Adversarial deformations for DNNs

Giovanni S. Alberti

Department of Mathematics, University of Genoa

May 27, 2019

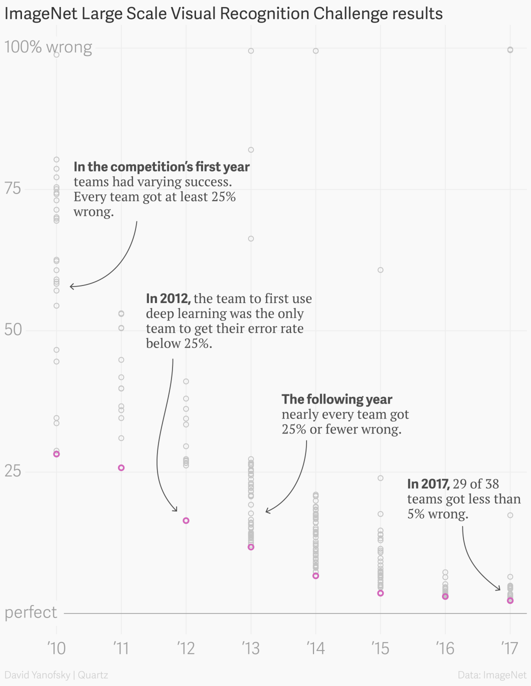
Joint work

Figure: Rima Alaifari
ETH Zürich

Figure: Tandri Gauksson
ETH Zürich

R. Alaifari, G. S. A. and T. Gauksson, *ADef: an Iterative Algorithm to Construct Adversarial Deformations*, ICLR 2019

Contents


ImageNet Large Scale Visual Recognition Challenge results
100% wrong
75
50
25
perfect
'10
'11
'12
'13
'14
'15
'16
'17
In the competition's first year teams had varying success. Every team got at least 25% wrong.
In 2012, the team to first use deep learning was the only team to get their error rate below 25%.
The following year nearly every team got 25% or fewer wrong.
In 2017, 29 of 38 teams got less than 5% wrong.
David Yanofsky | Quartz
Data: ImageNet

Adversarial attacks

- State of the art image classification is achieved by deep neural networks (DNNs).

Adversarial attacks

- State of the art image classification is achieved by deep neural networks (DNNs).
- Weakness: Adversarial examples — slight perturbations to input can lead to misclassification (Szegedy et al 2013).

Adversarial attacks

- State of the art image classification is achieved by deep neural networks (DNNs).
- Weakness: Adversarial examples — slight perturbations to input can lead to misclassification (Szegedy et al 2013).
- Gap between human and machine perception.
- Possible malicious attacks to fool classifiers.

Adversarial attacks

- State of the art image classification is achieved by deep neural networks (DNNs).
- Weakness: Adversarial examples — slight perturbations to input can lead to misclassification (Szegedy et al 2013).
- Gap between human and machine perception.
- Possible malicious attacks to fool classifiers. Defenses???

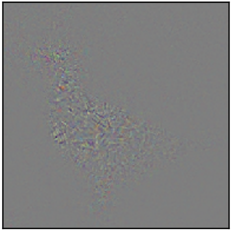
Spot the difference

Spot the difference

Original: ptarmigan

Deformed: partridge

Perturbation



ℓ^∞: 0.027

Image classifiers

- Grayscale square images of $P = w^2$ pixels are vectors in $X := \mathbb{R}^{w \times w} \cong \mathbb{R}^P$.

Image classifiers

- Grayscale square images of $P = w^2$ pixels are vectors in $X := \mathbb{R}^{w \times w} \cong \mathbb{R}^P$.
- A classifier of X into $L \geq 2$ categories is a mapping

$$\mathcal{K} : X \to \{1, \dots, L\}.$$

Image classifiers

- Grayscale square images of $P = w^2$ pixels are vectors in $X := \mathbb{R}^{w \times w} \cong \mathbb{R}^P$.
- A classifier of X into $L \geq 2$ categories is a mapping

$$\mathcal{K} : X \to \{1, \ldots, L\}.$$

- Implemented by

$$\mathcal{K}(x) = \underset{k=1,\ldots,L}{\arg\max}\, F_k(x)$$

for some mapping $F : X \to \mathbb{R}^L$

Image classifiers

- Grayscale square images of $P = w^2$ pixels are vectors in $X := \mathbb{R}^{w \times w} \cong \mathbb{R}^P$.
- A classifier of X into $L \geq 2$ categories is a mapping

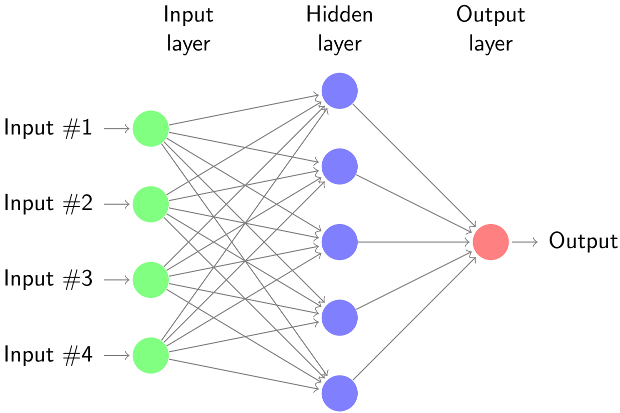
$$\mathcal{K} : X \to \{1, \ldots, L\}.$$

- Implemented by

$$\mathcal{K}(x) = \underset{k=1,\ldots,L}{\arg\max}\, F_k(x)$$

for some mapping $F : X \to \mathbb{R}^L$ (e.g. a neural network).

Structure of DNNs



Neural networks

Definition

A *feedforward neural network* of depth D is a mapping

$$F = F^D \circ F^{D-1} \circ \ldots \circ F^1$$

where

$$F^d : \mathbb{R}^{n_{d-1}} \to \mathbb{R}^{n_d}, \quad x \mapsto \rho(\mathbf{W}^d x + b^d)$$

for some $\mathbf{W}^d \in \mathbb{R}^{n_d \times n_{d-1}}$, $b^d \in \mathbb{R}^{n_d}$ and *activation function* $\rho : \mathbb{R} \to \mathbb{R}$ applied element-wise to $\mathbf{W}^d x + b^d$.

Neural networks

Definition

A *feedforward neural network* of depth D is a mapping

$$F = F^D \circ F^{D-1} \circ \ldots \circ F^1$$

where

$$F^d : \mathbb{R}^{n_{d-1}} \to \mathbb{R}^{n_d}, \quad x \mapsto \rho(\mathbf{W}^d x + b^d)$$

for some $\mathbf{W}^d \in \mathbb{R}^{n_d \times n_{d-1}}$, $b^d \in \mathbb{R}^{n_d}$ and *activation function* $\rho : \mathbb{R} \to \mathbb{R}$ applied element-wise to $\mathbf{W}^d x + b^d$.

- The entries of the matrices $\mathbf{W}^d$ and the vectors b^d are the free parameters and are learned during training.

Neural networks

Definition

A *feedforward neural network* of depth D is a mapping

$$F = F^D \circ F^{D-1} \circ \ldots \circ F^1$$

where

$$F^d : \mathbb{R}^{n_{d-1}} \to \mathbb{R}^{n_d}, \quad x \mapsto \rho(\mathbf{W}^d x + b^d)$$

for some $\mathbf{W}^d \in \mathbb{R}^{n_d \times n_{d-1}}$, $b^d \in \mathbb{R}^{n_d}$ and *activation function* $\rho : \mathbb{R} \to \mathbb{R}$ applied element-wise to $\mathbf{W}^d x + b^d$.

- The entries of the matrices $\mathbf{W}^d$ and the vectors b^d are the free parameters and are learned during training.
- In practice: many layers and $\|\mathbf{W}^d\| > 1 \quad \longrightarrow \quad$ stability unclear

Training

Given labeled data

$$(x_j, l_j) \in X \times \{1, \dots, L\}, \quad j = 1, \dots, m$$

find $F : X \to \mathbb{R}^L$ that captures the distribution.

Training

Given labeled data

$$(x_j, l_j) \in X \times \{1, \dots, L\}, \quad j = 1, \dots, m$$

find $F : X \to \mathbb{R}^L$ that captures the distribution. For example by minimizing the empirical risk

$$\mathcal{R}(F, (x_j, l_j)_{j=1}^m) = \frac{1}{m} \sum_{j=1}^{m} J(F, x_j, l_j)$$

where J is some loss function.

Adversarial perturbations

- Let $x \in X$ be a correctly classified image with label $l = \mathcal{K}(x)$.

Adversarial perturbations

- Let $x \in X$ be a correctly classified image with label $l = \mathcal{K}(x)$.
- Look for another image y close to x that is misclassified, i.e.

Adversarial perturbations

- Let $x \in X$ be a correctly classified image with label $l = \mathcal{K}(x)$.
- Look for another image y close to x that is misclassified, i.e.
  - *adversarial perturbation* $r = y - x$ such that $\|r\|$ is small and

Adversarial perturbations

- Let $x \in X$ be a correctly classified image with label $l = \mathcal{K}(x)$.
- Look for another image y close to x that is misclassified, i.e.
  - *adversarial perturbation* $r = y - x$ such that $\|r\|$ is small and
  - $\mathcal{K}(y) \neq l$.
- Universal perturbations:
  `https://www.youtube.com/watch?v=jhOu5yhe0rc`

Adversarial perturbations

- Let $x \in X$ be a correctly classified image with label $l = \mathcal{K}(x)$.
- Look for another image y close to x that is misclassified, i.e.
  - *adversarial perturbation* $r = y - x$ such that $\|r\|$ is small and
  - $\mathcal{K}(y) \neq l$.
- Universal perturbations:
  `https://www.youtube.com/watch?v=jhOu5yhe0rc`
- Adversarial patch:
  `https://www.youtube.com/watch?v=i1sp4X57TL4`

DeepFool algorithm

S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, 2015

Let $\mathcal{K} = \arg\max F$ be a trained classifier, let $x \in X$ be an image and $l = \mathcal{K}(x)$. The following procedure searches for y with $\mathcal{K}(y) \neq l$:

- Choose a target label $k \neq l$.

DeepFool algorithm

S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, 2015

Let $\mathcal{K} = \arg\max F$ be a trained classifier, let $x \in X$ be an image and $l = \mathcal{K}(x)$. The following procedure searches for y with $\mathcal{K}(y) \neq l$:

- Choose a target label $k \neq l$.
- Set $f := F_k - F_l$: need to have $f(y) > 0$.

DeepFool algorithm

S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, 2015

Let $\mathcal{K} = \arg\max F$ be a trained classifier, let $x \in X$ be an image and $l = \mathcal{K}(x)$. The following procedure searches for y with $\mathcal{K}(y) \neq l$:

- Choose a target label $k \neq l$.
- Set $f := F_k - F_l$: need to have $f(y) > 0$.
- Since $f(x + r) \approx f(x) + \nabla f(x) \cdot r$, define the perturbation

$$r = -\frac{f(x)}{\|\nabla f(x)\|^2} \nabla f(x)$$

and set $\hat{x} = x + r$.

DeepFool algorithm

S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, 2015

Let $\mathcal{K} = \arg\max F$ be a trained classifier, let $x \in X$ be an image and $l = \mathcal{K}(x)$. The following procedure searches for y with $\mathcal{K}(y) \neq l$:

- Choose a target label $k \neq l$.
- Set $f := F_k - F_l$: need to have $f(y) > 0$.
- Since $f(x + r) \approx f(x) + \nabla f(x) \cdot r$, define the perturbation

  $$r = -\frac{f(x)}{\|\nabla f(x)\|^2} \nabla f(x)$$

  and set $\hat{x} = x + r$.
- If $\mathcal{K}(\hat{x}) \neq l$, then we are successful. Otherwise, start at the top with x replaced by $\hat{x}$.

DeepFool algorithm

S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, 2015

Let $\mathcal{K} = \arg\max F$ be a trained classifier, let $x \in X$ be an image and $l = \mathcal{K}(x)$. The following procedure searches for y with $\mathcal{K}(y) \neq l$:

- Choose a target label $k \neq l$.
- Set $f := F_k - F_l$: need to have $f(y) > 0$.
- Since $f(x+r) \approx f(x) + \nabla f(x) \cdot r$, define the perturbation

  $$r = -\frac{f(x)}{\|\nabla f(x)\|^2} \nabla f(x)$$

  and set $\hat{x} = x + r$.
- If $\mathcal{K}(\hat{x}) \neq l$, then we are successful. Otherwise, start at the top with x replaced by $\hat{x}$.

The target label k may be selected at each iteration to minimize $\|r\|$.

Contents

Deformations

- Model images as elements of the space

$$X = L^2([0,1]^2) = \{x \colon [0,1]^2 \to \mathbb{R} : \int_{[0,1]^2} |x(s)|^2 \, ds < +\infty\}$$

Deformations

- Model images as elements of the space

$$X = L^2([0,1]^2) = \{x\colon [0,1]^2 \to \mathbb{R} : \int_{[0,1]^2} |x(s)|^2\, ds < +\infty\}$$

- Given a vector field $\tau : [0,1]^2 \to \mathbb{R}^2$, the deformed image is

$$x_\tau(s) = x(s + \tau(s)).$$

Deformations

- Model images as elements of the space

$$X = L^2([0,1]^2) = \{x\colon [0,1]^2 \to \mathbb{R} : \int_{[0,1]^2} |x(s)|^2\, ds < +\infty\}$$

- Given a vector field $\tau : [0,1]^2 \to \mathbb{R}^2$, the deformed image is

$$x_\tau(s) = x(s + \tau(s)).$$

- In this context, the distance between x and x_τ is not well quantified by a norm of $x - x_\tau$

Deformations

- Model images as elements of the space

$$X = L^2([0,1]^2) = \{x\colon [0,1]^2 \to \mathbb{R} : \int_{[0,1]^2} |x(s)|^2 \, ds < +\infty\}$$

- Given a vector field $\tau : [0,1]^2 \to \mathbb{R}^2$, the deformed image is

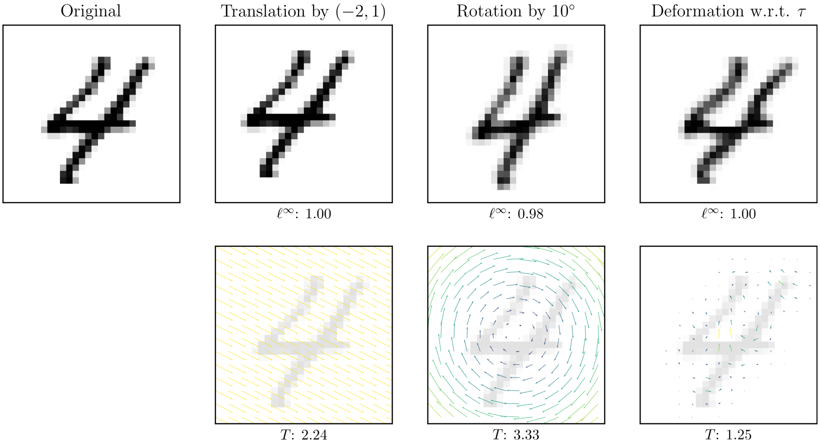
$$x_\tau(s) = x(s + \tau(s)).$$

- In this context, the distance between x and x_τ is not well quantified by a norm of $x - x_\tau$
- Instead, we measure it with a norm on τ:

$$\|\tau\|_T = \|\tau\|_{L^\infty([0,1]^2)} = \sup_{s\in[0,1]^2} \|\tau(s)\|_2$$

Examples of deformations

$$x_\tau(s) = x(s + \tau(s))$$



ADef: constructing adversarial deformations

ADef: constructing adversarial deformations

Let $\mathcal{K} = \arg\max F$ be a classifier, let $x \in X = L^2([0,1]^2)$ be an image and $l = \mathcal{K}(x)$. Goal: Find *small* τ s.t. $l \neq \mathcal{K}(x_\tau)$.

ADef: constructing adversarial deformations

Let $\mathcal{K} = \arg\max F$ be a classifier, let $x \in X = L^2([0,1]^2)$ be an image and $l = \mathcal{K}(x)$. Goal: Find *small* τ s.t. $l \neq \mathcal{K}(x_\tau)$.

- Let $k \neq l$ be a target label

ADef: constructing adversarial deformations

Let $\mathcal{K} = \arg\max F$ be a classifier, let $x \in X = L^2([0,1]^2)$ be an image and $l = \mathcal{K}(x)$. Goal: Find *small* τ s.t. $l \neq \mathcal{K}(x_\tau)$.

- Let $k \neq l$ be a target label
- Set $g \colon \tau \mapsto F_k(x_\tau) - F_l(x_\tau)$, search for τ s.t. $g(\tau) > 0$

ADef: constructing adversarial deformations

Let $\mathcal{K} = \arg\max F$ be a classifier, let $x \in X = L^2([0,1]^2)$ be an image and $l = \mathcal{K}(x)$. Goal: Find *small* τ s.t. $l \neq \mathcal{K}(x_\tau)$.

- Let $k \neq l$ be a target label
- Set $g\colon \tau \mapsto F_k(x_\tau) - F_l(x_\tau)$, search for τ s.t. $g(\tau) > 0$
- By linear approximation

$$g(\tau) \approx g(0) + (D_0 g)\tau,$$

with (Fréchet) derivative

$$(D_0 g)\tau = \int_{[0,1]^2} \alpha(s) \cdot \tau(s)\, ds, \quad \alpha(s) = (D_x F_k - D_x F_l)(s) \nabla x(s).$$

ADef: constructing adversarial deformations

Let $\mathcal{K} = \arg\max F$ be a classifier, let $x \in X = L^2([0,1]^2)$ be an image and $l = \mathcal{K}(x)$. Goal: Find *small* τ s.t. $l \neq \mathcal{K}(x_\tau)$.

- Let $k \neq l$ be a target label
- Set $g \colon \tau \mapsto F_k(x_\tau) - F_l(x_\tau)$, search for τ s.t. $g(\tau) > 0$
- By linear approximation

  $$g(\tau) \approx g(0) + (D_0 g)\tau,$$

  with (Fréchet) derivative

  $$(D_0 g)\tau = \int_{[0,1]^2} \alpha(s) \cdot \tau(s)\, ds, \quad \alpha(s) = (D_x F_k - D_x F_l)(s) \nabla x(s).$$

- Solve $(D_0 g)\tau = -g(0)$ in least-squares sense:

  $$\tau(s) = -\frac{g(0)}{\|\alpha\|^2_{L^2([0,1])}}\, \alpha(s)$$

ADef: constructing adversarial deformations

Let $\mathcal{K} = \arg\max F$ be a classifier, let $x \in X = L^2([0,1]^2)$ be an image and $l = \mathcal{K}(x)$. Goal: Find *small* τ s.t. $l \neq \mathcal{K}(x_\tau)$.

- Let $k \neq l$ be a target label
- Set $g \colon \tau \mapsto F_k(x_\tau) - F_l(x_\tau)$, search for τ s.t. $g(\tau) > 0$
- By linear approximation

  $$g(\tau) \approx g(0) + (D_0 g)\tau,$$

  with (Fréchet) derivative

  $$(D_0 g)\tau = \int_{[0,1]^2} \alpha(s)\cdot\tau(s)\,ds, \quad \alpha(s) = (D_x F_k - D_x F_l)(s)\nabla x(s).$$

- Solve $(D_0 g)\tau = -g(0)$ in least-squares sense:

  $$\tau(s) = -\frac{g(0)}{\|\alpha\|^2_{L^2([0,1])}}\,\alpha(s)$$

- Repeat until $\mathcal{K}(x^{(n)}) \neq l$ for $x^{(n)}(s) = x^{(n-1)}(s + \tau^{(n)}(s))$.

Contents

MNIST database

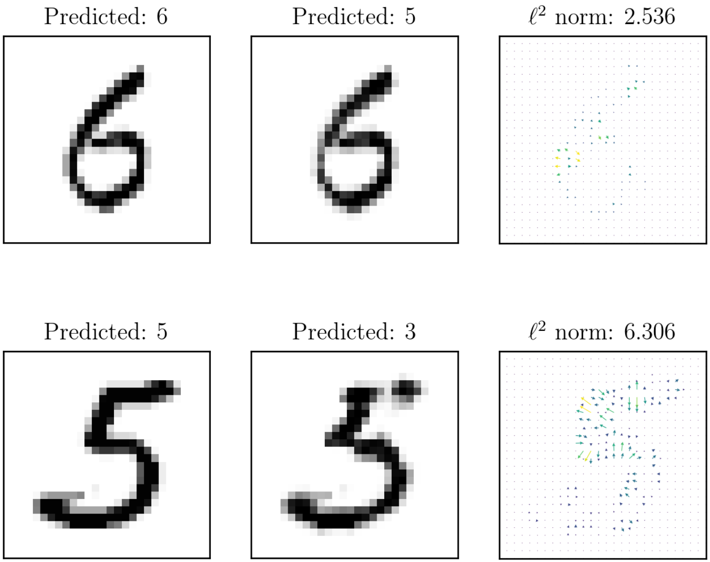
- 60 000 training images
- 10 000 test images
- 28×28 pixels

ILSVRC database (ImageNet)

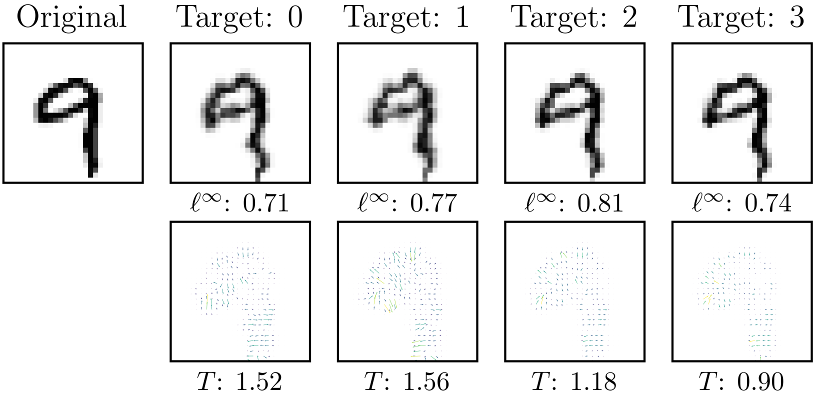
Figure: © Andrej Karpathy

- 1 000 image categories (classes)
- 1.2 million training images, 50 000 validation images
- 100 000 test images
- 256 $\times$ 256 pixels

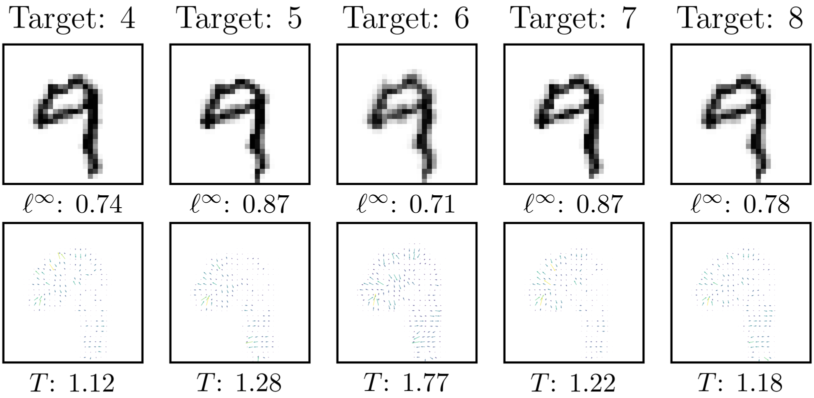
Example: MNIST with CNN



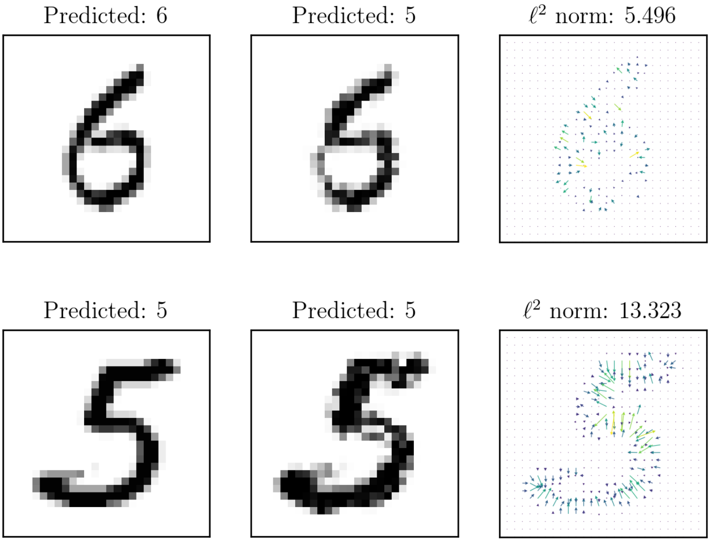
Example: Targeted attack on MNIST with CNN



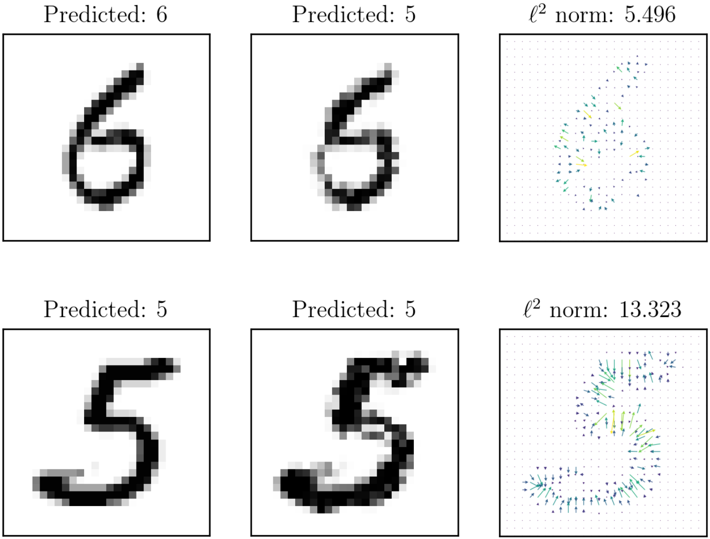
Example cont'd



Example: MNIST with scattering network



Example: MNIST with scattering network



Here: 2 predefined(!) layers + 1 fully-connected layer + SVM.

Results for ADef

Results for ADef

Three different networks:

- MNIST: convolutional neural network
- ImageNet (ILSVRC2012): Inception-v3
- ImageNet (ILSVRC2012): ResNet-101

Results for ADef

Three different networks:

- MNIST: convolutional neural network
- ImageNet (ILSVRC2012): Inception-v3
- ImageNet (ILSVRC2012): ResNet-101

Model	Accuracy	ADef success	Avg. # iterations
MNIST-CNN	99.41%	99.90%	9.779
Inception-v3	77.56%	98.94%	4.050
ResNet-101	76.97%	99.78%	4.176

Deformations are small

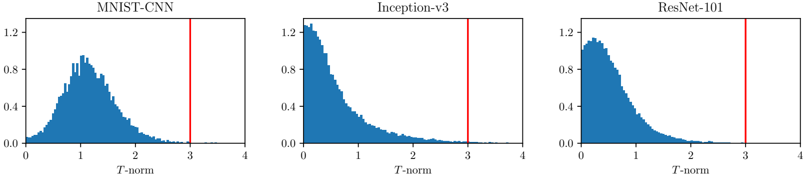
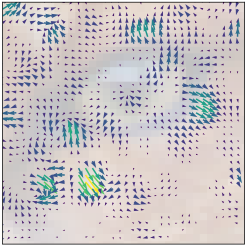


Figure: The (normalized) distribution of $\|\tau\|_T$ from the experiment. Deformations that fall to the left of the vertical line at $\varepsilon = 3$ are considered successful.

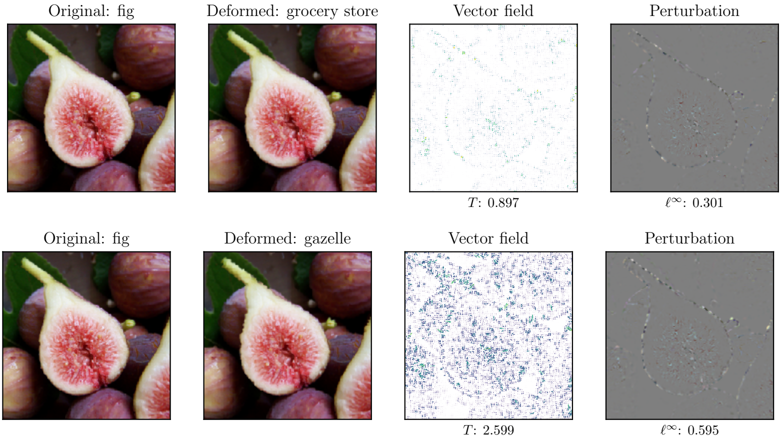
Example: ImageNet

Red fox

Shopping cart



Untargeted vs. targeted attack



Attack on adversarially trained networks

Model	Adv. training	Accuracy	PGD success	ADef success
MNIST-A	PGD	98.36%	5.81%	6.67%
	ADef	98.95%	100.00%	54.16%
MNIST-B	PGD	98.74%	5.84%	20.35%
	ADef	98.79%	100.00%	45.07%

Conclusions

- ADef: DNN can be fooled by adversarial deformations
- Defenses using deformations?
- Relevance for inverse problems

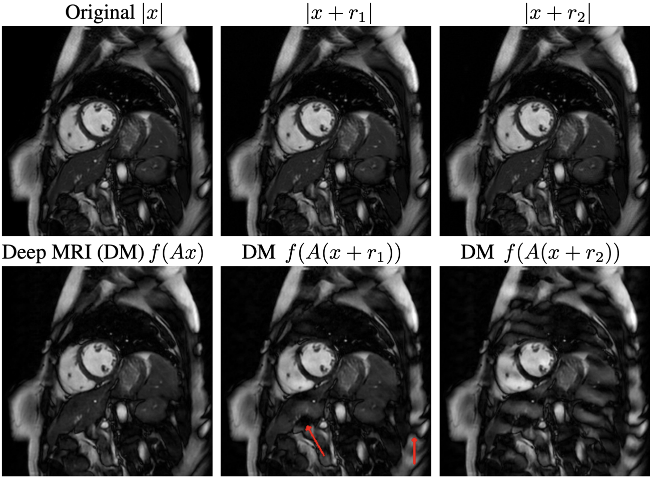


Figure: V. Antun, F. Renna, C. Poon, B. Adcock, A.C. Hansen, 2019

Summer School on Applied Harmonic Analysis and Machine Learning

Genoa, September 9-13, 2019

Home Outline Schedule Info Registration

[~] *Three minicourses on Signal Analysis and Big Data*

School speakers:

Rima Alaifari (ETH Zurich)
Gabriel Peyré (École Normale Supérieure, Paris)
José Luis Romero (University of Vienna)

Workshop speakers:

Massimo Fornasier (Technical University of Munich)
Anders Hansen (University of Cambridge)

Organizers:

Giovanni S. Alberti
Filippo De Mari
Ernesto De Vito
Lorenzo Rosasco
Matteo Santacesaria
Silvia Villa

Sponsors:

DIMA

SLIPGURU