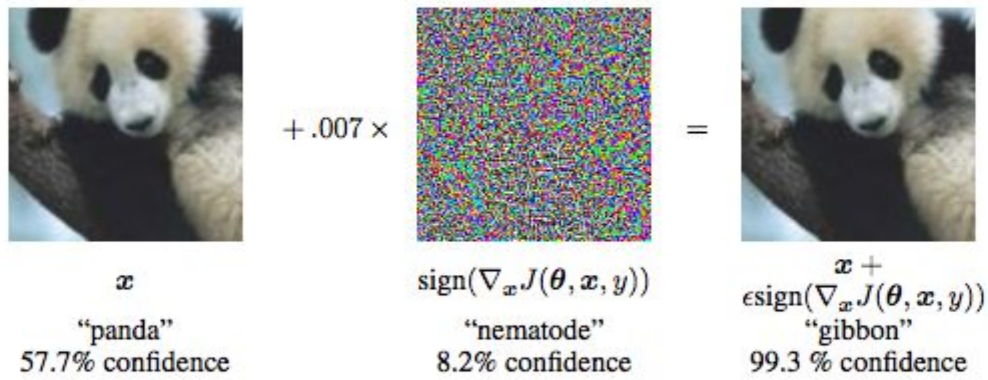
Adversarial Examples Are Not Bugs, They are Features

Michał Królikowski

based on "Adversarial Examples Are Not Bugs, They Are Features" by Ilyas et al.

Adversarial Examples:
high-level overview



- **imperceptibly perturbed** natural inputs that induce **erroneous predictions**

Adversarial Examples:
potential causes

- high dimensionality of the input space
- statistical fluctuations in the data
- peculiarities of the model
- **but what if the cause is natural and they're not a bug?**

Adversarial Examples: **hypothesis**

Adversarial vulnerability is a direct result of our models' sensitivity to well-generalizing features in the data.

Adversarial Examples: **hypothesis**

Adversarial vulnerability is a direct result of our models' sensitivity to well-generalizing features in the data.

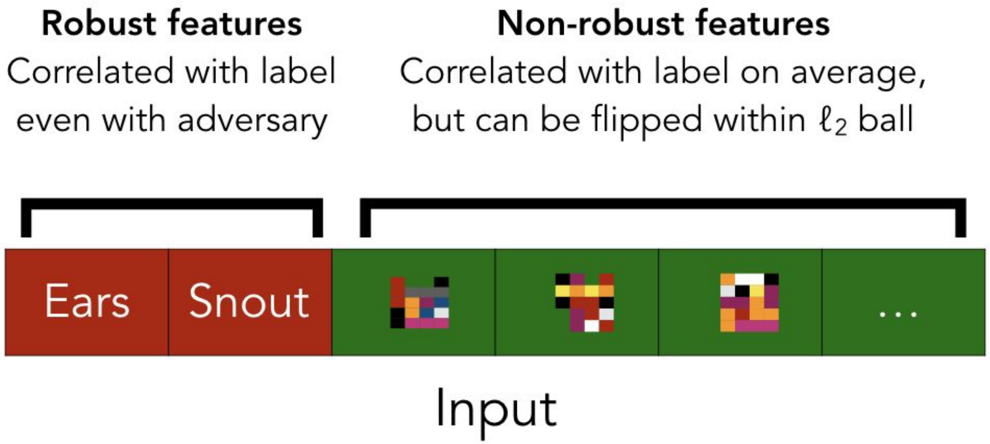
- our models want to get **the best accuracy** by any means necessary…

Adversarial Examples: **hypothesis**

Adversarial vulnerability is a direct result of our models' sensitivity to well-generalizing features in the data.

- our models want to get **the best accuracy** by any means necessary…
- …so maybe they **use some features, that we can't see**?

Robust vs. non-robust features



- idea: there are some features, that are **highly predictive yet imperceptible to humans**
  - machines don’t care if the feature is an “ear” or if it’s a complicated sequence of values

Let’s test it out!

Some definitions

- adversarial dataset
  - dataset modified by adding adversarial perturbations
- standard accuracy
  - accuracy on the non-modified test set
- adversarial accuracy
  - accuracy on the adversarial test set
- adversarial training
  - training the model on the adversarial dataset

First baseline

- train a **standard model** on a standard (non-adversarial) dataset
- test it on the adversarial dataset
- results: **not surprising**
  - accuracy on standard test set > 95%
  - **accuracy on adversarial test set < 5%**

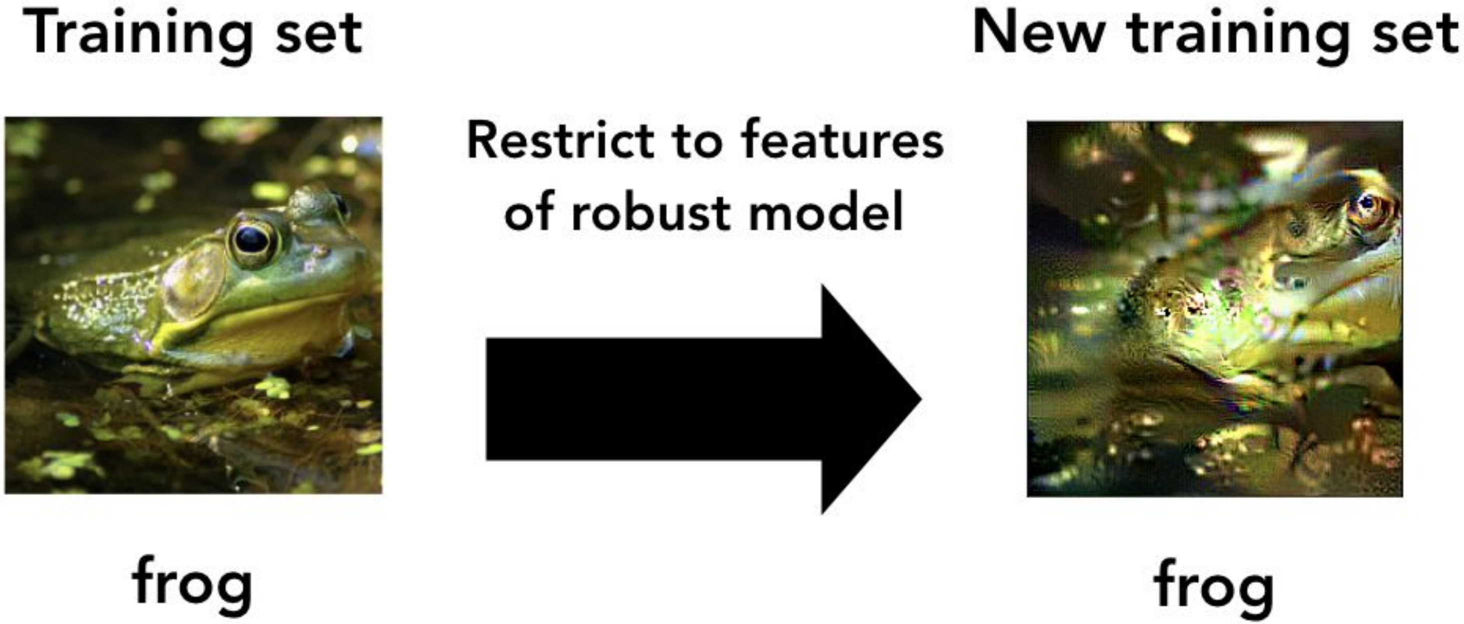
Second baseline

- train a **robust model** on a standard (non-adversarial) dataset
- test it on the adversarial dataset
- results: **not surprising**
  - accuracy on standard test set ~ 90%
  - accuracy on adversarial test set > 80%

First test

- create a “robustified” dataset:
    - samples that primarily contain robust features
    - idea: get rid of the non-robust features
- train a **standard model** on the “**robustified dataset”**
- results: **kind of surprising**
    - accuracy on standard test set > 80%
    - **accuracy on adversarial test set ~ 50% (!)**

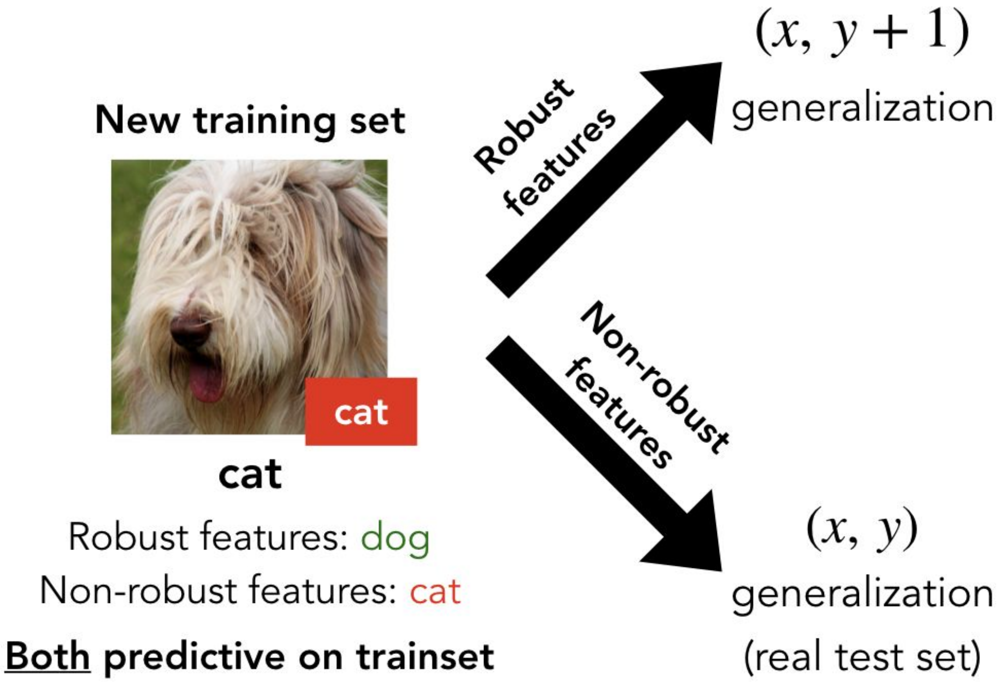
Robustified dataset



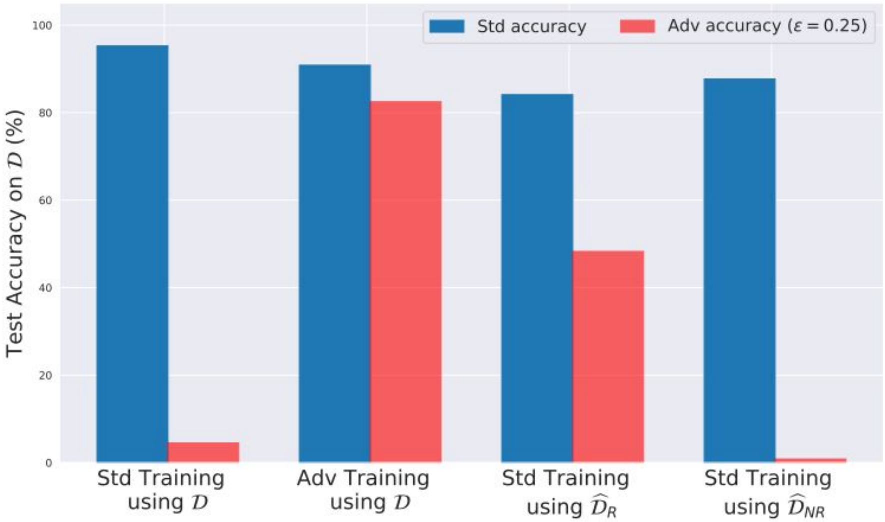
Second test

- create a dataset where the association between the **input** and the **output** is based only on the **non-robust features**
  - appears completely mislabeled to humans!
- train a **standard model** on this dataset
- results: **also kind of surprising**
  - **accuracy on standard test set > 85%**
  - accuracy on adversarial test set < 5%

Non-Robustified Dataset




"airplane"
"ship"
"dog"
"truck"
"frog"
$\mathcal{D}$
$\widehat{\mathcal{D}}_R$
$\widehat{\mathcal{D}}_{NR}$


Std accuracy
Adv accuracy ($\varepsilon = 0.25$)
Test Accuracy on $\mathcal{D}$ (%)
100
80
60
40
20
0
Std Training using $\mathcal{D}$
Adv Training using $\mathcal{D}$
Std Training using $\widehat{\mathcal{D}}_R$
Std Training using $\widehat{\mathcal{D}}_{NR}$

Conclusions

- adversarial examples can be thought as mainly *human phenomena*
- the non-robust features can be highly predictive
  - but not every non-robust features is!
- as long as the models rely on the non-robust features their explanations can be unintelligible to humans

Worth checking out!

- original paper: https://arxiv.org/abs/1905.02175
- MadryLab blog: http://gradientscience.org/
- discussion about the paper: https://distill.pub/2019/advex-bugs-discussion/