A Brief Introduction to Adversarial Examples

Gal Yona

Machine learning and us

Machine learning and us

UNSUPERVISED

פודקסט על Data Science בישראל

Machine learning and us

UNSUPERVISED

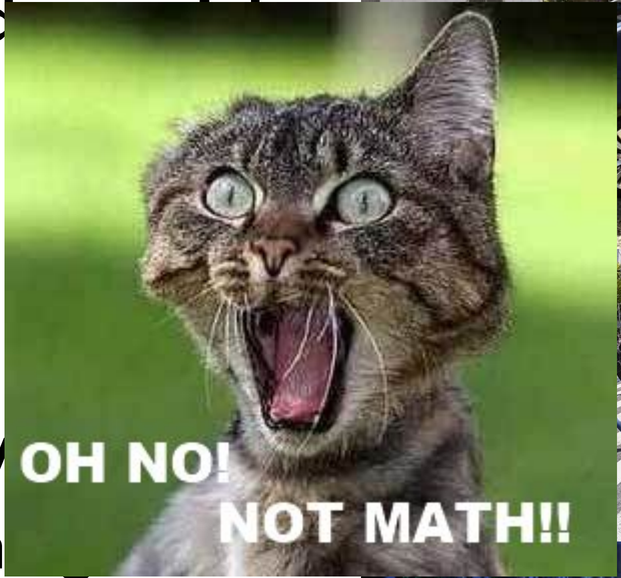
פודקסט על Data Science בישראל

Today’s talk

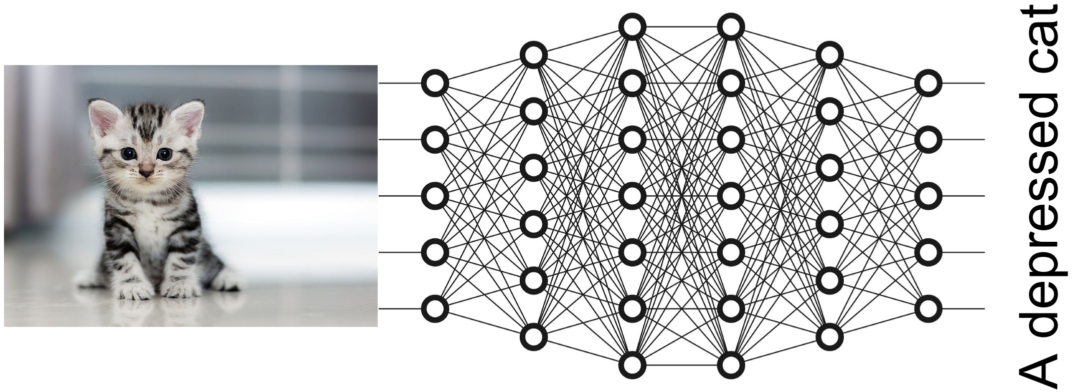
1. **Intro: What are adversarial examples?**
   a. Recent
   b. Intriguing
   c. Timely!
2. **Towards robust ML models**
3. **Adversarial Learning**

Today’s talk

1. **Intro: What are ad**
   examples?
   a. Recent
   b. Intriguing
   c. Timely!
2. **Towards robust M**
3. **Adversarial Learn**

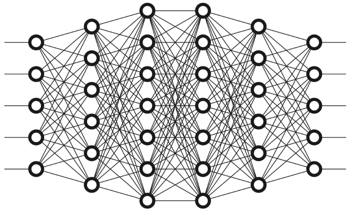


Understanding deep Learning w. visualization

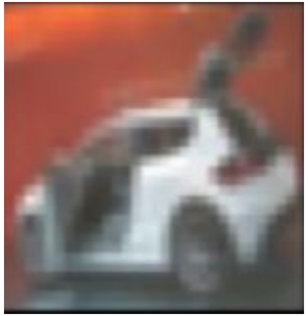


- Deep Learning: still a **black-box**
- Common way of “opening” up the black-box: **visualization**

Understanding deep Learning w. visualization



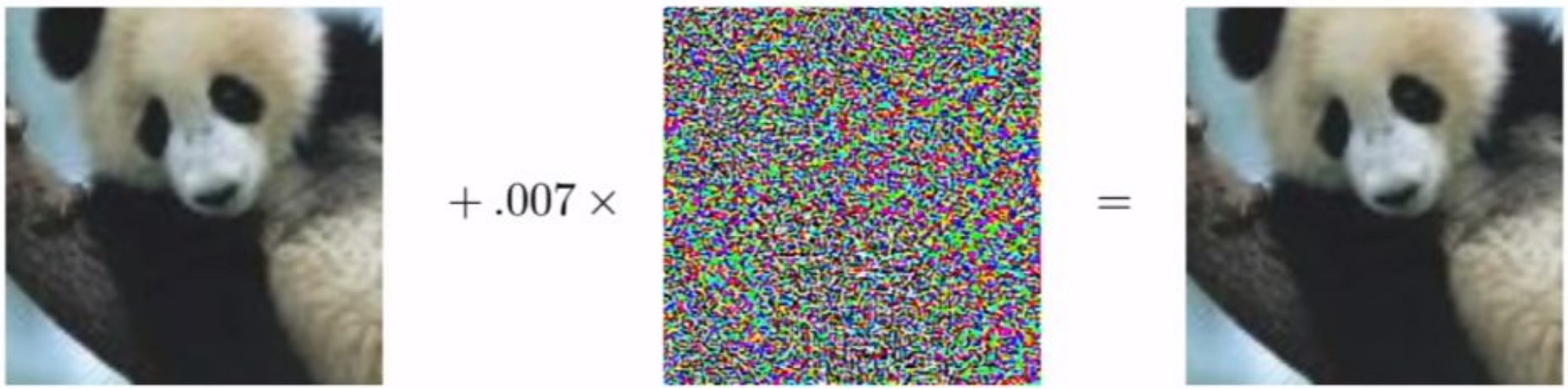
- **Example experiment**
  start with an image of an object, and ask: what changes do I need to make to that image so that the network will think that it is an airplane?



Understanding deep learning w. visualization

Szegedy et al., “**Intriguing properties of neural networks**” (2013)

Adversarial examples



"panda"
57.7% confidence

"gibbon"
99.3% confidence

Adversarial examples

A real gibbon

“panda”
57.7% confidence

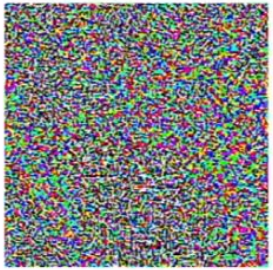
“gibbon”
99.3% confidence

“Imperceptible”

$\cong$

$\Leftrightarrow$

$.007 \times$

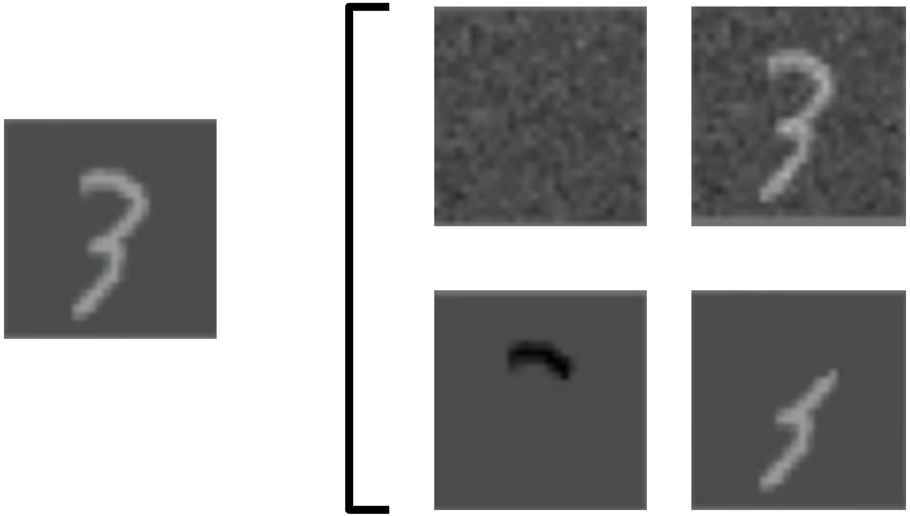


$\underbrace{\hphantom{.007 \times \text{noise image}}}$

small ∞ norm

“Imperceptible”

same l2 norm



“Imperceptible”

Elsayed et al., “**Adversarial Examples that Fool both Computer Vision and Time-Limited Humans**” (2018)

Why? The linear explanation

Goodfellow et al., "**Explaining and Harnessing Adversarial Examples**" (2015)

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\eta}$$

$$||\boldsymbol{\eta}||_\infty < \epsilon$$

$$\boldsymbol{w}^\top \tilde{\boldsymbol{x}} = \boldsymbol{w}^\top \boldsymbol{x} + \boldsymbol{w}^\top \boldsymbol{\eta}$$

Why? The linear explanation

Goodfellow et al., "**Explaining and Harnessing Adversarial Examples**" (2015)

$$\tilde{x} = x + \eta \qquad ||\eta||_\infty < \epsilon$$

$$w^\top \tilde{x} = w^\top x + \underbrace{w^\top \eta}_{\varepsilon mn}$$

Why? The linear explanation

Goodfellow et al., **"Explaining and Harnessing Adversarial Examples"** (2015)

$$\tilde{x} = x + \eta \qquad ||\eta||_\infty < \epsilon$$

$$w^\top \tilde{x} = w^\top x + \underbrace{w^\top \eta}_{\epsilon m n}$$

In high-dimensions, infinitesimal changes "add up":
Even a simple linear model can have adversarial examples if its input has sufficient dimensionality.

Beyond security

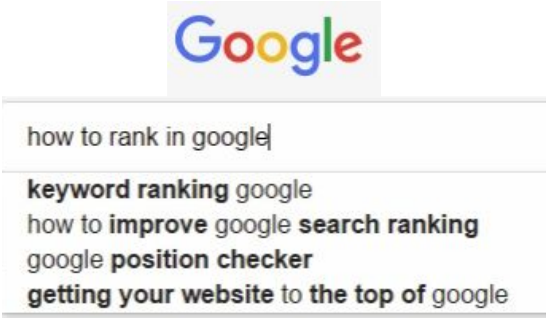
Madry et al., "**Fooling CNNs with Simple Transformations**" (2018)

Research in "Adversarial ML" is mostly around *malicious tampering*; but implications are much broader:

- Robustness against **natural fluctuations** in the underlying distribution
- Handling **feedback loops**: In high-stakes domains, incentives mean people may try to "game" the system.

  Example: Ranking search queries with ML

revolver mousetrap



Bob Sturm., “**Clever Hans, Clever Algorithms**”

What do we do?

(1) Standard classification objective

$$\mathbb{E}_{x,y\sim D}\left[L(f(x),y)\right]$$

Madry et al., "**Towards Deep Learning Models Resistant to Adversarial Attacks**" (2018)

What do we do? Robust classification!

(1) Standard classification objective

$$\mathbb{E}_{x,y\sim D}\left[L(f(x),y)\right]$$

(2) Robust classification objective

$$\mathbb{E}_{x,y\sim D}\left[\max_{x'\in P(x)} L(f(x'),y)\right]$$

Madry et al., "**Towards Deep Learning Models Resistant to Adversarial Attacks**" (2018)

Robust optimization by empirical risk minimization

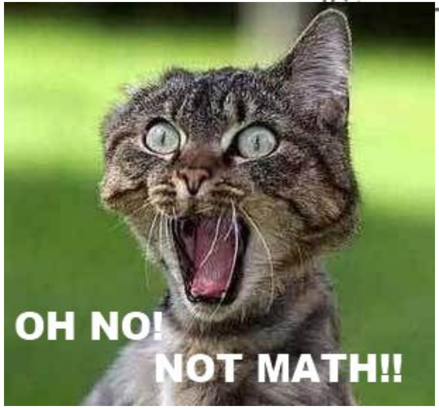
Q: How do we learn a classifier with small loss (2)?

$$\mathbb{E}_{x,y\sim D}\left[\max_{x'\in P(x)} L(f(x'),y)\right]$$

Robust optimization by empirical risk minimization

Q: How do we learn a classifier with small loss (2)?

$$\mathbb{E}_{\sim D}\left[\max_{x' \in P(x)} L(f(x'), y)\right]$$



Robust optimization by empirical risk minimization

Q: How do we learn a classifier with small loss (2)?

$$\mathbb{E}_{x,y\sim D}\left[\max_{x'\in P(x)} L(f(x'),y)\right]$$

A: Use an analogous "robustified" variant of ERM, i.e solve:

$$\min_{\theta} \frac{1}{n}\sum_{i=1}^{n} \max_{x'\in P(x_i)} L(f_\theta(x'),y_i)\ .$$

Robust optimization by empirical risk minimization

Q: How do we learn a classifier with small loss (2)?

$$\mathbb{E}_{x,y\sim D}\left[\max_{x'\in P(x)} L(f(x'),y)\right]$$

A: Use an analogous "robustified" variant of ERM, i.e solve:

Training a robust classifier

Attacking a particular neural network

$$\min_\theta \frac{1}{n}\sum_{i=1}^{n} \max_{x'\in P(x_i)} L(f_\theta(x'),y_i)\ .$$

$$\mathbb{E}_{x,y \sim D}[L(f(x), y)]$$

Standard SGD

Repeat:

Sample $\mathbf{x_1 ... x_m} \sim \mathbf{D}$

Compute gradients of the the loss **L** with parameters $\boldsymbol{\theta}$ w.r.t $\mathbf{x_1 ... x_m}$

Update $\boldsymbol{\theta}$ by taking a step in the direction opposite to the gradient

Training a robust classifier

Attacking a particular neural network

Solving the robustified ERM

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \max_{x' \in P(x_i)} L(f_\theta(x'), y_i) \;.$$

- SGD on the **outer minimization** problem requires **gradients of the inner maximization** problem
- Let **x*** denote the optimal solution of the inner maximization.
- Danskin's Theorem: $\nabla_\theta \phi_{x,y}(\theta) = \nabla_\theta L(f_\theta(x^*), y)$

Solving the robustified ERM

$\phi_{x,y}(\theta) =$

Training a robust classifier

Attacking a particular neural network

This highlights the **duality** between **attacking** a classifier and **training** a robust classifier:

if we have a good attack, we also have a method for finding good gradients of the robust loss.

- S… …he inner maximizatio…
- Let x* denote the o… …mal solution of the inner maximization.
- **Danskin's Theorem**: $\nabla_\theta \phi_{x,y}(\theta) = \nabla_\theta L(f_\theta(x^*), y)$

Adversarial Training

Repeat:

Sample $\mathbf{x}_1 \ldots \mathbf{x}_m \sim \mathbf{D}$

Compute adversarial perturbations $x^*_1 \ldots, x^*_m$

Compute gradients of the the loss **L** with parameters $\boldsymbol{\theta}$ w.r.t $x^*_1 \ldots x^*_m$

Update $\boldsymbol{\theta}$ by taking a step in the direction opposite to the gradient

Adversarial Training

Repeat:

Sample $\mathbf{x}_1 \ldots \mathbf{x}_m \sim \mathbf{D}$

Compute adversarial perturbations $x^*_1 \ldots, x^*_m$

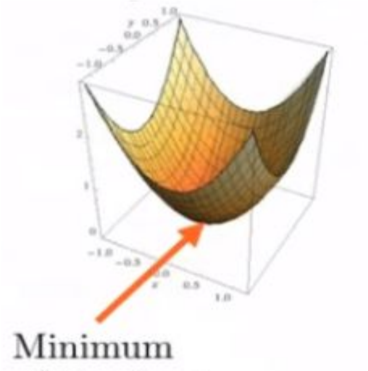
Compute gradients of the the loss **L** with parameters $\boldsymbol{\theta}$ w.r.t $x^*_1 \ldots x^*_m$

Update $\boldsymbol{\theta}$ by taking a step in the direction opposite to the gradient

provably **hard**, even for simple networks…

More generally: Adversarial Learning

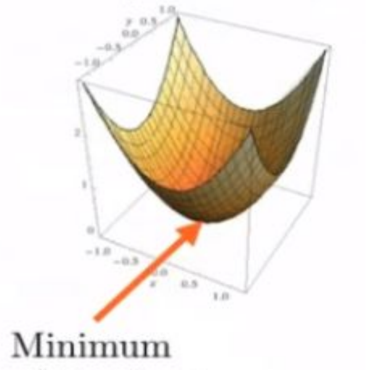
Traditional ML



1 Player, 1 Cost function

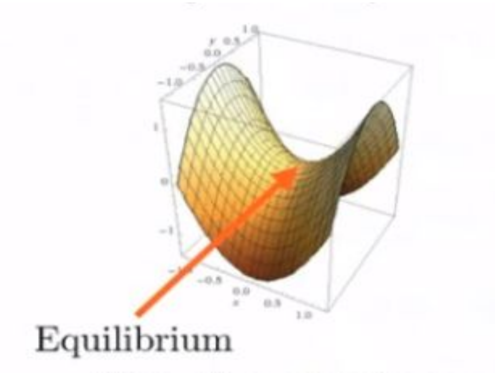
More generally: Adversarial Learning

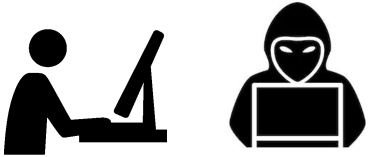
Traditional ML



1 Player, 1 Cost function

Adversarial ML





1+ Players, 1+ Cost functions

Revisiting the "robust ERM"

Adversarial training: a **minimax** problem, with the learning algorithm as the **minimizing** player, and the attacker as the **maximizing** player

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \max_{x' \in P(x_i)} L(f_\theta(x'), y_i) \ .$$

Recap

- Adversarial examples:
    - an intriguing, but also intuitive, phenomena
    - a solution *sketch:* adversarial training
- The same tools (robust classification, adversarial learning) can be useful even when there *isn't* a fear of an actual, real-world, adversary