Robust Machine Learning: Progress, Challenges, Humans

Dimitris Tsipras

@tsiprasd
gradient-science.org
joint work with

Logan Engstrom
Andrew Ilyas
Aleksandar Makelov
Shibani Santurkar
Ludwig Schmidt

Kunal Talwar
Brandon Tran
Alexander Turner
Adrian Vladu
Aleksander Mądry
Deep Learning can be amazing

Image Classification

Strategy Games

Machine Translation

Realistic Image Generation

Robotic Manipulation
ImageNet: A success story
ImageNet: A success story

ILSVRC top-5 Error on ImageNet

Have we achieved truly super-human performance?
Real-world deployment

Are ML systems ready for the real world?
Core issue: Brittleness

“pig” (91%) +0.005x adversarial noise = “airliner” (99%)

[Biggio et al. 2013] [Dalvi et al. 2004] [Lowd Meek 2005] [Globerson Roweis 2006] [Kolcz Teo 2009] [Barreno et al. 2010] [Biggio et al. 2010] [Biggio et al. 2014] [Srndic Laskov 2013]

Long history in “standard” ML:
Real-world perturbations?

[Athalye Engstrom Ilyas Kwok 2017]
More natural examples?

Training on rotations does not solve the problem

[Fawzi Frossard 2015]
[Engstrom Tran T Schmidt Madry 2017]
Black-box attacks?

Does black-box mean secure?  No.

Query attacks: Directly use input-output queries

Transfer attacks: Just attack a similar model

[Chen et al. 2017]

[_szegedy et al. 2013, Papernot et al. 2016]
Beyond images?

[Carlini Wagner. 2018]: Can arbitrarily confuse a speech recognition system

[Article: Super Bowl 50]
Paragraph: “Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver’s Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.”
Question: “What is the name of the quarterback who was 38 in Super Bowl XXXIII?”
Original Prediction: John Elway
Prediction under adversary: Jeff Dean

[Jia Liang 2017]: Irrelevant sentences confuse reading comprehension models

[Carlini Wagner et al. 2017]: Small changes can bypass malware detection systems
Why should we care?
Security

Already issues with spam and content filtering

[Sharif et al. 2016]

[Evtimov et al. 2018]
Reliability

What we expect from AI

ML models are very brittle

What we (sometimes) get
Human Alignment

How are DL models making predictions?

"pig" (91%)

+0.005x

adversarial noise

= 

"airliner" (99%)

Why is this important to the model?
How do we train robust models?

Our focus:

“pig” + “airliner” = “pig”
How do we find adv. examples?

Standard training

\[ \min_{\theta} \mathbb{E}_{x, y \sim D} [\text{loss}(\theta, x, y)] \]

Adversarial attacks

\[ \max_{\delta \in \Delta} \text{loss}(\theta, x + \delta, y) \]

Allowed perturbations: pixel-wise, rotations, …
How do we train robustly?

**Key observation:** Adversarial examples are not at odds with standard learning

Standard Generalization:

$$\min_{\theta} \mathbb{E}_{x,y \sim D} [loss(\theta, x, y)]$$

Adversarially Robust Generalization:

$$\min_{\theta} \mathbb{E}_{x,y \sim D} [\max_{\delta \in \Delta} loss(\theta, x + \delta, y)]$$

Explicit set of invariances
Towards robust models

\[
\min_\theta \mathbb{E}_{x,y \sim D} \left[ \max_{\delta \in \Delta} \text{loss}(\theta, x + \delta, y) \right]
\]

finding a robust model

(Stochastic) Gradient Descent on \( \theta \)

finding a worst-case perturbation

(Projected) Gradient Descent on \( \delta \)

(How do we get gradients of the max?)

**Theorem (Danskin):** Gradient at maximizer \( \rightarrow \) Gradient of max

\[
\nabla_y \max_x f(x, y) = \nabla_y f(x^*, y) \quad x^* = \arg \max_x f(x, y)
\]
Towards robust models

\[ \min_{\theta} \mathbb{E}_{x,y \sim D} \left[ \max_{\delta \in \Delta} \text{loss}(\theta, x + \delta, y) \right] \]

finding a robust model  
finding a worst-case perturbation

Improve robustness: Train on perturbed inputs

(aka “adversarial training” [Goodfellow et al. 2015])

Actually leads to **robust models** (with some care)
Key ingredient 1: Reliable attacks

We need to train on (almost) **worst-case inputs**

**But:** DNN loss is **non-convex**
Key ingredient 1: Reliable attacks

We need to train on (almost) **worst-case inputs**

**But:** DNN loss is **non-convex**

PGD can still find worst-case inputs **reliably**

**Consistent behavior from random starts**

---

![Graphs showing loss value over iterations](image)
Key ingredient 2: Capacity

Robust models may need to be more expressive.

Weak models can fail to train.

Higher capacity $\Rightarrow$ more robust.
Robust models

Reliable attacks

Sufficient capacity

Result: Adversarial loss decreases steadily
<table>
<thead>
<tr>
<th>Dataset</th>
<th>$\ell_\infty$-norm</th>
<th>$\ell_2$-norm</th>
<th>Rotation+Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST</td>
<td>$\varepsilon = 0.3$</td>
<td>$\varepsilon = 2.5$</td>
<td>$\varepsilon = \pm 3\text{px, }\pm 30^\circ$</td>
</tr>
<tr>
<td></td>
<td>89%</td>
<td>66%</td>
<td>98%</td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>$\varepsilon = 8/255$</td>
<td>$\varepsilon = 0.5$</td>
<td>$\varepsilon = \pm 3\text{px, }\pm 30^\circ$</td>
</tr>
<tr>
<td></td>
<td>53%</td>
<td>70%</td>
<td>82%</td>
</tr>
<tr>
<td>ImageNet</td>
<td>$\varepsilon = 4/255$</td>
<td>$\varepsilon = 1$</td>
<td>$\varepsilon = \pm 3\text{px, }\pm 30^\circ$</td>
</tr>
<tr>
<td></td>
<td>33%</td>
<td>50%</td>
<td>57%</td>
</tr>
</tbody>
</table>
Evaluating robustness can be hard

Many defenses are broken by adaptive attacks

[Carlini Wagner 2016] [Carlini Wagner 2017] [Carlini Wagner 2017] [Athalye et al. 2018] [Uesato et al. 2018]

Try multiple adaptive attacks

Release code and models
Formal robustness verification

Prove robustness on specific examples

Verification

[Tjeng et al. 2019]

MIP solvers

\[
\min_{x'} d(x', x) \\
\text{subject to } \arg\max_i (f_i(x')) \neq \lambda(x) \\
x' \in \mathcal{X}_{\text{valid}}
\]

Accurate but intractable

Certification

[Wong Kolter 2018]

Convex relaxation

Bounds might be too loose

Accurate and efficient verification largely open
Why is robust learning so hard?
Robust generalization is hard

$$\min_{\theta} \mathbb{E}_{x,y \sim D} \left[ \max_{\delta \in \Delta} \text{loss}(\theta, x + \delta, y) \right]$$
Robust generalization is hard

\[
\min_\theta \mathbb{E}_{x,y \sim D} \left[ \max_{\delta \in \Delta} \text{loss}(\theta, x + \delta, y) \right]
\]

\[
\min_\theta \mathbb{E}_{x,y \sim D} [\text{loss}(\theta, x, y)]
\]

Robust Accuracy

>50% overfitting

Doesn’t happen “normally”

Is robust learning fundamentally harder?
Robust generalization is hard

**Theorem:** The sample complexity of robust generalization can be significantly larger than that of “standard” generalization.

**Specifically:** There exists a $d$-dimensional distribution where:

→ A **single sample** is enough to learn a good (standard) classifier

→ **But:** Need at least $\Omega(\sqrt{d})$ samples for a robust classifier
Robust generalization is hard

**Theorem:** The sample complexity of robust generalization can be significantly larger than that of “standard” generalization.

**Empirically:**

![Graphs showing test accuracy vs. training set size for MNIST, CIFAR-10, and SVHN datasets with varying epsilon values.](image-url)
Does robustness improve accuracy?

**Data augmentation:** Train on random transformations of the input

→ Significantly improves test accuracy.

Adversarial training ↔ Augment with the “most helpful” example

Does adversarial training improve **standard accuracy**?
Does robustness improve accuracy?

![Graph showing comparison between standard and robust models with respect to accuracy and training sample size.](image)

Small sample  
Large sample

Why are robust models **less accurate**?
Does robustness improve accuracy?

**Theorem:** There can exist an inherent trade-off between accuracy and robustness (no “free lunch”).

- **Strong correlation** with label
- **Weak correlation** with label

**Standard Training:** use all the features to maximize accuracy

**Adversarial Training:** use only strong features (lower accuracy)
ML vs. “classical” security
Classical security exploits

Attacker use unintended vulnerabilities to manipulate system

Spectre: Side-effects of speculative execution

Heartbleed: Missing out-of-bounds read checks

"Correct" software should be unbreakable
ML security exploits

Robust features
Correlated with label even with adversary

Non-robust features
Correlated with label on average, but can be manipulated

Adversary manipulates input features used for classification
Predictive non-robust features

- Features small in $L_2$-norm

### Accuracy

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>CIFAR10</th>
<th>R. ImageNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>95%</td>
<td>97%</td>
</tr>
<tr>
<td>Non-robust features</td>
<td>44%</td>
<td>64%</td>
</tr>
</tbody>
</table>

Other examples of **unintuitive** features

- **Linear directions** [Jetley et al 2018]
- **High-frequency patterns** [Yin et al 2019]
- **Texture** [Geirhos et al 2019]
Back to adversarial examples

Non-robust features can be quite predictive

We train classifiers to maximize accuracy:
No wonder they utilize non-robust features

Relying on non-robust features directly leads to adversarial vulnerability

Thus: Adversarial examples are not bugs, they are features
Consequences

**Transferability**: Models learn similar non-robust features

Adversarial Transferability \((\text{ResNet-50} \rightarrow X)\)

Test accuracy of X trained on non-robust features from ResNet-50
Consequences

Dataset robustification: Removing non-robust features can improve standard classifiers

Training set

New training set

Restrict to features of robust model

Standard training yields robust classifiers

frog

“Robustified” frog
Humans vs ML Models

Equally valid classification methods

We need to explicitly enforce robustness
Robustness beyond security: Robust models are more human-aligned
**Input Manipulation**

**Key Idea:** Manipulate class scores for **robust models**

Class maximization introduces salient features
Downstream applications

Image Generation
- cliff
- anemone fish
- mashed potato
- coffee pot

Image Translation
- house finch
- armadillo
- chow
- jigsaw
- Norwich terrier
- notebook

Superresolution

Inpainting
Better representations

Direct feature visualization

Seed | Max(different coordinates)
--- | ---
[Images of seed and maximized features]

(insect legs)

Maximized from noise
Most activated
Least activated

Feature manipulation

Add stripes

Interpolation

Better representations
Conclusions
Takeaways

ML models are really **brittle**

Brittleness can arise from **non-robust features**

Robust optimization **can lead to robust models**

Robustness as a tool for **human-aligned** models
Future directions

- More robust models
- Different perturbation sets
- More comprehensive theoretical models
- Further exploration of robust models