td>0.82</td>
<td>12.22</td>
<td>8.85</td>
<td>89.29</td>
<td>3.31</td>
</tr>
<tr>
<td>kNN</td>
<td>11.75</td>
<td>42.89</td>
<td>82.16</td>
<td>82.95</td>
<td>41.65</td>
</tr>
</tbody>
</table>

**Target Machine Learning Technique**

Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples [arXiv preprint]
Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow

22
Properly-blinded attacks on real-world remote systems

<table>
<thead>
<tr>
<th>Remote Platform</th>
<th>ML technique</th>
<th>Number of queries</th>
<th>Adversarial examples misclassified (after querying)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MetaMind</td>
<td>Deep Learning</td>
<td>6,400</td>
<td>84.24%</td>
</tr>
<tr>
<td>Amazon Web Services™</td>
<td>Logistic Regression</td>
<td>800</td>
<td>96.19%</td>
</tr>
<tr>
<td>Google Cloud Platform</td>
<td>Unknown</td>
<td>2,000</td>
<td>97.72%</td>
</tr>
</tbody>
</table>

All remote classifiers are trained on the MNIST dataset (10 classes, 60,000 training samples)
Defending against adversarial examples
Learning models robust to adversarial examples is hard

Error spaces containing adversarial examples are large

Learning or detecting adversarial examples creates an arms race

Is attacking machine learning easier than defending it? [Blog post at www.cleverhans.io]
Ian Goodfellow and Nicolas Papernot
What makes a successful deep neural network?

<table>
<thead>
<tr>
<th>Layer name</th>
<th>Neural architecture</th>
<th>Representation spaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Softmax</td>
<td>![Softmax diagram]</td>
<td>![Panda representation]</td>
</tr>
<tr>
<td>3rd hidden</td>
<td>![3rd hidden diagram]</td>
<td>![Panda representation]</td>
</tr>
<tr>
<td>2nd hidden</td>
<td>![2nd hidden diagram]</td>
<td>![Panda representation]</td>
</tr>
<tr>
<td>1st hidden</td>
<td>![1st hidden diagram]</td>
<td>![Panda representation]</td>
</tr>
<tr>
<td>Inputs</td>
<td>![Input diagram]</td>
<td>![Panda representation]</td>
</tr>
</tbody>
</table>
What makes a successful adversarial example?

<table>
<thead>
<tr>
<th>Layer name</th>
<th>Neural architecture</th>
<th>Representation spaces</th>
<th>Nearest neighbors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Softmax</td>
<td><img src="image" alt="Softmax diagram" /></td>
<td><img src="image" alt="Softmax representation" /></td>
<td><img src="image" alt="Nearest neighbors" /></td>
</tr>
<tr>
<td>3rd hidden</td>
<td><img src="image" alt="3rd hidden diagram" /></td>
<td><img src="image" alt="3rd hidden representation" /></td>
<td><img src="image" alt="Nearest neighbors" /></td>
</tr>
<tr>
<td>2nd hidden</td>
<td><img src="image" alt="2nd hidden diagram" /></td>
<td><img src="image" alt="2nd hidden representation" /></td>
<td><img src="image" alt="Nearest neighbors" /></td>
</tr>
<tr>
<td>1st hidden</td>
<td><img src="image" alt="1st hidden diagram" /></td>
<td><img src="image" alt="1st hidden representation" /></td>
<td><img src="image" alt="Nearest neighbors" /></td>
</tr>
<tr>
<td>Inputs</td>
<td><img src="image" alt="Input diagram" /></td>
<td><img src="image" alt="Input representation" /></td>
<td><img src="image" alt="Nearest neighbors" /></td>
</tr>
</tbody>
</table>
Nearest neighbors indicate support from training data...

<table>
<thead>
<tr>
<th>Layer name</th>
<th>Neural architecture</th>
<th>Representation spaces</th>
<th>Nearest neighbors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Softmax</td>
<td><img src="image" alt="Softmax diagram" /></td>
<td><img src="image" alt="Representation spaces diagram" /></td>
<td><img src="image" alt="Nearest neighbors diagram" /></td>
</tr>
<tr>
<td>3rd hidden</td>
<td><img src="image" alt="3rd hidden diagram" /></td>
<td><img src="image" alt="Representation spaces diagram" /></td>
<td><img src="image" alt="Nearest neighbors diagram" /></td>
</tr>
<tr>
<td>2nd hidden</td>
<td><img src="image" alt="2nd hidden diagram" /></td>
<td><img src="image" alt="Representation spaces diagram" /></td>
<td><img src="image" alt="Nearest neighbors diagram" /></td>
</tr>
<tr>
<td>1st hidden</td>
<td><img src="image" alt="1st hidden diagram" /></td>
<td><img src="image" alt="Representation spaces diagram" /></td>
<td><img src="image" alt="Nearest neighbors diagram" /></td>
</tr>
<tr>
<td>Inputs</td>
<td><img src="image" alt="Inputs diagram" /></td>
<td><img src="image" alt="Representation spaces diagram" /></td>
<td><img src="image" alt="Nearest neighbors diagram" /></td>
</tr>
</tbody>
</table>
Deep k-Nearest Neighbors (DkNN) classifier

1. Searches for **nearest neighbors** in the training data at each layer
2. Estimates the **nonconformity** of input $x$ for each possible label $y$
3. Apply conformal prediction to compute:
   a. **Confidence**
      
      “How likely is the prediction given the training data?”
   b. **Credibility**
      
      “How relevant is the training data to the prediction?”
Example applications of DkNN credibility

**Adversarial examples**

- Clean inputs
- Basic Iterative Method
- Carlini & Wagner

**Out of distribution**

- cifar (softmax)
- rotate (softmax)
- cifar (dknn)
- rotate (dknn)

**Mislabeled inputs**

- Dataset: 5
  - DkNN: 1
    - Image 1
  - Dataset: 5
    - DkNN: 7
    - Image 2
  - Dataset: 1
    - DkNN: 1
    - Image 3
  - Dataset: 2
    - DkNN: 1
    - Image 4
  - Dataset: 1
    - DkNN: 2
    - Image 5
Implications for the attacker and defender

**Attacker**

**Defender**

Reject low credibility predictions:

-> explicit tradeoff between clean accuracy and adversarial accuracy

Active learning: more training data through human labeling of rejected predictions

Contributes to breaking “black-box” myth
Some (surprising) connections to fairness & interpretability

Adversarial Examples that Fool both Human and Computer Vision [arXiv preprint]
Gamaleldin F. Elsayed, Shreya Shankar, Brian Cheung, Nicolas Papernot, Alex Kurakin, Ian Goodfellow, Jascha Sohl-Dickstein
Machine Learning with Privacy
Types of adversaries and our threat model

Model querying (black-box adversary)
Shokri et al. (2016) *Membership Inference Attacks against ML Models*
Fredrikson et al. (2015) *Model Inversion Attacks*

Model inspection (white-box adversary)
Zhang et al. (2017) *Understanding DL requires rethinking generalization*

In our work, the threat model assumes:
- Adversary can make a potentially unbounded number of queries
- Adversary has access to model internals
A definition of privacy: *differential privacy*

Randomized Algorithm

Answer 1
Answer 2
... 
Answer n

Randomized Algorithm

Answer 1
Answer 2
... 
Answer n
Private Aggregation of Teacher Ensembles (PATE)

Sensitive Data

Partition 1 → Teacher 1
Partition 2 → Teacher 2
Partition 3 → Teacher 3
... → ...
Partition n → Teacher n

Training → Data flow

Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data [ICLR 2017 best paper]
Nicolas Papernot, Martin Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar
Aggregation

Count votes

\[ n_j(\vec{x}) = |\{i : i \in 1..n, f_i(\vec{x}) = j\}| \]

Take maximum

\[ f(x) = \arg \max_j \left\{ n_j(\vec{x}) \right\} \]
Intuitive privacy analysis

If most teachers agree on the label, it does not depend on specific partitions, so the privacy cost is small.

If two classes have close vote counts, the disagreement may reveal private information.
Noisy aggregation

Count votes

\[ n_j(x) = |\{ i : i \in 1..n, f_i(x) = j \}| \]

Add Laplacian noise

\[ Lap \left( \frac{1}{\varepsilon} \right) \]

Take maximum

\[ f(x) = \arg \max_j \left\{ n_j(x) + Lap \left( \frac{1}{\varepsilon} \right) \right\} \]
Teacher ensemble

Sensitive Data

Partition 1 → Teacher 1
Partition 2 → Teacher 2
Partition 3 → Teacher 3
... → ... → Aggregated Teacher
Partition n → Teacher n

Training → Data flow
Why train an additional “student” model?

The aggregated teacher violates our threat model:

1. Each prediction increases total privacy loss.
   Privacy budgets create a tension between the accuracy and number of predictions.

2. Inspection of internals may reveal private data.
   Privacy guarantees should hold in the face of white-box adversaries.
Student training

Sensitive Data

Partition 1

Partition 2

Partition 3

Partition n

Teacher 1

Teacher 2

Teacher 3

Teacher n

Aggregated Teacher

Student

Queries

Available to the adversary

Not available to the adversary

Public Data

Sensitive Data

Training

Inference

Data flow
Deployment

Available to the adversary

Student

Queries

Inference
Differential privacy analysis

Differential privacy:
A randomized algorithm $M$ satisfies $(\varepsilon, \delta)$ differential privacy if for all pairs of neighbouring datasets $(d, d')$, for all subsets $S$ of outputs:

$$Pr[M(d) \in S] \leq e^{\varepsilon} Pr[M(d') \in S] + \delta$$

Application of the **Moments Accountant** technique (Abadi et al, 2016)

Strong **quorum** $\Rightarrow$ Small privacy cost

Bound is **data-dependent**: computed using the empirical quorum
Trade-off between student accuracy and privacy
Synergy between utility and privacy

1. Check privately for consensus
2. Run noisy argmax only when consensus is sufficient
Trade-off between student accuracy and privacy

![Graph showing the trade-off between student accuracy and privacy with Selective PATE highlighted.](image)
Economist Charles Goodhart posited in 1975 that …

“When a measure becomes a target, it ceases to be a good measure”

As ML models make more and more decisions, we will have to satisfy them, and they will become targets.
Thank you for listening!