Adversarial Attacks on Machine Learning Models

Leo de Castro[1], Julian Chacon-Castano[2], Mohamadou B. Bah[2]

{ldec, julianch, bellabah}@mit.edu

[1]MIT CSAIL, [2]MIT Electrical Engineering and Computer Science

6.867 Fall 2019

Motivation

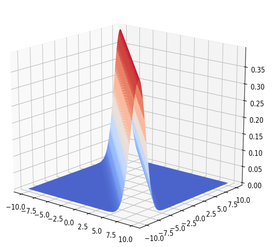
For a model trained on a private dataset, how much information about the dataset is leaked when an adversary has black-box access to the model?

In broad terms, our project explores adversarial attacks on machine learning models. We do so first in a toy "gaussian" setting, then in an image CIFAR-10 setting.

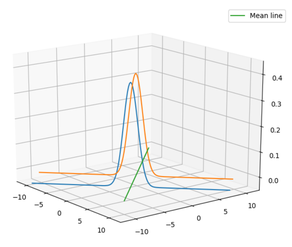
- Toy setting: a "adversary" attempts to uncover a distribution hidden by a "challenger".
- Image Setting: training data reconstruction. We aim to reconstruct a single datapoint (image) within the dataset that the target model was trained on.

Model Hiding, Toy Setting

- We introduce the concept of a Gauss2D surface, which in $\mathbb{R}^3$ is Gaussian in one direction, and uniform along the other.
- Naturally, we can define slices along the mean line (green); we refer to these slices as training slices. A training distribution $\mathcal{D}$ is instantiated from a slice s via $\mathcal{D} = \mathcal{D}(s)$



Gauss 2D Surface



Training Slices

Maximum Likelihood Estimation of Gaussian Parameters Under Noise

Given N noisy samples $\{y_i\}_{i=1}^N$, $Y = X + \xi$ where the samples are sampled from an underlying Gaussian distribution $X \sim \mathcal{N}(\mu_x, \sigma_x^2)$ subject to additive zero-mean Gaussian noise $\zeta \sim \mathcal{N}(0, \sigma_e^2)$, how can we recover μ_x, σ_x^2?

Recognize that

$$Z \sim \mathcal{N}(\mu_z, \sigma_z^2) = \mathcal{N}(\mu_x, \sigma_x^2 + \sigma_e^2)$$

In which case, via the MLE of the Gaussian parameters:

$$\widetilde{\mu_x} = \mu_z = \frac{1}{N}\sum_{i=1}^{N} y_i \qquad \widetilde{\sigma_x^2} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \mu_x)^2 - \sigma_e^2$$

Model Hiding Results

A single round between a challenger $\mathcal{C}$ and adversary $\mathcal{A}$ is as follows:

(1) $\mathcal{A}$ selects two slices s_0 and s_1 that define two training distributions $\mathcal{D}_0$ and $\mathcal{D}_1$
(2) $\mathcal{A}$ sends $\mathcal{D}_0$ and $\mathcal{D}_1$ to $\mathcal{C}$
(3) $\mathcal{C}$ selects a bit b uniformly at random and trains a model $\mathcal{M}$ using the training distribution $\mathcal{D}_b$
(4) $\mathcal{C}$ sends $\mathcal{M}$ to $\mathcal{A}$
(5) $\mathcal{A}$ makes a constant number of black-box queries on $\mathcal{M}$. $\mathcal{A}$ produces a bit b'
(6) $\mathcal{A}$ wins if $b' = b$ and loses otherwise

$\mathcal{A}$ is assessed by the number of rounds it wins out of K rounds

Standard model – Learning Gauss Estimator

Using MLE, we obtain estimates for μ_x, σ_x^2. But to fully parametrize a training slice s_i, we need to pinpoint the single point over the mean line. This is a regression problem of the form $\boldsymbol{A}c = \boldsymbol{b}$, where we solve for c:

$$\{y_i\}_{i=1}^N = \{(\boldsymbol{Y}_{i,1}, \boldsymbol{Y}_{i,2})\}_{i=1}^N, \boldsymbol{Y} \in \mathbb{R}^{N\times 2}$$

$$\left[\boldsymbol{Y}_{:,1}, \boldsymbol{1}_{N\times 1}\right] c = \boldsymbol{Y}_{:,2}$$

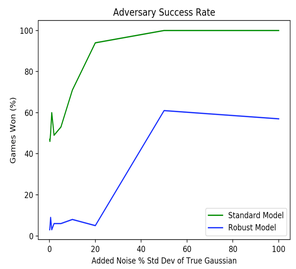
$c, \widetilde{\mu_x}, \widetilde{\sigma_x^2}$ is a full parametrization of an estimated training slice $\widetilde{s_i}$, which can be used to predict the PDF of a point $\widetilde{s_i}(y_i) \equiv PDF(y_i)$

- The "uniform" mean loss between an estimated $\widetilde{s_i}$ and s_i is given by

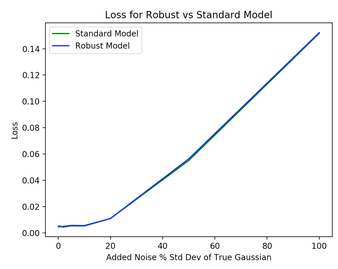
$$\frac{1}{N}\sum_{k=1}^{N} |\widetilde{s_i}(y_k) - s_i(y_k)|$$

Robust Model – Robust Learning Gauss Estimator

- For the model for $\widetilde{s_i}$, define a sphere around its estimated center.
- When an adversary is querying points, if the challenger recognizes that the adversary is "close" because it is picking points with high informational content, i.e. the points cross some threshold, then:
    - The challenger adds $\mathcal{N}(0, 0.1)$ noise to the prediction $\widetilde{s_i}(\cdot)$ to thwart the adversary.
- Of course, with this fixed strategy, a prepared adversary under no query limit could detect when the challenger is perturbing the PDF values it sends.



Percentage of Games Won by Adversary vs. % Noise Std. Dev

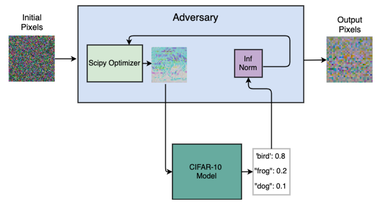


Uniform loss for Robust and Standard Model vs. % Noise Std. Dev

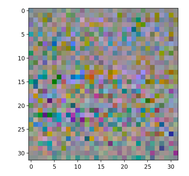
Image Setting – Reconstruction

Within this setting is a Target model T, a pretrained ResNet on CIFAR-10 dataset D.

- T: $\mathbb{R}^{32\times 32\times 3} \to \mathbb{R}^{10}$, the model T outputs a SoftMax vector $z \in \mathbb{R}^{10}, \sum_i z_i = 1$ for the 10 classes $\{c_1, \ldots, c_{10}\}$ in D when given RGB image x.
- This attack involves an adversary $\mathcal{A}$ optimizing over pixels in an image to maximize the l_∞ norm of z
- The optimizer of choice is Powell's Method, which finds the local minimum of some loss $\mathcal{L}$ without taking derivatives, and without needing $\mathcal{L}$ be differentiable.

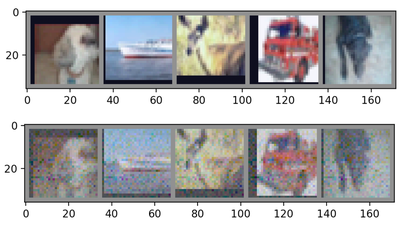


Training-Data Reconstruction Pipeline



Sample Optimizer Output x^*

- The procedure is seeded with an initial guess x' of pink-noise and attempts to converge on a "sensible image" x^* after $K = 200$ iterations.
- Although $x^* \notin D$, interestingly, the pretrained classifier classifies x^* as some class $c_i = \text{argmax}(T(x^*))$ with a confidence > 0.95
- When $\mathcal{A}'s$ output (classified as c_1) is added to another image x_p(classified as c_{10}), it causes T to misclassify x_p as c_1.



The resulting classifications after adding $\mathcal{A}'s$ output to the "clean" image. Inspired by perturbation attacks on machine learning models explored in the literature.

Target Model Classification On Input Images						
Clean	dog	ship	deer	truck	dog
Added Noise	bird	airplane	bird	truck	bird

Conclusion

Certain models without deliberate defense mechanisms are vulnerable to information leaks, even when black-boxed.

- Limited number of queries
- Adversary detection -- how to determine whether an agent is malicious
- Returning values at random

Future Directions

- Experiment with randomized decision boundaries in the robust model
- Quantitively describe how number of queries affects adversary performance.

Presented as our Final Project for 2019 Fall 6.867 Machine Learning course at MIT