Adversarial Machine Learning
—An Introduction

With slides from: Binghui Wang
Outline

• Machine Learning (ML)
• Adversarial ML
• Attack
  • Taxonomy
  • Capability
• Adversarial Training
• Conclusion
Outline

• Machine Learning (ML)
• Adversarial ML
• Attack
  • Taxonomy
  • Capability
• Adversarial Training
• Conclusion
Machine Learning (ML)

• Define ML Tasks
  • Supervised, semi-supervised, unsupervised, reinforcement learning

• Data Collection and Preprocessing
  • Sensors, camera, I/O, etc;

• Apply ML Algorithm
  • Training phase: Learn ML Model (Parameter and Hyperparameter Learning)
  • Testing (Inference) phase: Inference on unseen data.

• Theoretical Support: PAC Model of Learning
ML Is Ubiquitous

• Cancer diagnosis
• Self-driving cars
• Unmanned aerial vehicle
• Surveillance and access-control
• ...
Outline

• Machine Learning (ML)
• Adversarial ML
• Attack
  • Taxonomy
  • Capability
• Adversarial Training
• Conclusion
What Do You See
What Do You See Now

[Szegedy et al., ICLR '14]
What Do You See Now
Adversarial ML

• A research field that lies at the intersection of ML and computer security (e.g., biometric authentication, network intrusion detection, and spam filtering).

• ML algorithms in real-world applications mainly focus on effective or/and efficient, while few techniques and design decisions keep the ML models secure and robust!

• Adversarial ML: ML in adversarial settings.

• Attack is a major component.
Outline

• Machine Learning (ML)
• Adversarial ML
• Attack
  • Taxonomy
  • Capability
• Adversarial Training
• Conclusion
Attack

• Attack Taxonomy
  • **Poisoning (Causative) Attack**: Attack on training phase. Attackers attempt to learn, influence, or corrupt the ML model itself.
Attack

• Attack Taxonomy
  • **Evasion (Exploratory) Attack**: Attack on **testing** phase. Do not tamper with ML model, but instead cause it to produce adversary selected outputs.
Attack

• Attack Taxonomy
  • **Model Inversion Attack**: Extract private and sensitive inputs by leveraging the outputs and ML model.
  • **Model Extraction Attack**: Extract model parameters via querying the model.

![Image]

Figure 1: An image recovered using a new model inversion attack (left) and a training set image of the victim (right). The attacker is given only the person’s name and access to a facial recognition system that returns a class confidence score.

<table>
<thead>
<tr>
<th>Model</th>
<th>OHE</th>
<th>Binning</th>
<th>Queries</th>
<th>Time (s)</th>
<th>Price ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circles</td>
<td>-</td>
<td>Yes</td>
<td>278</td>
<td>28</td>
<td>0.03</td>
</tr>
<tr>
<td>Digits</td>
<td>-</td>
<td>No</td>
<td>650</td>
<td>70</td>
<td>0.07</td>
</tr>
<tr>
<td>Iris</td>
<td>-</td>
<td>Yes</td>
<td>644</td>
<td>68</td>
<td>0.07</td>
</tr>
<tr>
<td>Adult</td>
<td>Yes</td>
<td></td>
<td>1,485</td>
<td>149</td>
<td>0.15</td>
</tr>
</tbody>
</table>

Table 7: Results of model extraction attacks on Amazon. OHE stands for one-hot-encoding. The reported query count is the number used to find quantile bins (at a granularity of $10^{-3}$), plus those queries used for equation-solving. Amazon charges $0.0001 per prediction [1].
Evasion Attack (Most Common)

• The most common attack. It can be further classified into

• **White-Box**: Attackers know full knowledge about the ML algorithm, ML model, (i.e., parameters and hyperparameters), architecture, etc.

• **Black-Box**: Attackers almost know nothing about the ML system (perhaps know number of features, ML algorithm).
White-Box Evasion Attack

• Given a function (LogReg, SVM, DNN, etc) $F : X \mapsto Y$, where $X$ is a input feature vector, and $Y$ is an output vector.

• An attacker expects to construct an adversarial sample $X^*$ from $X$ by adding a perturbation vector $\delta_X$ such that

$$\arg\min_{\delta_X} ||\delta_X|| \text{ s.t. } F(X + \delta_X) = Y^*$$

• where $X^* = X + \delta_X$ and $Y^*$ is the desired adversarial output.

• Solving this problem is non-trivial, when $F$ is nonlinear or/and nonconvex.
White-Box Evasion Attack

• Approximate Solution: Jacobian-based Data Augmentation
  • **Direction Sensitivity Estimation**: Evaluate the sensitivity of model F at the input point corresponding to sample X
    \[ \nabla F(X) = \frac{\partial F(X)}{\partial X} = \left[ \frac{\partial F_j(X)}{\partial x_i} \right]_{i=1..M, j=1..N} \nabla \\
    \]
  • **Perturbation Selection**: Select perturbation affecting sample X’s classification

• Other Solutions
  • Fast sign gradient method
  • DeepFool
  • ...
White-Box Evasion Attack

Fig. 3: 

Adversarial crafting framework: Existing algorithms for adversarial sample crafting [7], [9] are a succession of two steps: (1) direction sensitivity estimation and (2) perturbation selection. Step (1) evaluates the sensitivity of model $F$ at the input point corresponding to sample $X$. Step (2) uses this knowledge to select a perturbation affecting sample $X$'s classification. If the resulting sample $X + \delta X$ is misclassified by model $F$ in the adversarial target class (here 4) instead of the original class (here 1), an adversarial sample $X^*$ has been found. If not, the steps can be repeated on updated input $X \leftarrow X + \delta X$. 
White-Box Evasion Attack
Black-Box Evasion Attack

• Adversarial Sample Transferability
  • Cross model transferability: The same adversarial sample is often misclassified by a variety of classifiers with different architectures
  • Cross training-set transferability: The same adversarial sample is often misclassified trained on different subsets of the training data.

• Therefore, an attacker can
  • First train his own (white-box) substitute model
  • Then generate adversarial samples
  • Finally, apply the adversarial samples to the target ML model
Black-Box Evasion Attack

Figure 3: **Training of the Substitute DNN Architecture** $F$: the attacker (1) collects an initial substitute training set $S_0$ and (2) selects a substitute architecture $F$. Using the oracle $\tilde{O}$, the attacker (3) labels $S_0$ and (4) trains substitute DNN $F$. After (5) Jacobian-based dataset augmentation, steps (3) through (5) are repeated for several substitute epochs $\rho$. 

Mathematical expression: 

$$S_{\rho+1} = \{\bar{x} + \lambda_{\rho+1} \cdot \text{sgn}(J_F[\tilde{O}(\bar{x})]) : \bar{x} \in S_\rho \} \cup S_\rho$$
Outline

• Machine Learning (ML)
• Adversarial ML
• Attack
  • Taxonomy
  • Capability
• Adversarial Training
• Conclusion
Adversarial Training

- Adversarial samples can cause any ML algorithm fail to work.
- However, they can be leveraged to build a more accurate model.
- Called adversarial training: learning with a adversary.
- A two-player game.
Adversarial Training

• Min-max objective function

$$\min_{\theta} \max_{\epsilon: \|\epsilon\|_p \leq \sigma} \mathcal{L}(x + \epsilon; \theta)$$

• Unified gradient regularization framework

$$\min_{\theta} \mathcal{L}(x) + \sigma \|\nabla_x \mathcal{L}\|_{p^*}$$
Outline

• Machine Learning (ML)
• Adversarial ML
• Attack
  • Taxonomy
  • Capability
• Adversarial Training
• Conclusion
Conclusion

• ML algorithms and methods are vulnerable to many types of attack.
• Adversarial examples shows its transferability in ML models, i.e., either cross-models (inter or intra) or cross-training sets.
• However, adversarial examples can be leveraged to improve the performance or the robustness of ML models.