Adversarial Classification

Jonas Geiselhart

Institute for Information Security

Seminar Informationssicherheit und Kryptographie, Juli 2021

Main Source: Dalvi et al. 2004

The Problem

- Maliciously engineered inputs, called adversarial input, can lead to wrong results in machine learning algorithms, without being noticable for humans.

Adversarial Example: From “Stop” to “120km/h”

https://kennysong.github.io/adversarial.js/

The Problem

- Maliciously engineered inputs, called adversarial input, can lead to wrong results in machine learning algorithms, without being noticable for humans.
- In many modern Tasks:
  - Natural language processing
  - Visual classification
  - Audio Recognition
  - Detection of malicious software

Thread Models

- Different Objectives (Classification, Segmentation, ...)
- White Box or Black Box attacks
- Targeted or general wrong classification
- Single or iterative attacks
- Different Perturbation (L_0, L_2, L_∞, ...)

$\rightarrow$ Our Constraint: Single, white box attack

Yuan et al. 2019

Formal Definition

Input as an Instance $X = (X_1, X_2, ..., X_n)$ of all possible Inputs χ.
Input either malicious $+$ or innocent $-$.

Adversarial Classification as Game between Adversary and Classifier on a test set $\mathcal{T}$:

- Classifier tries to predict the correct class $(+/-)$ for the instances of $\mathcal{T}$.
- Adversary tries to modify $\mathcal{T}$, so that the classifier recognizes x' instead of x for the Instance X.

Models

Classifier

- V_i Cost of Measuring X_i
- $U_C(y_c, y)$ Utility of classifying y_c as y

$$U_C = \sum_{(x,y)\in\mathcal{X}\mathcal{Y}} P(x,y)\left[U_C(\mathcal{C}(\mathcal{A}(x)),y) - \sum_{X_i\in\mathcal{X}_C(x)} V_i\right]$$

Adversary

- $W(x, x_i)$ Cost of changing classification x to x_i
- $U_A(y_c, y)$ Utility y_c being classified as y

$$U_A = \sum_{(x,y)\in\mathcal{X}\mathcal{Y}} P(x,y)\left[U_A(\mathcal{C}(\mathcal{A}(x)),y) - W(x,\mathcal{A}(x))\right]$$

$\rightarrow$ U_C & U_A form a Nash-Equilibirum.

Strategies & Implications - Adversarial Strategy

Adversary startegy as $\mathcal{A}(x)$

- Only modify the input if the gain of utility is more than the cost of modifying the instance
- Find algorithm that minimizes $W(x, x')$, but still fools the classifier $\rightsquigarrow$ *minimum cost camouflage* (MCC).
- Given perfect information MCC(x) is $NP - hard$, but can be discretizied and approximated.
- In reality: perfect information is extremely rare

Strategies & Implications - Classifier Strategy

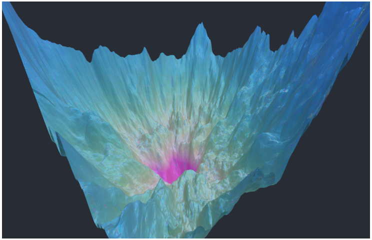
Classifier strategy as $\mathcal{C}(x)$, assumes perfect adversarial strategy

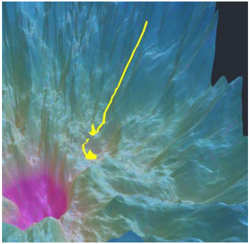
- Classify each test instance as +(tampered) or - (untampered).
- $U(+|x) > U(-|x) \iff$x is a positive instance
  $\rightsquigarrow$ Probabilities of an instance being (not) manipulated necessary
- Approach: $\exists x' : MCC(x') = x$
  - Let GV be an subset of Values of potentially adversarial values
  - If GV is sufficiently large, assume its adversarial ($P_A(X'_A(x')|+)$ is large), else search for $\{x'|\mathcal{A}(x') = x \wedge x' \neq x\}$

- In reality: other measures more successful

Generating Adverserial Input

- A lot of algorithms, with different constraints or objectives.
- Simple, widely used concept: gradient descend.
  - Drastically increase the confidence of one (wrong) feature-prediction.
  - Use iterative Queries to discretize a loss function & find pertubation.





Carlini 2019, https://losslandscape.com/explorer

Classifier Mitigation

ML Perspective:

- Adversarial Training
  - Add adversarial inputs to training set
  - Penalize special (security-critical) pattern learning
- Change loss-function topology

$\rightsquigarrow$ affect the performance of the classifier, do not scale well

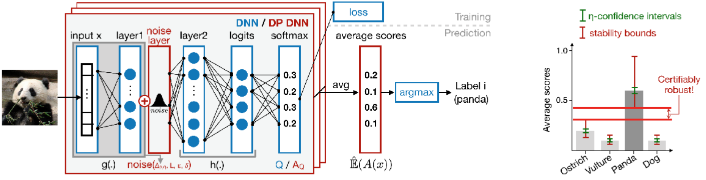
$\rightsquigarrow$ do not enhance theoretical security

Ilyas et al. 2019,Carlini 2019

Classifier Mitigation

Certified robustness approach:

- Differential privacy (e.g. PixelDP for visual classification)
  - Randomize Computation so that small changes in input only have limited (predictable) effect on the end result.



(a) PixelDP DNN Architecture (b) Robustness Test Example

1: **Architecture.** (a) In blue, the original DNN. In red, the noise layer that provides the (ϵ, δ)-DP guarantees. The noise can be a

Aus: www.semanticscholar.org/paper/Certified-Robustness-to-Adversarial-Examples-with-Lécuyer-Atlidakis/3e86a51d1f2051ab8f448b66c6dcc17924d17cfa,
Lecuyer et al. 2019

Carlini 2019, Lecuyer et al. 2019, Tramèr et al. 2020

Classifier Mitigation

Certified robustness approach:

- Differential privacy (e.g. PixelDP for visual classification)
  - Randomize Computation so that small changes in input only have limited (predictable) effect on the end result.
- Provable defences still a very open research field
- Problems:
  - Lack of measuring robustness (perturbation)
  - Generalize to different thread model
  - Scale the approaches

- No great incentive to build rigorous defences

Carlini 2019, Lecuyer et al. 2019, Tramèr et al. 2020

Summary on Security

- No great incentive to build rigorous defences
- Small, limited gurantees come with fundamental trade-offs in general accuracy
- Situations in which adversarial robustness is important:
  - One failure does not matter!
  - Human is affected in interaction.
  - Classifier has to be stable.
- Best effort approach is the best we can do (for all we know).

Tramèr et al. 2020

Overview of future Challenges

- How (well) can we improve robustness?
  - Increase robust accuracy and standard accuracy
- Deeper mechanics of machine learning
  - Feature or bugs ?
  - Robust and non-robust features ?
  - Human cognition
- Generalizing properties of adversarial classification

Ilyas et al. 2019

Sources

- Carlini, N. (Oct. 2019). "On Evaluating Adversarial Robustness". In: *Camlis Keynote: On Evaluating Adversarial Robustness*. Camlis Organisation. URL: https://www.youtube.com/watch?v=-p2il-V-0fk.
- Dalvi, N. et al. (2004). "Adversarial Classification". In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '04. Seattle, WA, USA: Association for Computing Machinery, pp. 99–108. URL: https://doi.org/10.1145/1014052.1014066.
- Ilyas, A. et al. (2019). *Adversarial Examples Are Not Bugs, They Are Features*. arXiv: 1905.02175 [stat.ML].
- Lecuyer, M. et al. (2019). "Certified Robustness to Adversarial Examples with Differential Privacy". In: *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 656–672.
- Tramèr, F. et al. (2020). *Fundamental Tradeoffs between Invariance and Sensitivity to Adversarial Perturbations*. arXiv: 2002.04599 [cs.LG].
- Yuan, X. et al. (2019). "Adversarial Examples: Attacks and Defenses for Deep Learning". In: *IEEE Transactions on Neural Networks and Learning Systems* 30.9, pp. 2805–2824.