Adversarial Examples and Human-ML Alignment

Aleksander Mądry

Based on joint works with:

Logan Engstrom  Andrew Ilyas  Shibani Santurkar
Brandon Tran  Dimitris Tsipras  Alexander Turner

@aleks_madry  gradientscience.org
Deep Networks: Towards Human Vision

Pig: 91%
Dog: 3%
Cat: 2%

ILSVRC top-5 Error on ImageNet
Deep Networks: Towards Human Vision

→ “Meaningful” data representations

Cross-task generalization

Generative models

[Brock et al 2018] + [Isola 2018]
So: Are we on the right path?

(Is all we need “just” scaling up?)

Message for today: Models *deviate* from human perception in *unexpected* ways

→ It is all about features
Deep Networks: Towards Human Vision?

But...

Pig (91%) + 0.005x = Airplane (99%)

Adversarial Examples: Imperceptible changes fool models

Pig: 91%
Dog: 3%
Cat: 2%
...

[Szegedy et al 2013] [Biggio et al 2013]
Deep Networks: Towards Human Vision?
Why do adv. examples exist?

Unifying theme: Adversarial examples are aberrations
A Natural View on Adversarial Examples

“Useless” directions model is unreasonably sensitive to

Useful features that actually help in good classification

Adversary only changes these features to create an adversarial example

Underlying belief: “Better” models would avoid this sensitivity
But: Is this view justified?
Why Are Adv. Perturbations Bad?

But: This is only a “human” perspective
Human Perspective

dog


cat
ML Perspective

Image is meaningless

Classes are meaningless

Only goal: Max (test) accuracy
ML Perspective

dog

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

cat
ML Perspective

tap
toc
ML Perspective

tap

toc
ML Perspective

dog + meaningless perturbation = cat

tap + meaningless perturbation = ?

toc
Are adversarial perturbations indeed meaningless?

[Ilyas Santurkar Tsipras Engstrom Tran M ‘19]
Simple experiment

Training set
(cats vs. dogs)

New training set
(“mislabelled”)

Evaluate on
original test set

Classifier

Adv. ex.
towards the
other class
Simple experiment

Training set (cats vs. dogs)

New training set ("mislabelled")

Evaluate on original test set

How well will this model do?
Simple experiment

Training set (cats vs. dogs)

New training set ("mislabeled")

Evaluate on original test set

Classifier

Result: **Good accuracy** on the **original** test set

(e.g., 78% on CIFAR-10 cats vs. dogs)
What’s going on?

What if adversarial perturbations are not aberrations but features?
The Robust Features Model

Useless directions

Useful features
The Robust Features Model

Useless directions

Robust features
Correlated with label even when perturbed
The Robust Features Model

Useless directions

Robust features
Correlated with label even when perturbed

Non-robust features
Correlated with label, but can be flipped via perturbation

When maximizing (test) accuracy: **All useful** features are good
And: **Non-robust** features are often great!

That’s why our models pick on them (and **become vulnerable to adversarial perturbations**).
The Simple Experiment: A Second Look

All robust features are misleading

But: Non-robust features suffice for good generalization
What now?

A (new) perspective on adversarial robustness

But also: Provides insight into how our models learn
Human vs ML Model Priors

These are **equally valid** classification methods
→ No reason for our models to favor the “human” one
In fact, models...

...can learn from high-frequency components [Yin et al 2019]

...depend too much on texture [Geirhos et al 2019]

...can depend too much on image statistics [Jo & Bengio 2017]

...can be invariant to task-relevant features [Jacobsen et al 2019]

...depend unintuitively on linear directions [Jetley et al 2018]

Adversarial examples are largely a **human** phenomenon

These are **equally valid** classification methods

→ No reason for our models to favor the “human” one
Consequence: Interpretability

Models that use non-robust features cannot be human interpretable.

For instance: Input Saliency Maps

Image  
Gradient  
SmoothGrad

No hope for interpretability without intervention at training time.

Post-hoc interpretations may mask features models depend on.
Consequence: Training Modifications

To get **robust models** we need to explicitly train them to ignore non-robust features

Standard Training: $\min_{\theta} \mathbb{E}_{(x,y) \sim D}[\ell(\theta; x, y)]$

Robust Training: $\min_{\theta} \mathbb{E}_{(x,y) \sim D}[\max_{\delta \in \Delta} \ell(\theta; x + \delta, y)]$

Enforces **additional restrictions (priors)** on what features models can use to make predictions
**Consequence: Robustness Tradeoffs**

**Robust** models can only leverage **robust** features

(Even though non-robust features do help with accuracy)

→ May get a **lower standard accuracy**
   (vide [Tsipras Santurkar Engstrom Turner M ’18])

→ Need **more data** to get a given (robust) accuracy
   (vide [Schmidt Santurkar Tsipras Talwar M ’18])
What if we force models to rely solely on robust features?

[Tsipras Santurkar Engstrom Turner M ’18]
[Engstrom Ilyas Santurkar Tsipras Tran M ’19]
[Santurkar Tsipras Tran Ilyas Engstrom M ’19]
Robustness → Perception Alignment

Prediction: dog

Pixel influence “heatmap” (standard)

Pixel influence “heatmap” (robust)

Models become more (human) perception aligned
Robustness $\rightarrow$ Better Representations

Robust representation distance tends to align better with perceptual distance
Robustness → Better Representations

Direct feature visualization

Seed | Max(different coordinates) | (insect legs) | Most activated | Least activated

Feature manipulation

Add stripes

Interpolation

Robust representations transfer better across tasks
[Salman Ilyas Engstrom Kapoor M ’20]
Robustness \(\rightarrow\) CV Applications

**Generation**

- cliff
- anemone fish
- mashed potato
- coffee pot
- house finch
- armadillo
- chow
- jigsaw
- Norwich terrier
- notebook

**Image Translation**

- clif
- anemone fish
- mashed potato
- coffee pot
- house finch
- armadillo
- chow
- jigsaw
- Norwich terrier
- notebook

**Superresolution**

- cliff
- anemone fish
- mashed potato
- coffee pot
- house finch
- armadillo
- chow
- jigsaw
- Norwich terrier
- notebook

**Inpainting**

- cliff
- anemone fish
- mashed potato
- coffee pot
- house finch
- armadillo
- chow
- jigsaw
- Norwich terrier
- notebook
More Broadly

It is also about choosing what features our models should use.
Problem: Correlations can be weird
Problem: Correlations can be weird

“CNNs were able to detect where an x-ray was acquired [...] and calibrate predictions accordingly.”

[Zech et al. 2018]

“...if an image had a ruler in it, the algorithm was more likely to call a tumor malignant...”

[Esteva et al. 2017]

“Predictive” patterns can be misleading
“Counterfactual” Analysis with Robust Models

Robustness = Framework for controlling what correlations to extract

label: “insect”; prediction: “dog”
Takeaways
Adversarial examples arise from **non-robust features** in the data

→ These features **do** help in generalization (a lot!) and that’s why our models like to rely on them

→ Interpretability needs to be addressed **at training time**

Robustness induces more “human-aligned” representations

→ Enable a broad range of vision applications (in a simple way)

→ Support findings (simple) counterfactuals
But: It is really about how (and what) our models learn

- What is the “right” notion of generalization?
- What features do we want our models to use?
- How much do we value human alignment/interpretability?

Adversarial robustness =
Framework for feature engineering
How can/should robust ML view inform/learn from neuroscience?

Questions?

(See the materials on the website)