How intelligent is artificial intelligence ?

–

On the surprising and mysterious secrets of deep learning

Anders C. Hansen (Cambridge and UiO)
Vegard Antun (UiO)

Joint work with:

B. Adcock (SFU) A. Bastounis (Cambridge)
M. Colbrook (Cambridge) N. Gottschling (Cambridge)
C. Poon (Bath) F. Renna (Porto)
L. Thesing (Cambridge) V. Vlacic (ETH)

Oslo, May 2019

Main goal: Secure and Safe AI

Main issues:

- AI techniques will replace humans in problem solving.
- AI techniques will replace established algorithms in the sciences.

AI replacing humans

- Self-driving vehicles
- Automated diagnosis in medicine
- Automated decision processes
- Automated weapon systems
- Any security system based on face or voice recognition

AI replacing algorithms

- Medical imaging (MRI, CT, etc)
- Microscopy
- Imaging problems in general
- Radar
- Sonar
- Inverse problems in general
- PDEs

Layout of the talks

Day I : AI techniques will replace humans in problem solving: Classification.

Day II : AI techniques will replace established algorithms in the sciences: Inverse problems and image reconstruction.

Testing AI, Philosophical questions.

Day III : Technical issues.

Why is suddenly AI such a big deal?

The Pioneers

Forbes Billionaires Innovation Leadership Money Consumer Industry

Turing Award And $1 Million Given To 3 AI Pioneers

Nicole Martin Contributor

AI & Big Data

I write about technology, data and privacy.

Winners of Turning Award NEW YORK TIMES

The Association for Computing Machinery (ACM) awarded Yoshua Bengio, Geoffrey Hinton and Yann LeCun with what many consider the "Nobel Prize of computing," for the innovations they've made in AI.

Cookies on Forbes

Citation from the Turing Award jury

Select Technical Accomplishments

The technical achievements of this year's Turing Laureates, which have led to significant breakthroughs in AI technologies include, but are not limited to, the following:

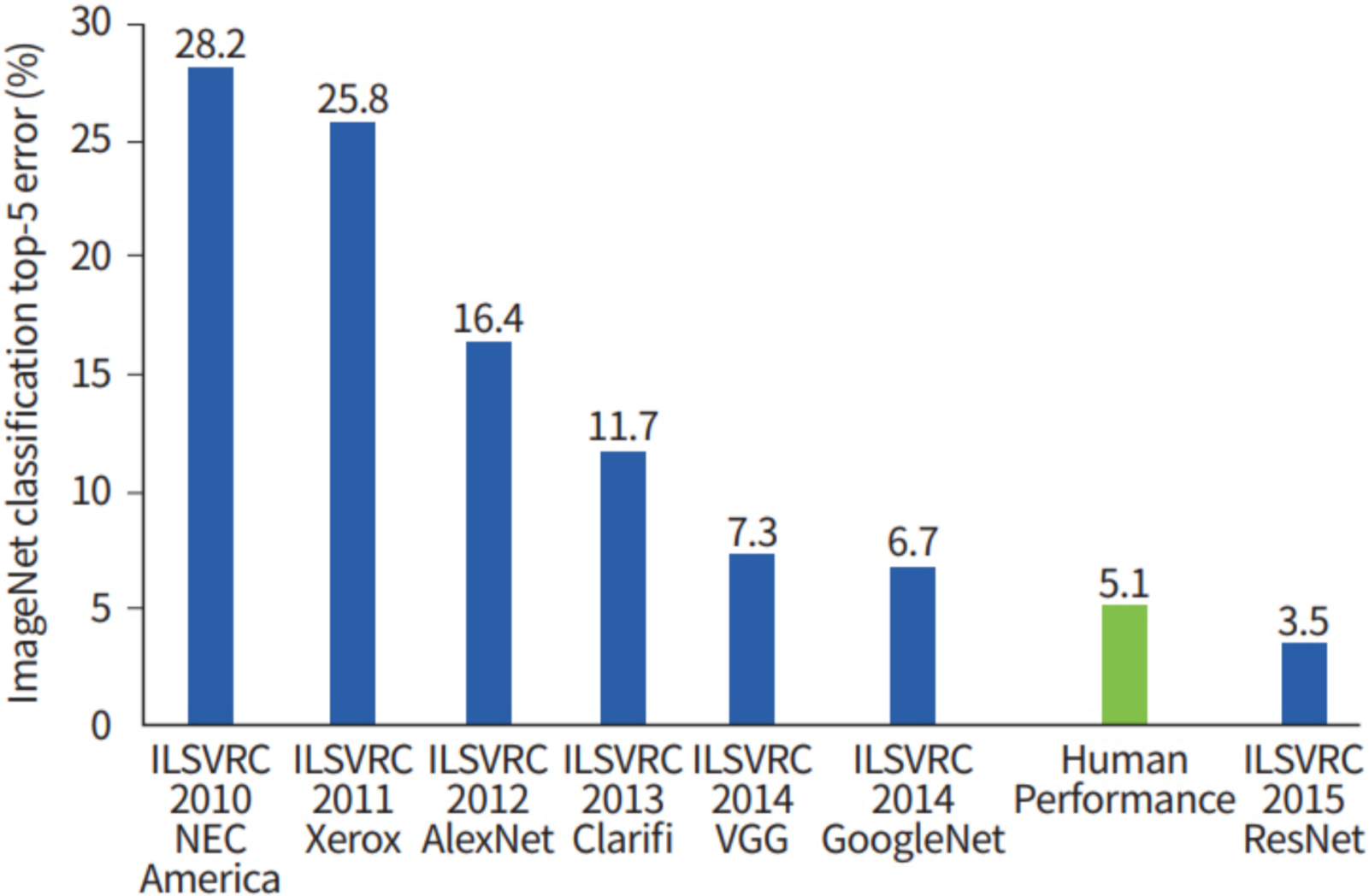
Geoffrey Hinton

Backpropagation: In a 1986 paper, "Learning Internal Representations by Error Propagation," co-authored with David Rumelhart and Ronald Williams, Hinton demonstrated that the backpropagation algorithm allowed neural nets to discover their own internal representations of data, making it possible to use neural nets to solve problems that had previously been thought to be beyond their reach. The backpropagation algorithm is standard in most neural networks today.

Boltzmann Machines: In 1983, with Terrence Sejnowski, Hinton invented Boltzmann Machines, one of the first neural networks capable of learning internal representations in neurons that were not part of the input or output.

Improvements to convolutional neural networks: In 2012, with his students, Alex Krizhevsky and Ilya Sutskever, Hinton improved convolutional neural networks using rectified linear neurons and dropout regularization. In the prominent ImageNet competition, Hinton and his students almost halved the error rate for object recognition and reshaped the computer vision field.

Before and after 2012 - The ImageNet competition



Before and after 2012 - The ImageNet competition

Top 5 ILSVRC 2012 Results		
1st	Error: 16.4%	Deep Learning
2nd	Error: 26.1%	Other approach
3rd	Error: 26.9%	Other approach
4th	Error: 29.5%	Other approach
5th	Error: 34.4%	Other approach
Top 5 ILSVRC 2017 Results		
1st	Error: 2.3%	Deep Learning
2nd	Error: 2.5%	Deep Learning
3rd	Error: 2.7%	Deep Learning
4th	Error: 3.0%	Deep Learning
5th	Error: 3.2%	Deep Learning

Table : Results from ImageNet Large Scale Visual Recognition Competition (ILSVRC).

Strong confidence in deep learning

The New Yorker quotes Geoffrey Hinton (April 2017):

"They should stop training radiologists now."

FDA NEWS RELEASE

FDA permits marketing of artificial intelligence-based device to detect certain diabetes-related eye problems

For Immediate Release: April 11, 2018

The U.S. Food and Drug Administration today permitted marketing of the first medical device to use artificial intelligence to detect greater than a mild level of the eye disease diabetic retinopathy in adults who have diabetes.

Diabetic retinopathy occurs when high levels of blood sugar lead to damage in the blood vessels of the retina, the light-sensitive tissue in the back of the eye. Diabetic retinopathy is the most common cause of vision loss among the more than 30 million Americans living with diabetes and the leading cause of vision impairment and blindness among working-age adults.

nature
International journal of science

Letter | Published: 21 March 2018

Image reconstruction by domain-transform manifold learning

Bo Zhu, Jeremiah Z. Liu, Stephen F. Cauley, Bruce R. Rosen & Matthew S. Rosen

Nature **555**, 487–492 (22 March 2018) | Download Citation

Abstract

Image reconstruction is essential for imaging applications across the physical and life sciences, including optical and radar systems, magnetic resonance imaging, X-ray computed tomography, positron

AI replaces algorithms in medical imaging

nature methods

Research Highlights | Published: 27 April 2018

Imaging

AI transforms image reconstruction

Rita Strack

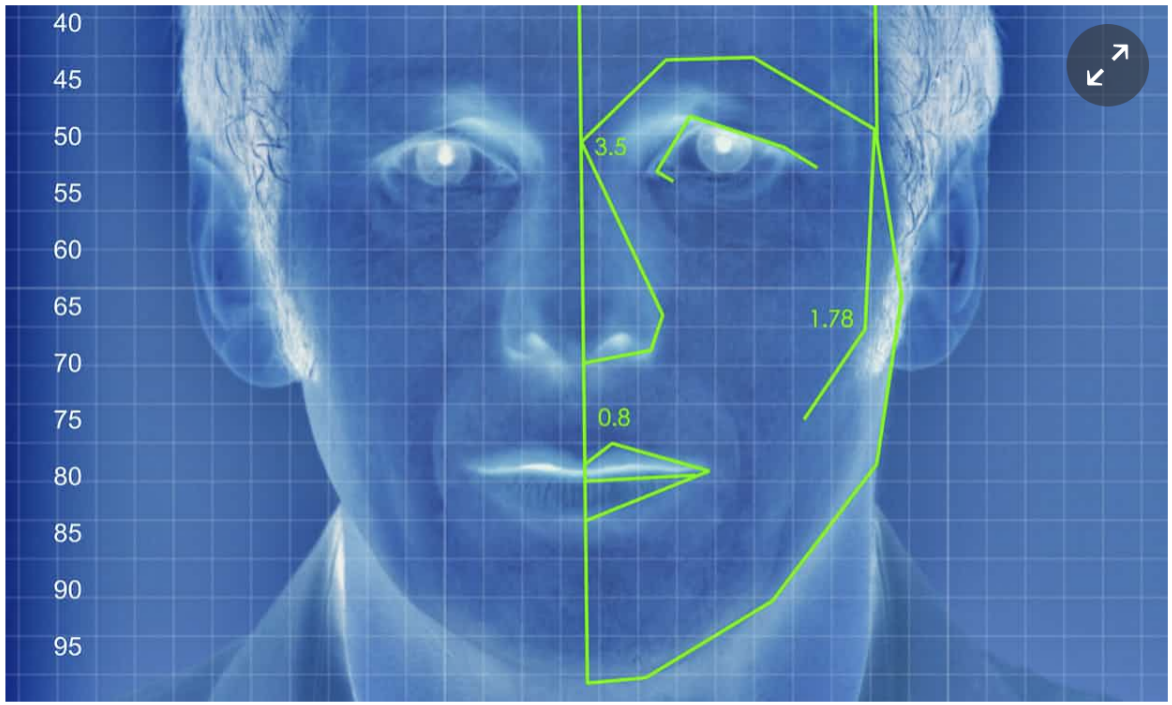
Nature Methods **15**, 309 (2018) | Download Citation

A deep-learning-based approach improves the speed, accuracy, and robustness of biomedical image reconstruction.

... and AI seems to be used for other things as well

New AI can guess whether you're gay or straight from a photograph

An algorithm deduced the sexuality of people on a dating site with up to 91% accuracy, raising tricky ethical questions



An illustrated depiction of facial analysis technology similar to that used in the experiment. Illustration: Alamy

Artificial intelligence can accurately guess whether people are gay or straight based on photos of their faces, according to new research that suggests machines can have significantly better “gaydar” than humans.

The study from Stanford University - which found that a computer algorithm

Given a classification function $f : \mathbb{R}^d \to \{0, 1\}$, find an approximation $\tilde{f} : \mathbb{R}^d \to \{0, 1\}$ to f.

Construct $\tilde{f}$ based on a training set $\mathcal{T} = \{x^1, \ldots, x^r\} \subset \mathbb{R}^d$ for which we know $f(x^j)$ for $j = 1, \ldots, r$.

Test $\tilde{f}$ on a classification (or a test) set $\mathcal{C} = \{y^1, \ldots, y^s\}$. Success is measure by

$$\frac{|\{y^j \in \mathcal{C} \mid f(y^j) = \tilde{f}(y^j)\}|}{s}$$

Neural networks

Let $\mathcal{NN}_{\mathbf{N},L,d}$, with $\mathbf{N} = (N_L, N_{L-1}, \ldots, N_1, N_0 = d)$ denote the set of all L-layer neural networks. That is, all mappings $\phi : \mathbb{R}^d \to \mathbb{R}^{N_L}$ of the form

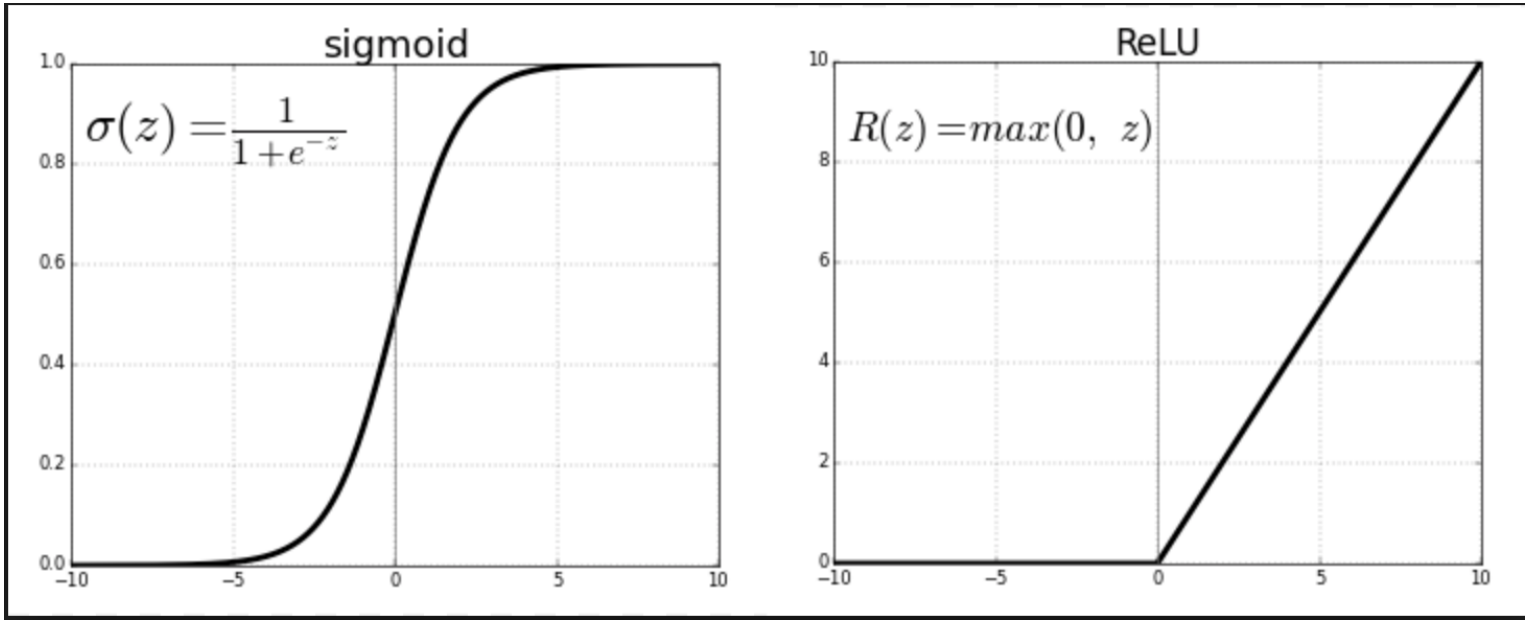
$$\phi(x) = W_L(\rho(W_{L-1}(\rho(\ldots \rho(W_1(x)))))), \quad x \in \mathbb{R}^d.$$

$$W_j y = A_j y - b_j, \qquad A_j \in \mathbb{R}^{N_j \times N_{j-1}}, \quad b_j \in \mathbb{R}^{N_j}$$

$$\rho : \mathbb{R} \to \mathbb{R}$$

is some non-linear function that acts pointwise on a vector.

Choices of ρ



Approximation qualities of neural nets

The universal approximation theorem:

Theorem 1 (Pinkus, Acta Numerica 1999)

Let $\rho \in C(\mathbb{R})$. Then the set of neural networks is dense in $C(\mathbb{R}^d)$ in the topology of uniform convergence on compact sets, if and only if ρ is not a polynomial.

The interpolation theorem:

Theorem 2 (Pinkus, Acta Numerica 1999)

Let $\rho \in C(\mathbb{R})$ and assume that ρ is not a polynomial. For any k distinct points $\{x_j\}_{j=1}^k \subset \mathbb{R}^d$ and associated data $\{\alpha_j\}_{j=1}^k \subset \mathbb{R}$. Then there exists a neural network ϕ such that

$$\phi(x_j) = \alpha_j, \qquad j = 1, \ldots, k.$$

Training neural nets

Given a classification function $f : \mathbb{R}^d \to \{0, 1\}$, a training set $\mathcal{T} = \{x^1, \ldots, x^r\} \subset \mathbb{R}^d$, a classification set $\mathcal{C} = \{y^1, \ldots, y^s\}$, and a cost function $C : \mathbb{R}^r \times \mathbb{R}^r \to \mathbb{R}_+$, compute

$$\phi \in \underset{\tilde{\phi} \in \mathcal{NN}_{\mathbf{N},L,d}}{\operatorname{argmin}} \; C(v, w),$$

with

$$v = \{\tilde{\phi}(x^j)\}_{j=1}^r, \quad w = \{f(x^j)\}_{j=1}^r.$$

Typical choice is $C(v, w) = \|v - w\|_p^p$.

What could go wrong?

Deep learning is demonstrating super human behaviour.

There is a mathematical theory suggesting that neural nets have all the approximation qualities that are needed.

What could possibly go wrong?

-

AI replacing humans

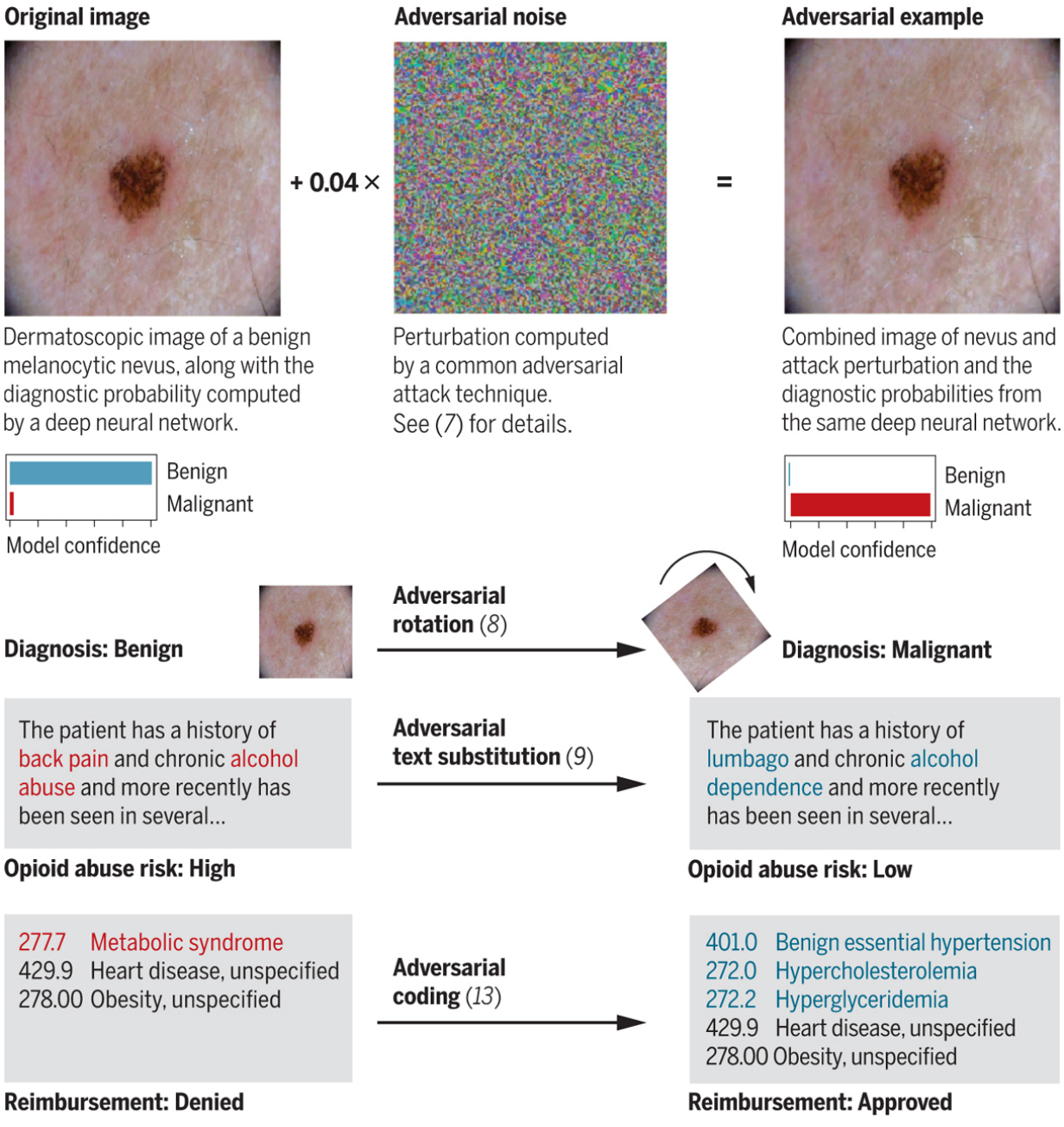
Adversarial attacks on medical machine learning

Samuel G. Finlayson[1], John D. Bowers[2], Joichi Ito[3], Jonathan L. Zittrain[2], Andrew L. Beam[4], Isaac S. Kohane[1]
+ See all authors and affiliations

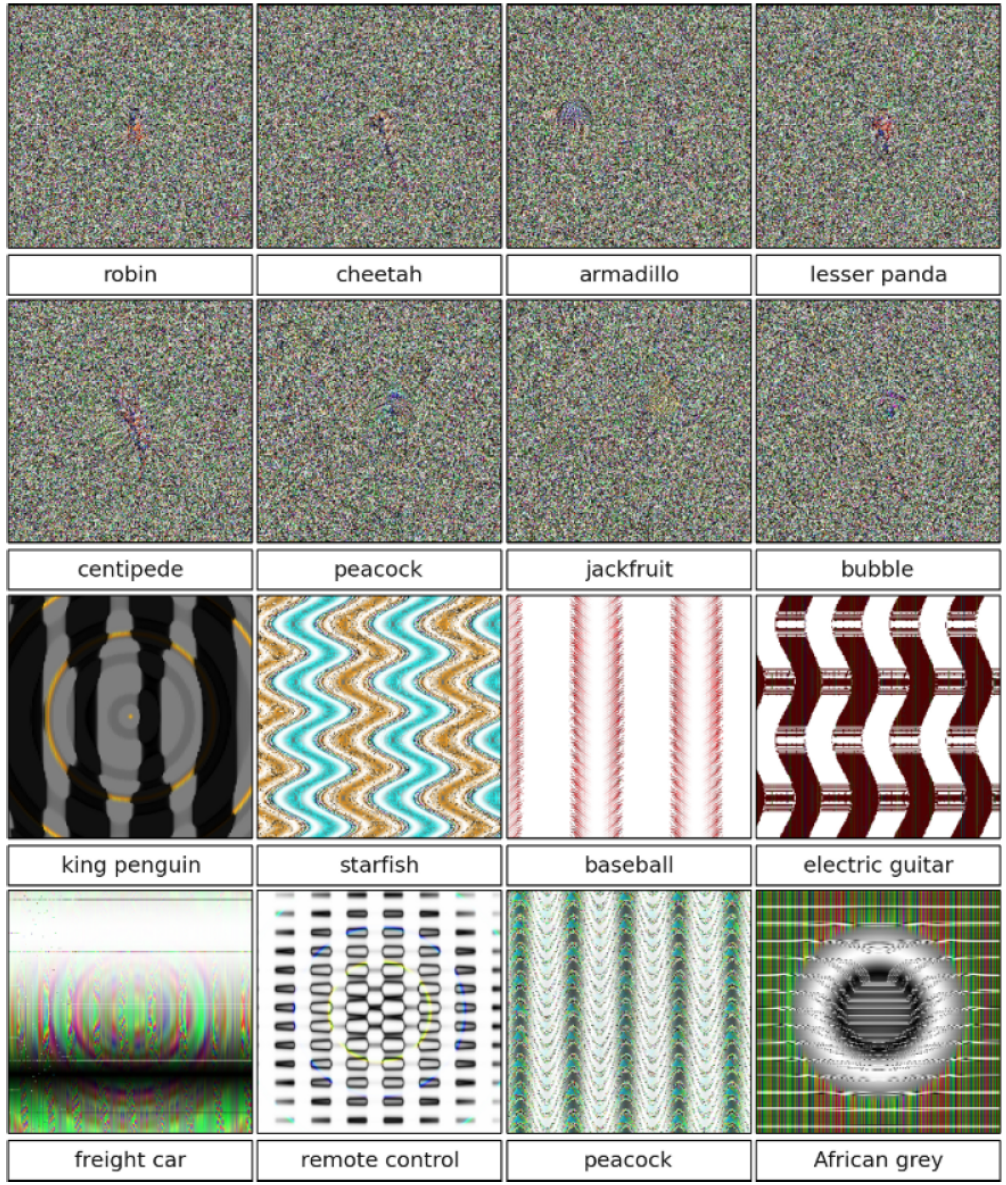
Science 22 Mar 2019:
Vol. 363, Issue 6433, pp. 1287-1289
DOI: 10.1126/science.aaw4399

With public and academic attention increasingly focused on the new role of machine learning in the health information economy, an unusual and no-longer-esoteric category of vulnerabilities in machine-learning systems could prove important. These vulnerabilities allow a small, carefully designed change in how inputs are presented to a system to completely alter its output, causing it to confidently arrive at manifestly wrong conclusions. These advanced techniques to subvert otherwise-reliable machine-learning systems—so-called adversarial attacks—have, to date, been of interest primarily to computer science researchers (*1*). However, the landscape of often-competing interests within health care, and billions of dollars at stake in systems' outputs, implies considerable problems. We outline motivations that various players in the health care system may have to use adversarial attacks and begin a discussion of what to do about them. Far from discouraging continued innovation with medical machine learning, we call for active engagement of medical, technical, legal, and ethical experts in pursuit of efficient, broadly available, and effective health care that machine learning will enable.

What could go wrong?



What has deep learning actually learned?



"Deep neural networks are easily fooled: High confidence predictions for unrecognizable images", A. Nguyen, J. Yosinski, and J. Clune. *2015 IEEE Conference on Computer Vision and Pattern Recognition*.

Deep Fool

Deep Fool was established at EPFL in order to study the stability of neural networks.

Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard

The Robustness of Deep Networks

A geometrical perspective

Deep Fool in practice

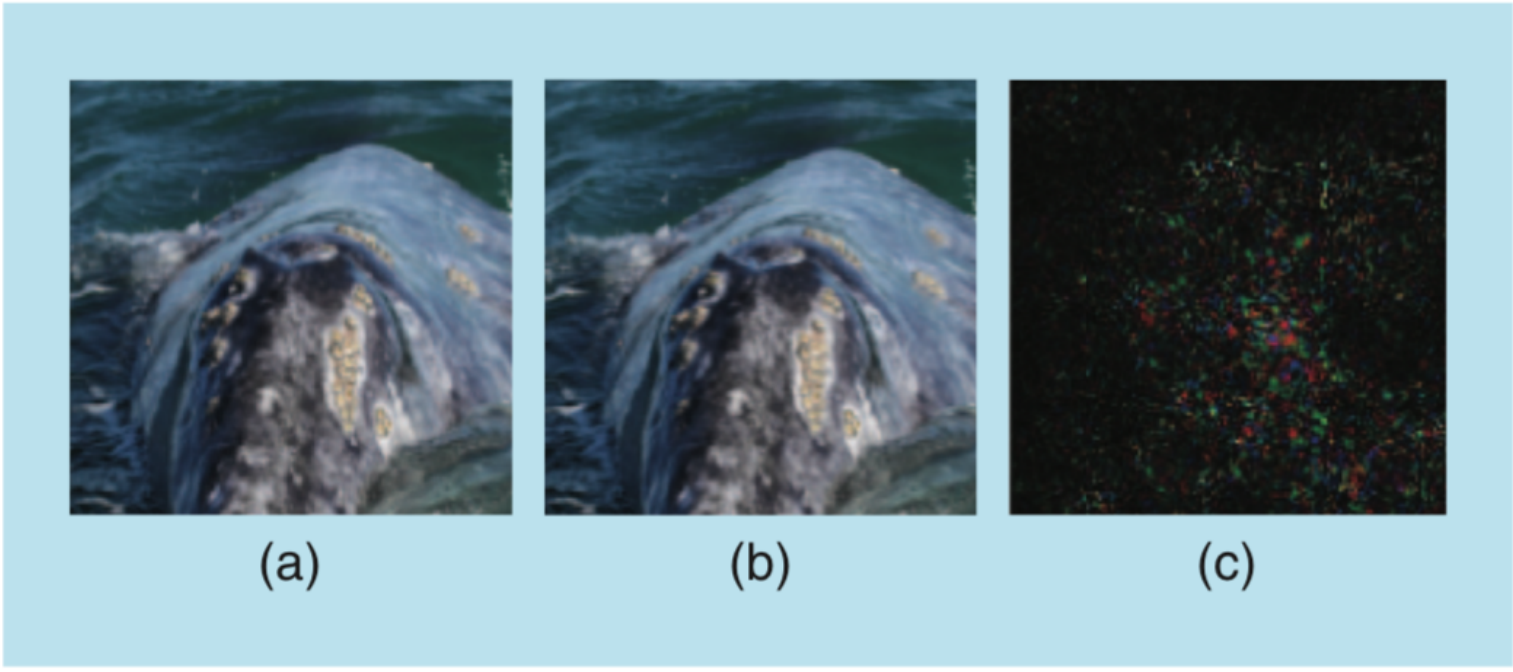


FIGURE 1. An example of an adversarial perturbations in state-of-the-art neural networks. (a) The original image that is classified as a “whale,” (b) the perturbed image classified as a “turtle,” and (c) the corresponding adversarial perturbation that has been added to the original image to fool a state-of-the-art image classifier [5].

Deep Fool: Universal perturbations

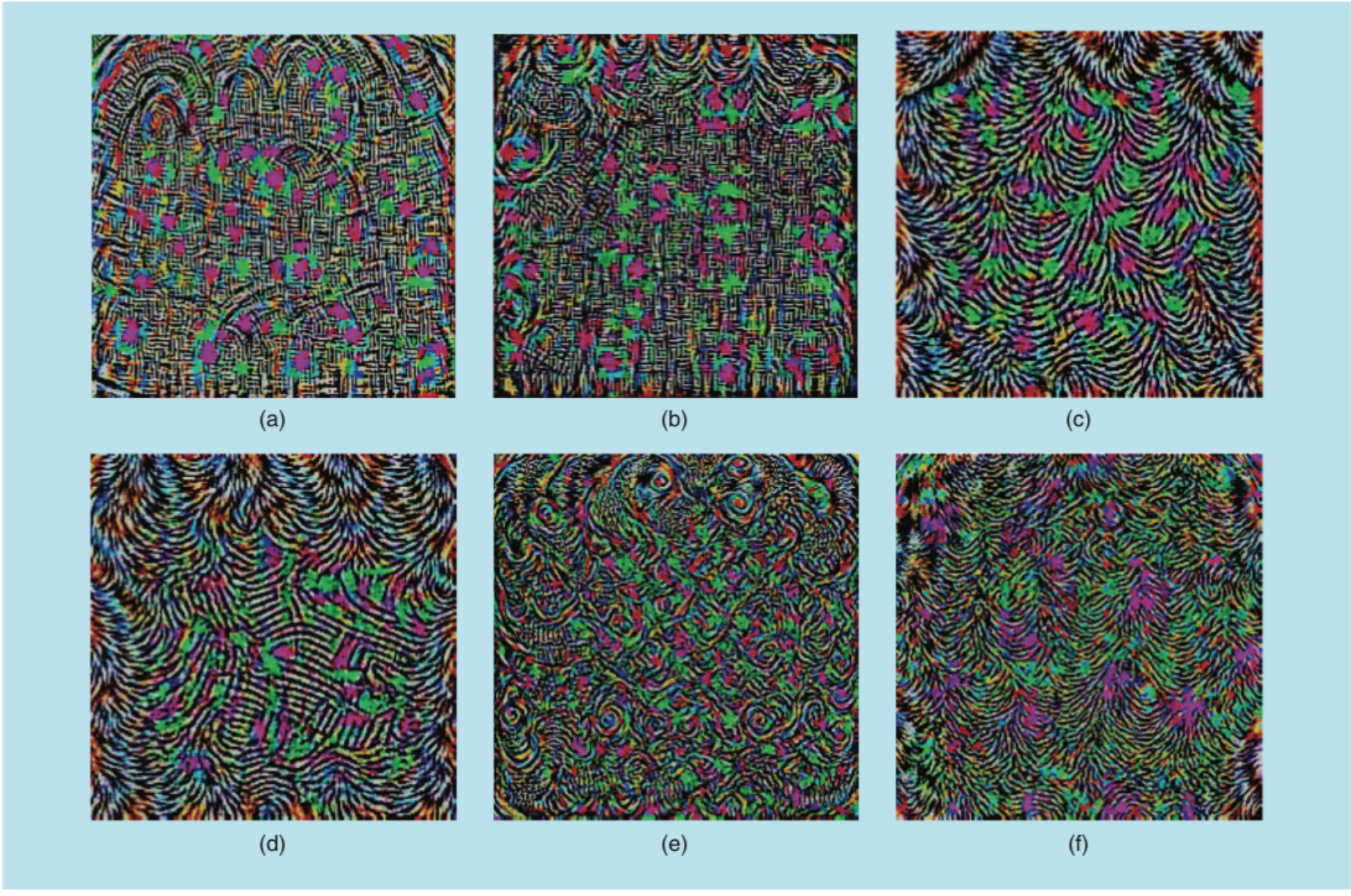


FIGURE 3. Universal perturbations computed for different deep neural network architectures. The pixel values are scaled for visibility. (a) CaffeNet, (b) VGG-F, (c) VGG-16, (d) VGG-19, (e) GoogLeNet, and (f) ResNet-152.

Deep Fool: Examples

FIGURE 4. Examples of natural images perturbed with the universal perturbation and their corresponding estimated labels with GoogLeNet. (a)–(h) Images belonging to the ILSVRC 2012 validation set. (i)–(l) Personal images captured by a mobile phone camera. (Figure used courtesy of [22].)

Deep Fool: Examples

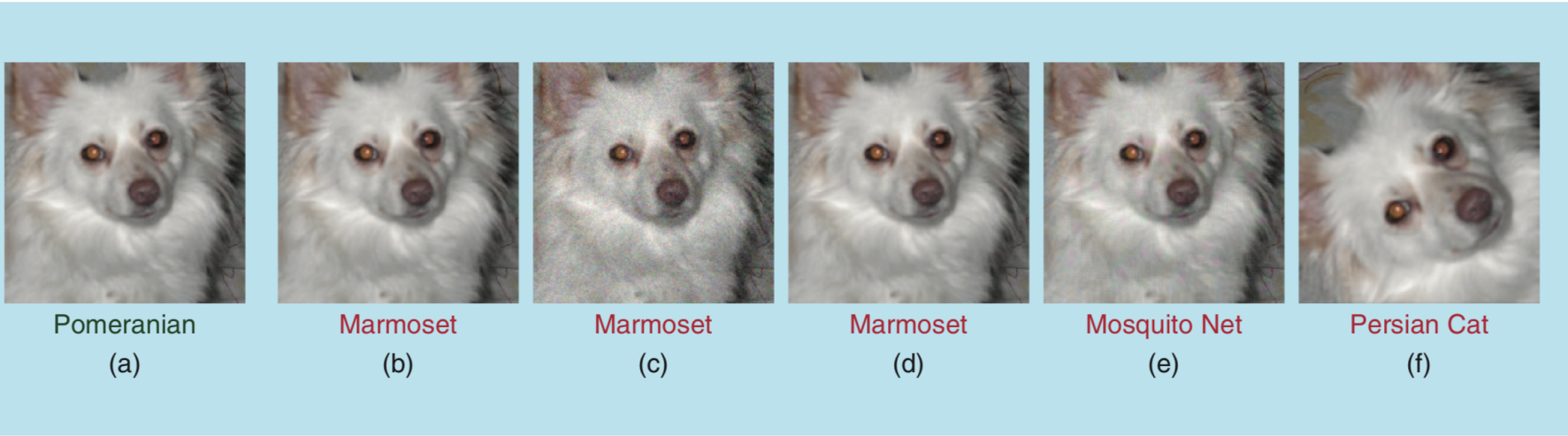


FIGURE 5. (a) The original image. The remaining images are minimally perturbed images (along with the corresponding estimated label) that misclassify the CaffeNet deep neural network. (b) Adversarial perturbation, (c) random noise, (d) semirandom noise with $m = 1{,}000$, (e) universal perturbation, (f) affine transformation. (Figure used courtesy of [17].)

Deep Fool: Examples

	VGG-F	CaffeNet	GoogLeNet	VGG-16	VGG-19	ResNet-152
VGG-F	**93.7%**	71.8%	48.4%	42.1%	42.1%	47.4%
CaffeNet	74.0%	**93.3%**	47.7%	39.9%	39.9%	48.0%
GoogLeNet	46.2%	43.8%	**78.9%**	39.2%	39.8%	45.5%
VGG-16	63.4%	55.8%	56.5%	**78.3%**	73.1%	63.4%
VGG-19	64.0%	57.2%	53.6%	73.5%	**77.8%**	58.0%
ResNet-152	46.3%	46.3%	50.5%	47.0%	45.5%	**84.0%**

Table : The rows indicate the architecture for which the universal perturbations is computed, and the columns indicate the architecture for which the fooling rate is reported.

Robust Physical-World Attacks on Deep Learning Visual Classification

Kevin Eykholt*[1], Ivan Evtimov*[2], Earlence Fernandes[2], Bo Li[3],
Amir Rahmati[4], Chaowei Xiao[1], Atul Prakash[1], Tadayoshi Kohno[2], and Dawn Song[3]

[1]University of Michigan, Ann Arbor
[2]University of Washington
[3]University of California, Berkeley
[4]Samsung Research America and Stony Brook University

Abstract

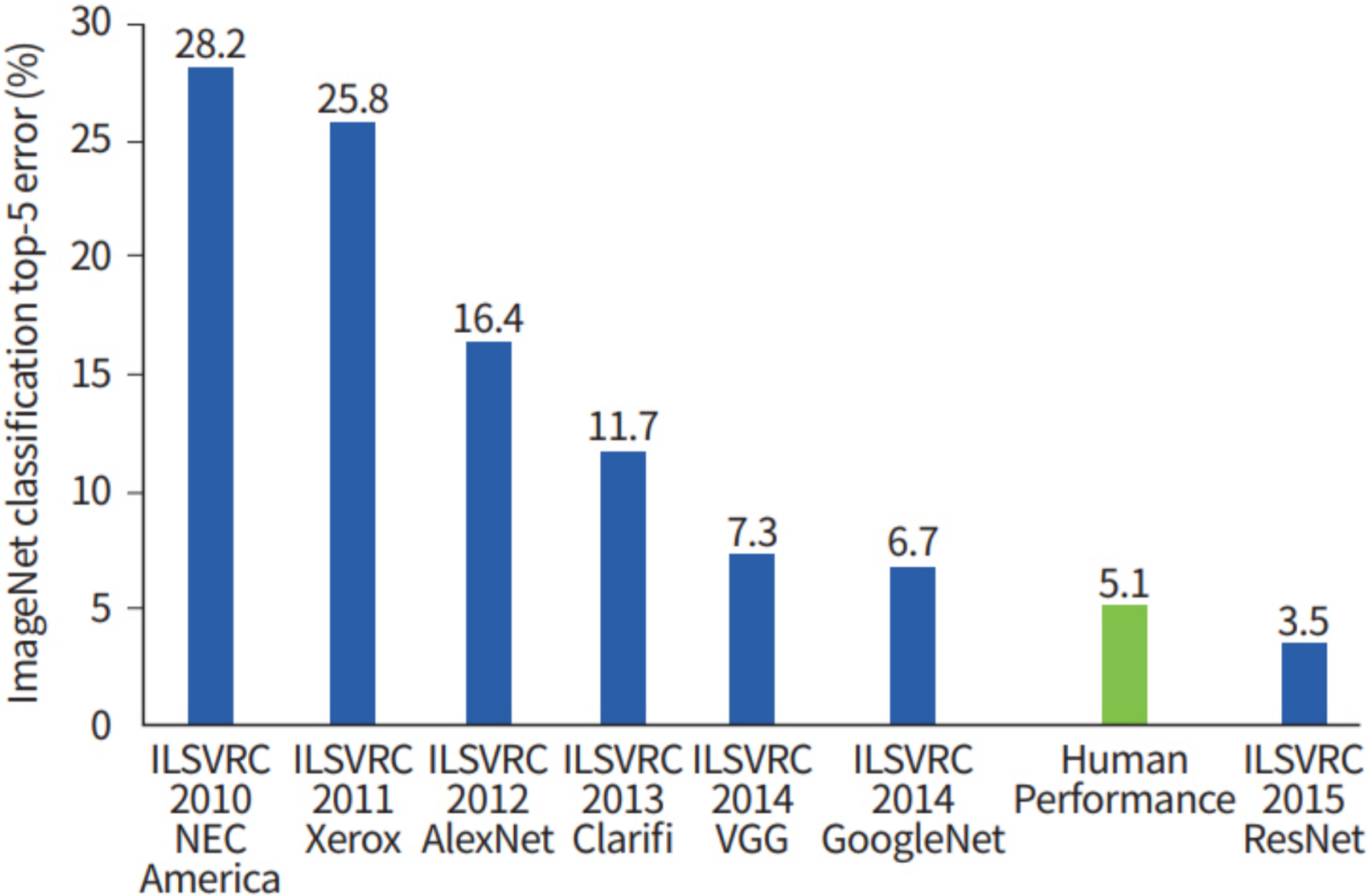
Recent studies show that the state-of-the-art deep neural networks (DNNs) are vulnerable to adversarial examples, resulting from small-magnitude perturbations added to the input. Given that that emerging physical systems are using DNNs in safety-critical situations, adversarial examples could mislead these systems and cause dangerous situations. Therefore, understanding adversarial examples in the physi-

these successes, they are increasingly being used as part of control pipelines in physical systems such as cars [8,17], UAVs [4,24], and robots [40]. Recent work, however, has demonstrated that DNNs are vulnerable to adversarial perturbations [5,9,10,15,16,22,25,29,30,35]. These carefully crafted modifications to the (visual) input of DNNs can cause the systems they control to misbehave in unexpected and potentially dangerous ways.

Structural perturbations

Structural perturbations can also cause the network to fail.

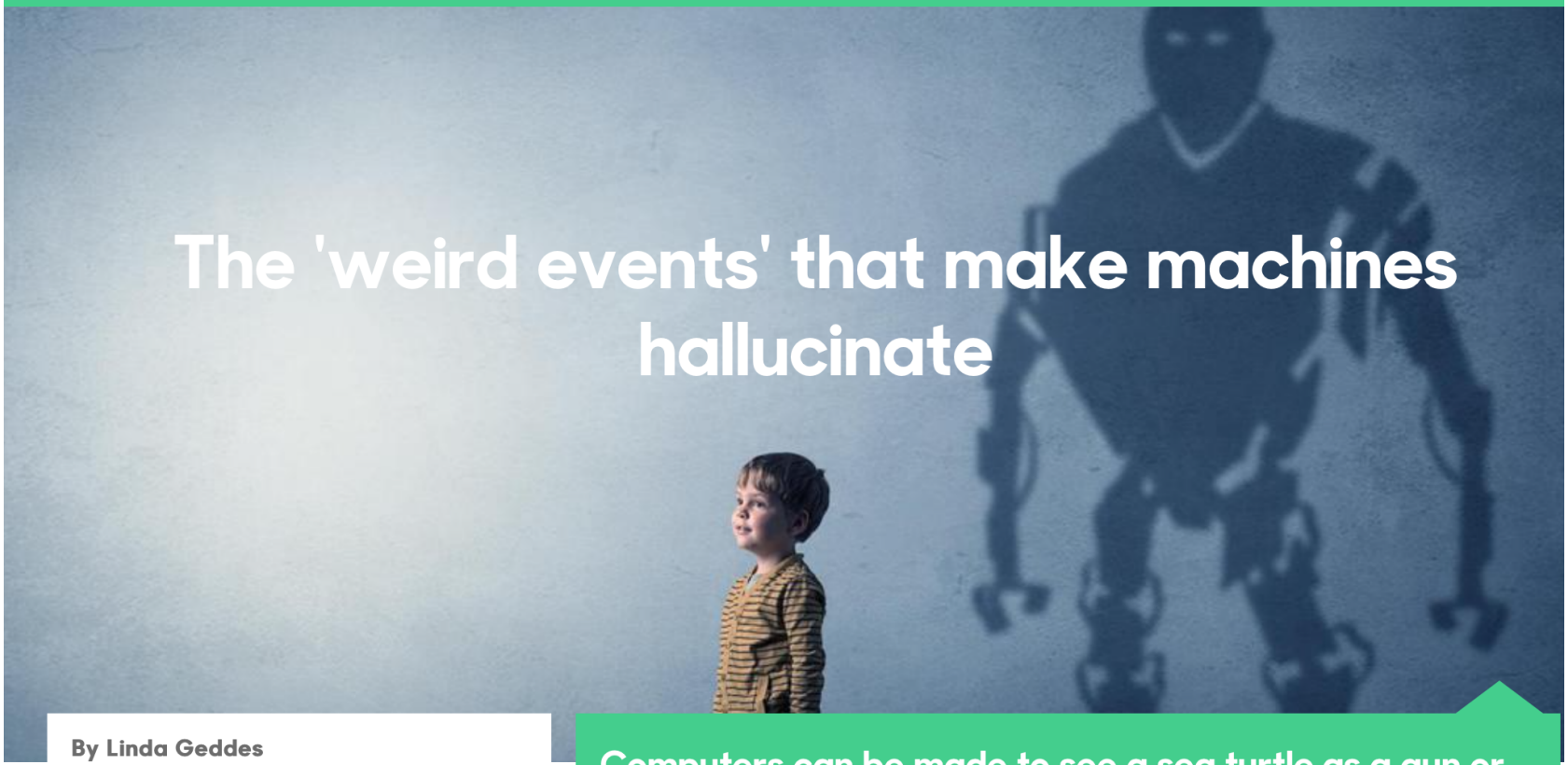
Before and after 2012 - The ImageNet competition



BBC Sign in | News | Sport | Weather | Shop | Reel | Travel | More | Search

machine minds | What is BBC Future? | Latest | Best of.. | Machine Minds | Future Now

The 'weird events' that make machines hallucinate



By Linda Geddes

5 December 2018

Computers can be made to see a sea turtle as a gun or hear a concerto as someone's voice, which is raising concerns about using artificial intelligence in the real world.

Strong confidence in deep learning

The New Yorker quotes Geoffrey Hinton (April 2017):

"They should stop training radiologists now."

The Instability Test

Given a neural network Φ that is able to classify images i.e.

$$\Phi(x_{\text{cat}}) = \text{cat},$$

find a perturbation x_δ with $\|x_\delta\| \leq \delta$, where $\delta > 0$ is small, such that

$\Phi(x_{\text{cat}} + x_\delta) = \text{firetruck}$ (or some other label that is not cat).

Finding such an x_δ is often referred to as an adversarial attack.

Theorem 3 (Bastounis, Hansen, Vlacic)

There is an uncountable family of classification functions $f : \mathbb{R}^{N_0} \to \{0,1\}$ *such that for any neural network dimensions* $\mathbf{N} = (N_L, N_{L-1}, \ldots, N_1, N_0)$ *with* $N_0, L \geq 2$ *and any* $0 < \epsilon < 1/(K+M)$ *where* M *is arbitrarily large and* $K \geq 3(N_1+1)\cdots(N_{L-1}+1)$ *we have the following. There exist uncountably many training sets* $\mathcal{T} = \{x^1, \ldots, x^K\}$ *and uncountably many classification sets* $\mathcal{C} = \{y^1, \ldots, y^M\}$ *such that there is a*

$$\tilde{\Phi} \in \operatorname*{argmin}_{\Phi \in \mathcal{NN}_{\mathbf{N},L}} C(v,w), \quad v_j = \Phi(x^j), \quad w_j = f(x^j),$$

where $1 \leq j \leq K$, *and* $C(v,w) = 0$ *iff* $v = w$, *such that*

$$\tilde{\Phi}(x) = f(x) \quad \forall x \in \mathcal{T} \cup \mathcal{C}.$$

However, there exists uncountably many $v \in \mathbb{R}^{N_0}$ *such that*

$$|\tilde{\Phi}(v) - f(v)| \geq 1/2, \qquad \|v - x\|_\infty \leq \epsilon \text{ for some } x \in \mathcal{T}.$$

Moreover, there is another neural network $\hat{\Phi}$ *such that*

$$\hat{\Phi}(x) = f(x) \quad \forall x \in \mathcal{B}_\epsilon^\infty(\mathcal{T} \cup \mathcal{C}).$$

The paradox of deep learning

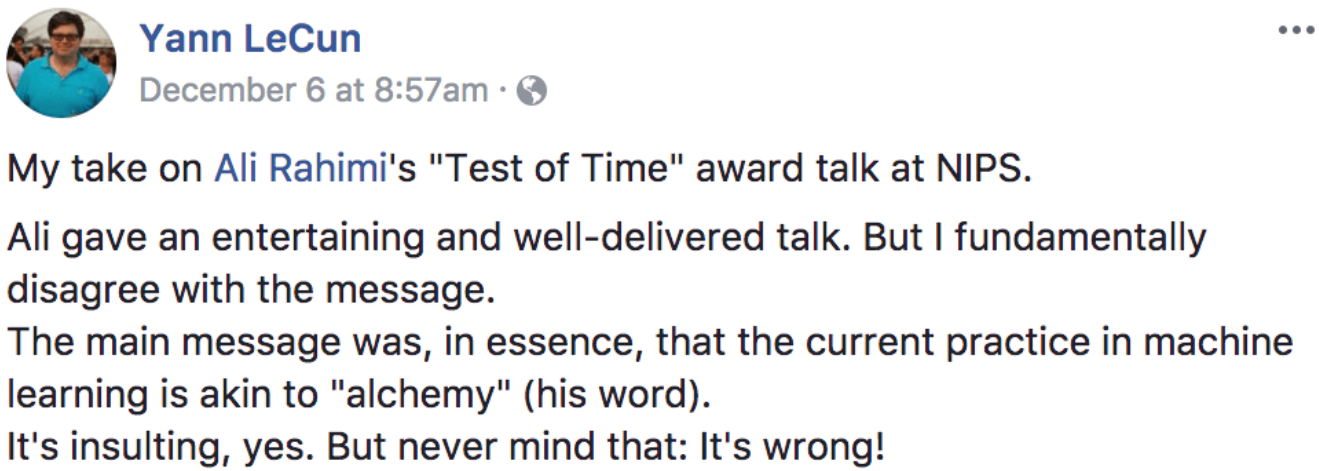
The paradox of the previous theorem:

- The trained network $\tilde{\Phi}$ becomes highly successful, yet incredibly unstable.
- However, there exists another neural network $\hat{\Phi}$ that has the same success rate and is stable.

Key question: Can the stable neural network be constructed in a (Turing) recursive way?

Do we know what we are doing?

Google's Ali Rahimi, winner of the Test-of-Time award 2017 (NIPS), "Machine learning has become alchemy. ... I would like to live in a society whose systems are built on top of verifiable, rigorous, thorough knowledge, and not on alchemy."



Yann LeCun
December 6 at 8:57am ·

My take on Ali Rahimi's "Test of Time" award talk at NIPS.

Ali gave an entertaining and well-delivered talk. But I fundamentally disagree with the message.
The main message was, in essence, that the current practice in machine learning is akin to "alchemy" (his word).
It's insulting, yes. But never mind that: It's wrong!

What could possibly go wrong?

-

AI replacing standard algorithms

nature
International journal of science

Letter | Published: 21 March 2018

Image reconstruction by domain-transform manifold learning

Bo Zhu, Jeremiah Z. Liu, Stephen F. Cauley, Bruce R. Rosen & Matthew S. Rosen

Nature **555**, 487–492 (22 March 2018) | Download Citation

Abstract

Image reconstruction is essential for imaging applications across the physical and life sciences, including optical and radar systems, magnetic resonance imaging, X-ray computed tomography, positron

Transforming image reconstruction with AI

nature methods

Research Highlights | Published: 27 April 2018

Imaging

AI transforms image reconstruction

Rita Strack

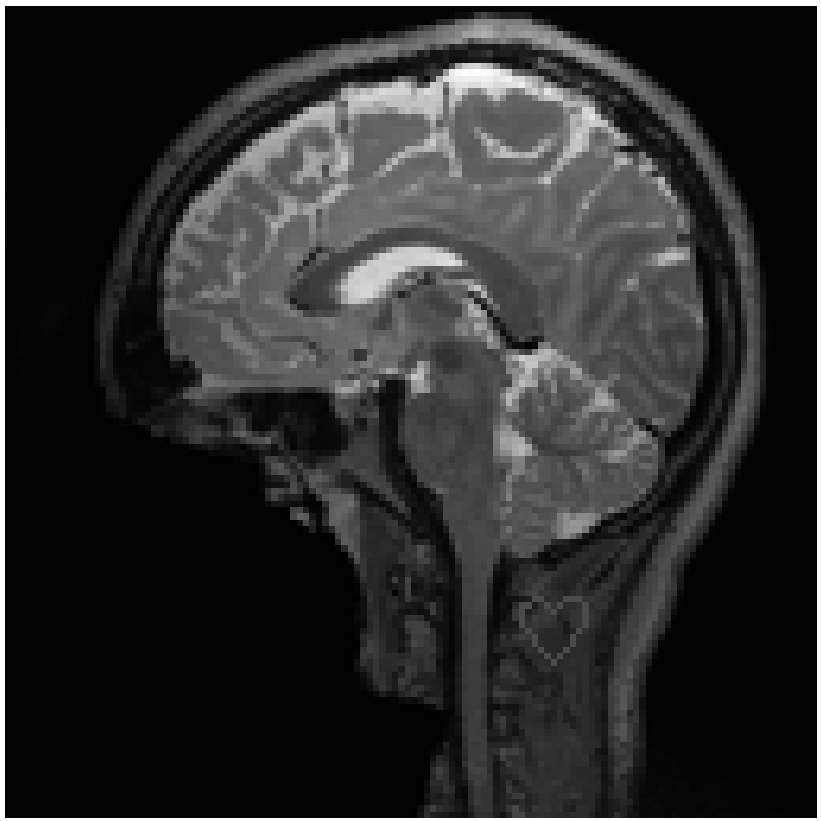
Nature Methods **15**, 309 (2018) | Download Citation

A deep-learning-based approach improves the speed, accuracy, and robustness of biomedical image reconstruction.

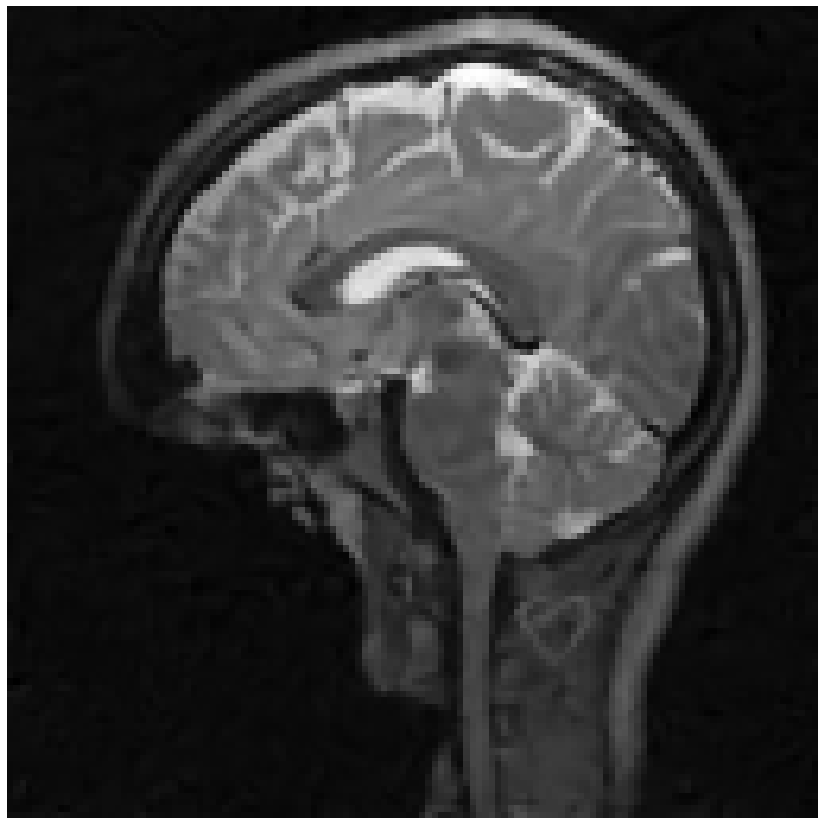
Instability of DL in Inverse Problems - MRI

Experiment from "On instabilities of deep learning in image reconstruction - Does AI come at a cost?", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen

Original



AUTOMAP Network

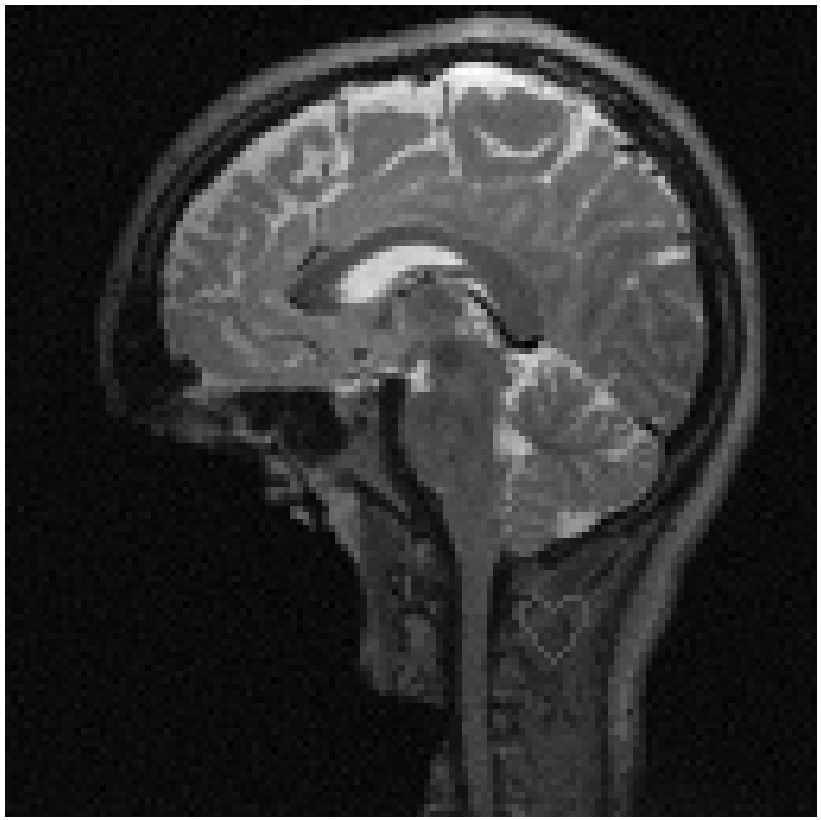


AUTOMAP network from "Image reconstruction by domain-transform manifold learning", B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, M. S. Rosen. *Nature* (March. 2018).

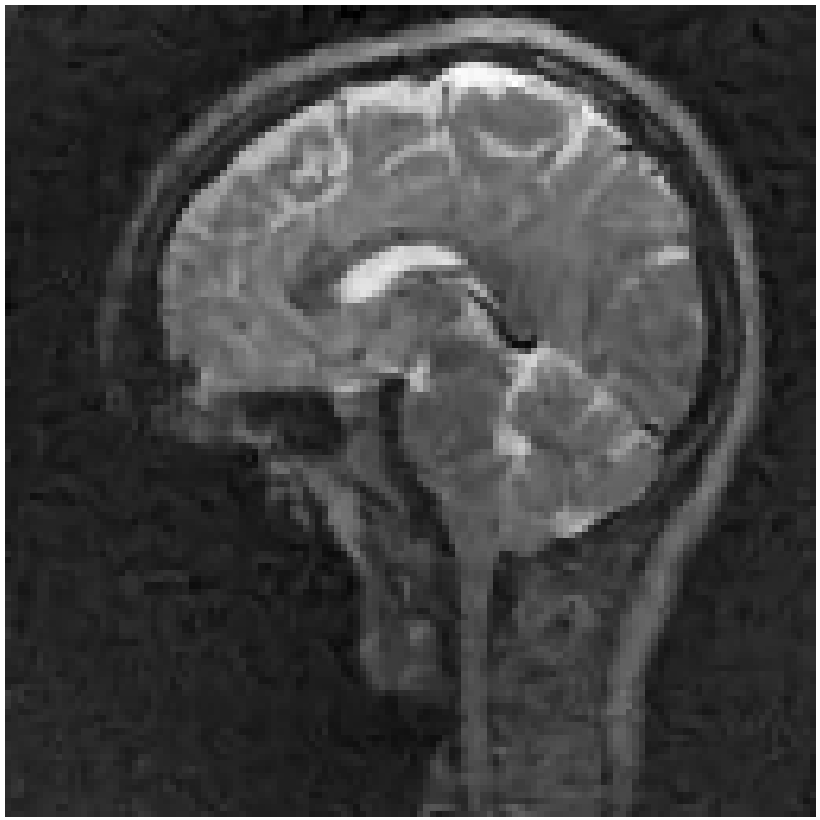
Instability of DL in Inverse Problems - MRI

Experiment from "On instabilities of deep learning in image reconstruction - Does AI come at a cost?", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen

Original + tiny pert.



AUTOMAP Network

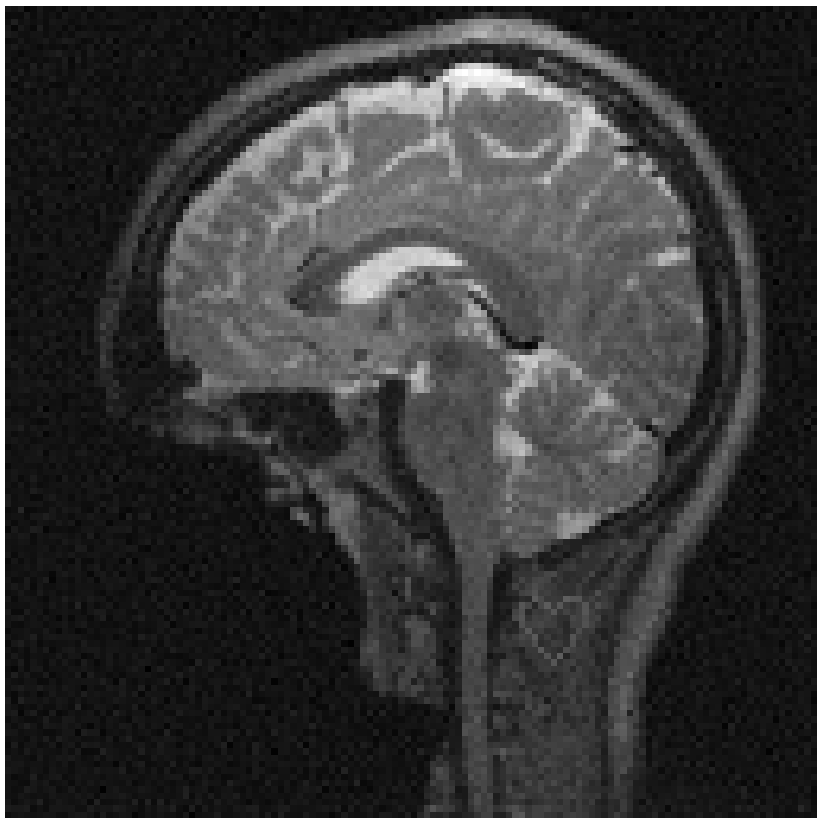


AUTOMAP network from "Image reconstruction by domain-transform manifold learning", B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, M. S. Rosen. *Nature* (March. 2018).

Instability of DL in Inverse Problems - MRI

Experiment from "On instabilities of deep learning in image reconstruction - Does AI come at a cost?", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen

Original + tiny pert.



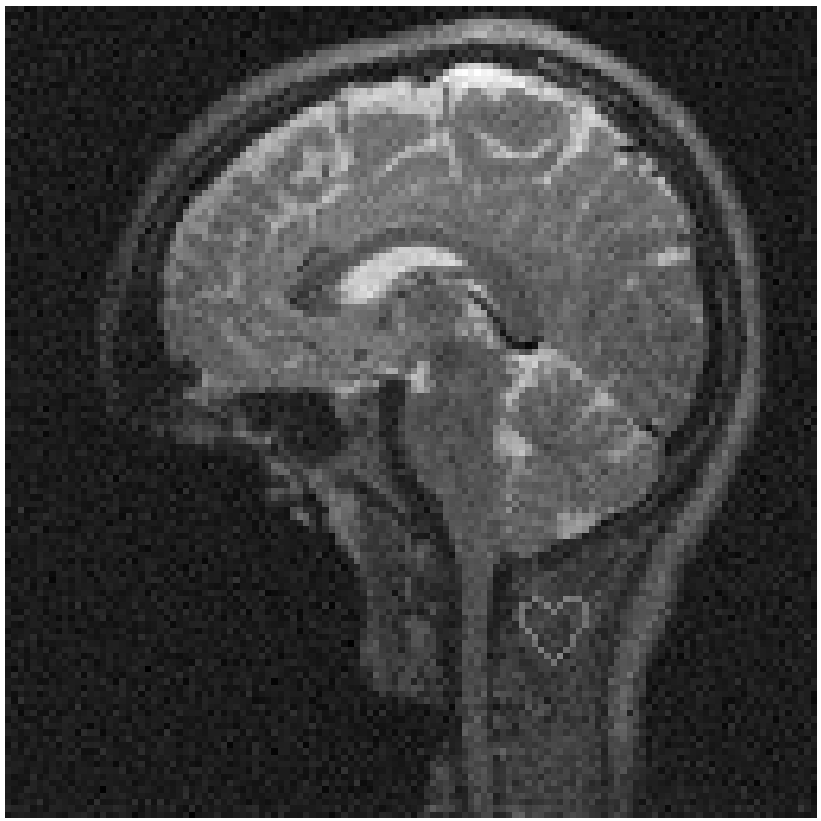
AUTOMAP Network

AUTOMAP network from "Image reconstruction by domain-transform manifold learning", B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, M. S. Rosen. *Nature* (March. 2018).

Instability of DL in Inverse Problems - MRI

Experiment from "On instabilities of deep learning in image reconstruction - Does AI come at a cost?", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen

Original + tiny pert.



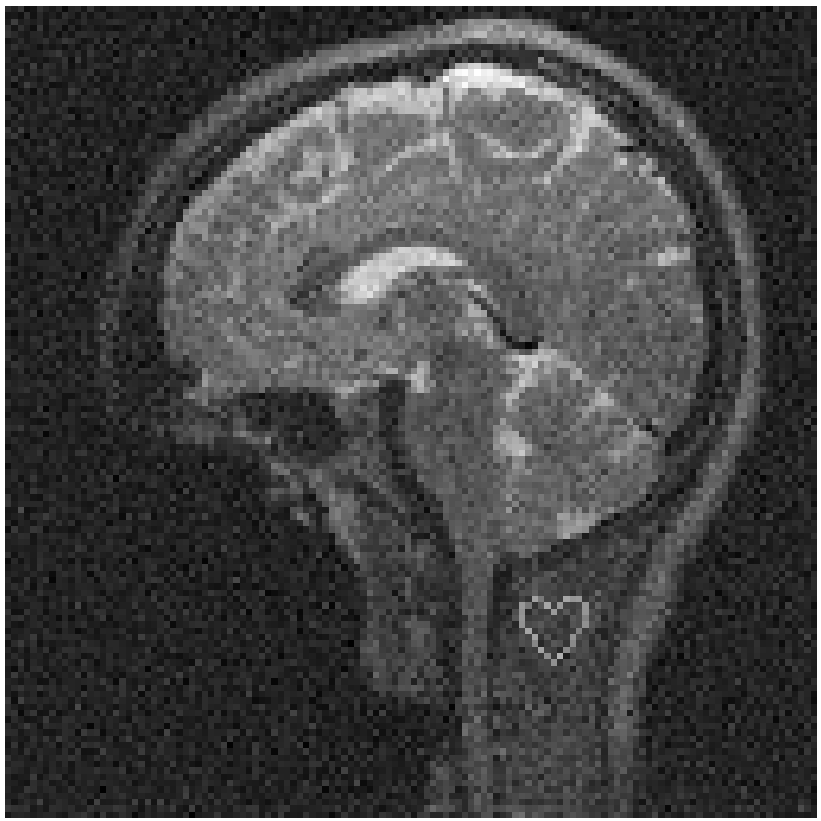
AUTOMAP Network

AUTOMAP network from "Image reconstruction by domain-transform manifold learning", B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, M. S. Rosen. *Nature* (March. 2018).

Instability of DL in Inverse Problems - MRI

Experiment from "On instabilities of deep learning in image reconstruction - Does AI come at a cost?", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen

Original + tiny pert.

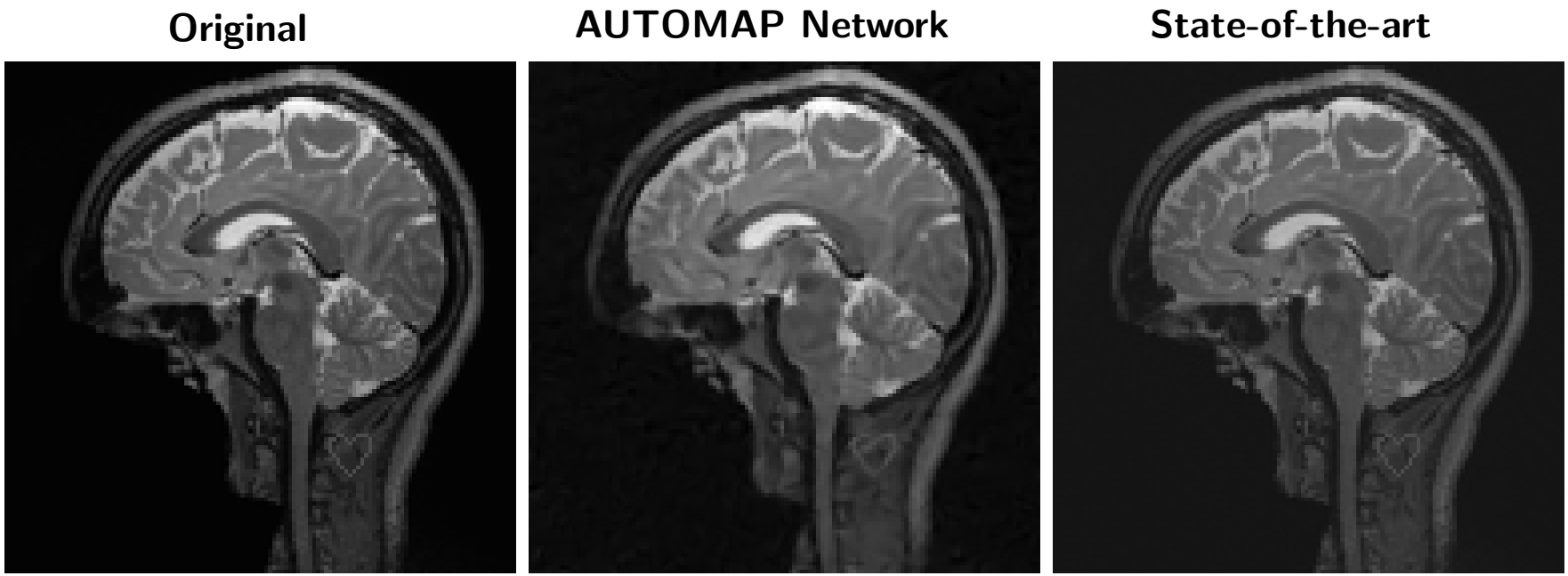


AUTOMAP Network

AUTOMAP network from "Image reconstruction by domain-transform manifold learning", B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, M. S. Rosen. *Nature* (March. 2018).

Comparison with state-of-the-art

Experiment from "On instabilities of deep learning in image reconstruction - Does AI come at a cost?", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen

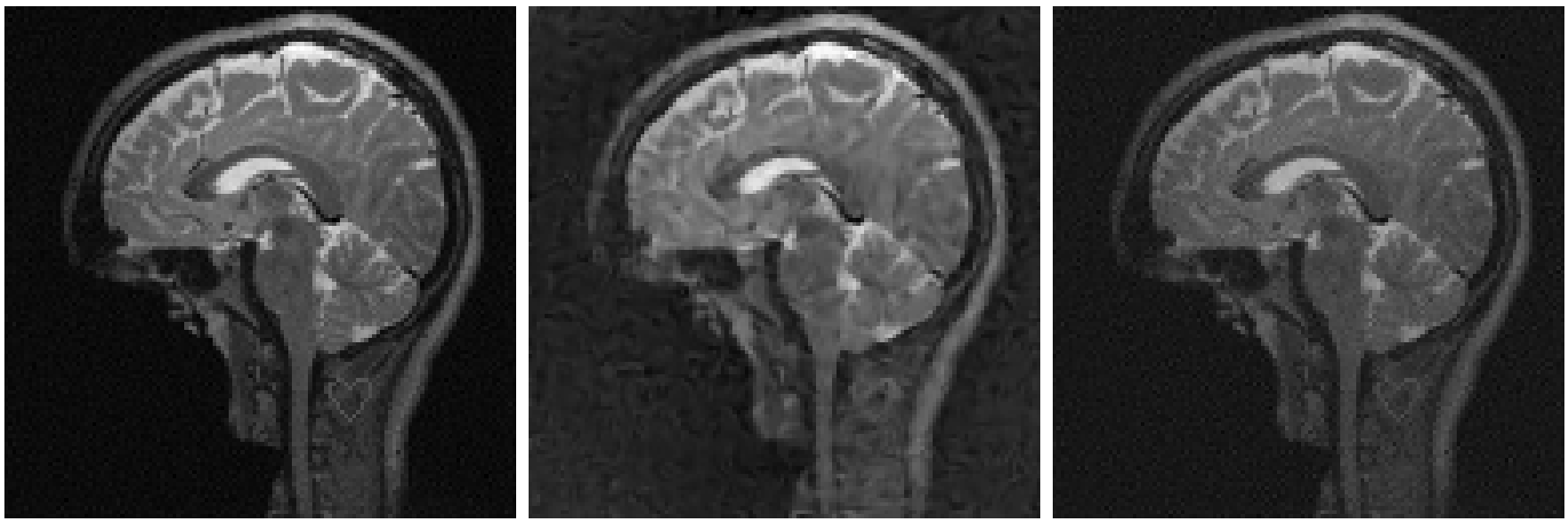


AUTOMAP network from "Image reconstruction by domain-transform manifold learning", B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, M. S. Rosen. *Nature* (March. 2018).

Comparison with state-of-the-art

Experiment from "On instabilities of deep learning in image reconstruction - Does AI come at a cost?", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen

Original + tiny pert. | **AUTOMAP Network** | **State-of-the-art**



AUTOMAP network from "Image reconstruction by domain-transform manifold learning", B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, M. S. Rosen. *Nature* (March. 2018).

Comparison with state-of-the-art

Experiment from "On instabilities of deep learning in image reconstruction - Does AI come at a cost?", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen

Original + tiny pert. **AUTOMAP Network** **State-of-the-art**

AUTOMAP network from "Image reconstruction by domain-transform manifold learning", B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, M. S. Rosen. *Nature* (March. 2018).

Comparison with state-of-the-art

Experiment from "On instabilities of deep learning in image reconstruction - Does AI come at a cost?", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen

Original + tiny pert. **AUTOMAP Network** **State-of-the-art**

AUTOMAP network from "Image reconstruction by domain-transform manifold learning", B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, M. S. Rosen. *Nature* (March. 2018).

Comparison with state-of-the-art

Experiment from "On instabilities of deep learning in image reconstruction - Does AI come at a cost?", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen

Original + tiny pert. **AUTOMAP Network** **State-of-the-art**

AUTOMAP network from "Image reconstruction by domain-transform manifold learning", B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, M. S. Rosen. *Nature* (March. 2018).

What does deep learning actually learn?

Conjecture 1 (False structures in classification)

The current training process in deep learning for classification forces the neural network to learn a different (false) structure and not the actual structure of the classification problem. There are three main components:

(i) *(**Success**) The false structure correlates well with the original structure, hence one gets a high success rate.*

(ii) *(**Instability**) The false structure is unstable, and thus the network is susceptible to adversarial attacks.*

(iii) *(**Simplicity**) The false structure is much simpler than the desired structure, and hence easer to learn e.g. fewer data are needed and the numerical algorithm used in the training easily converges to the neural network that captures the false structure.*

A thought experiment

Consequences of Conjecture 1

Negative consequences:

(i) The success of deep learning in classification is not due to the fact that networks learn the structures that humans associate with image recognition, but rather that the network picks up unstable false structures in images that are potentially impossible for humans to detect. This means that instability, and hence vulnerability to adversarial attacks, can never be removed until one guarantees that no false structure is learned. This means a potential complete overhaul of modern AI.

(ii) The success is dependent of the simple yet unstable structures, thus the AI does not capture the intelligence of a human.

(iii) Since one does not know which structure the network picks up, it becomes hard to conclude what the neural network actually learns, and thus harder to trust its prediction. What if the false structure gives wrong predictions?

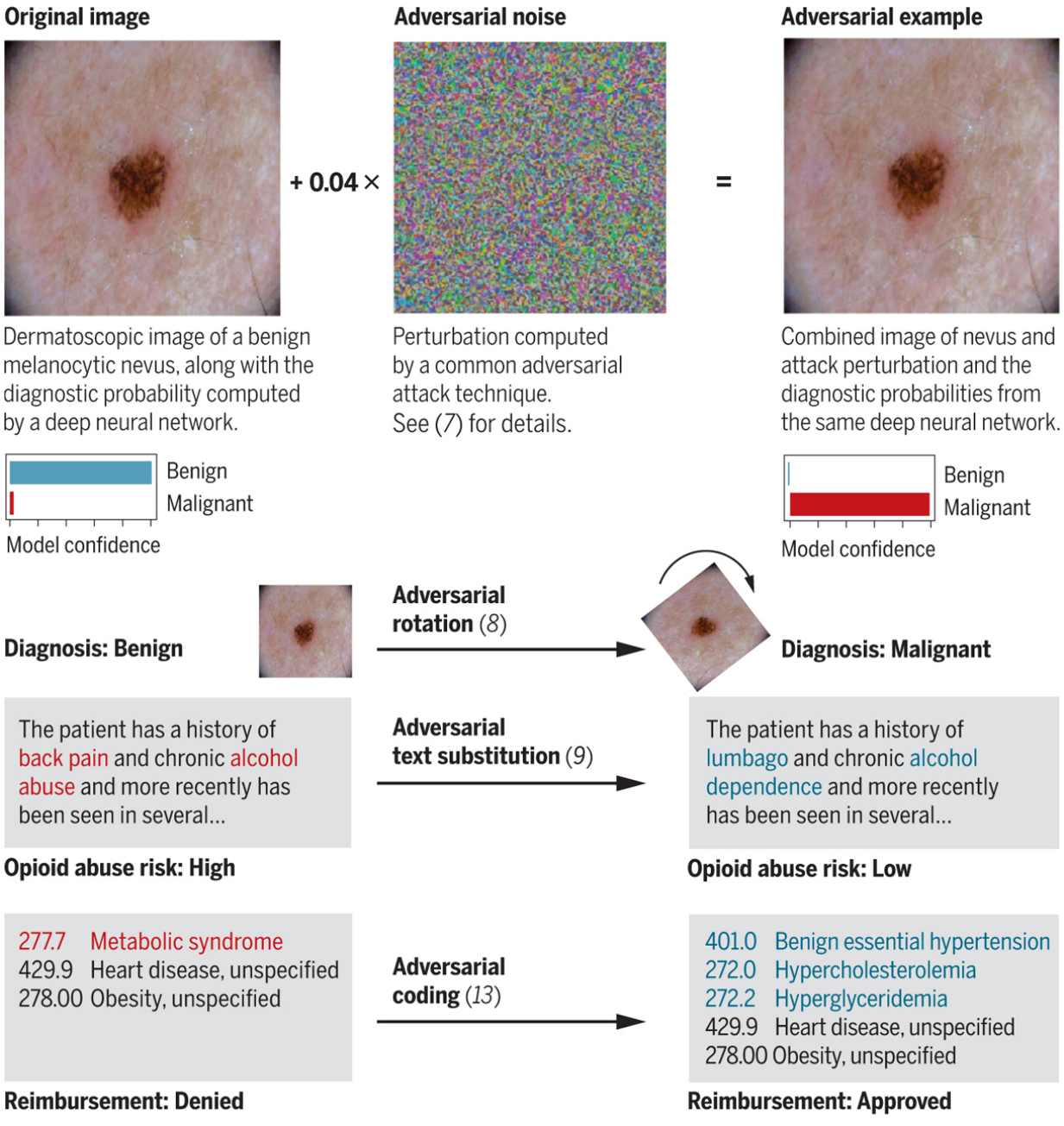
Consequences of Conjecture 1

Positive consequences:

(I) Deep learning captures structures that humans cannot detect, and these structures require very little data and computing power in comparison to the true original structures, however, they generalise rather well compared to the original structure. Thus, from an efficiency point of view, the human brain may be a complete overkill for certain classification problems, and deep learning finds a mysterious effective way of classifying.

(II) The structure learned by deep learning may have information that the human may not capture. This structure could be useful if characterised properly. For example, what if there is structural information in the data that allows for accurate prediction that the original structure could not do?

What could go wrong?



Definition 4 (The original structure and false structures)

Consider $\mathcal{M} \subset \mathbb{R}^d$ and a string $L = \{\alpha_1, \ldots, \alpha_N\}$ of unique predicates on $\mathcal{M}$, where $N \in \mathbb{N}$ such that for each $x \in \mathcal{M}$ there is a unique $\alpha_j \in L$ such that $\alpha_j(x) = \text{true}$. For such x define $f(x) = j$. We say that the pair (f, L) is *the original structure*. A *false structure for f relative to $\mathcal{T} \subsetneq \mathcal{M}$* is a pair (g, L'), where $L' = \{\beta_1, \ldots, \beta_N\}$ a collection of unique predicates with $g : \mathcal{M} \to \{1, \ldots, N\}$ such that $g(x) = j$ iff $\beta_j(x) = \text{true}$. Moreover, $\beta_j \neq \alpha_j$ for all $j \in \{1, \ldots, N\}$ and

$$g(x) = j \text{ iff } f(x) = j \, \forall \, x \in \mathcal{T}, \tag{1}$$

moreover

$$\mathcal{C} = \{x \in \mathcal{M} \setminus \mathcal{T} \mid f(x) = i,\, g(x) = j, \text{ for some } i \neq j\} \neq \emptyset. \tag{2}$$

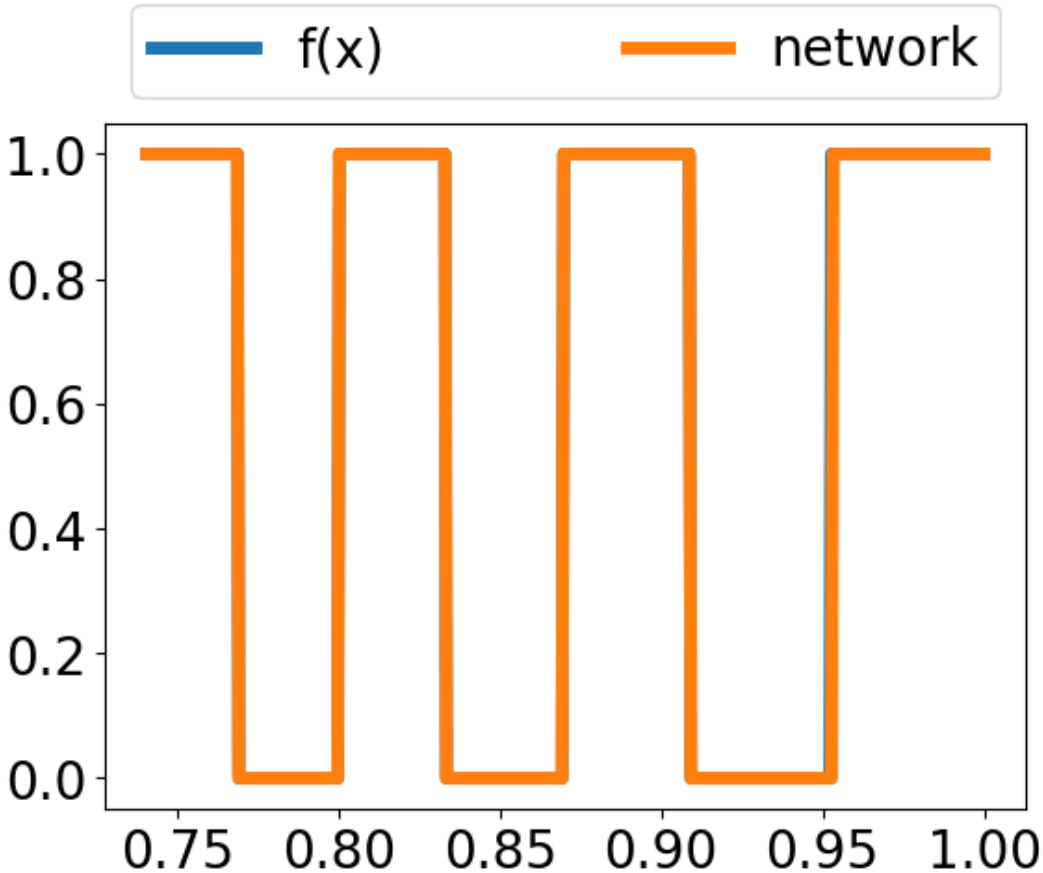
We say that g is a partial false structure if $\alpha_j \neq \beta_j$ for at least two different $j \in \{1, \ldots, N\}$ (as opposed to all).

The original structure: Case 1

A simple example are the functions

$$f_a : \mathcal{M} \to \{0,1\}, \quad \mathcal{M} = [b,1] \times [0,1], \quad f_a(x) = \left\lceil \frac{a}{x_1} \right\rceil \mod 2, \tag{3}$$

for $a > 0$ and $0 < b < 1$.



The original structure

Consider the predicates

$$\alpha_0(x) = \left\lceil \frac{a}{x_1} \right\rceil \text{ is odd.}$$

$$\alpha_1(x) = \left\lceil \frac{a}{x_1} \right\rceil \text{ is even.}$$

Let $L = \{\alpha_1, \alpha_2\}$ then (f_a, L) is the original structure.

We note that this function is constant on each of the intervals

$$\frac{a}{k+1} \leq x_1 < \frac{a}{k},$$

where $k \in \mathbb{N}$.

To ensure that f_a is stable with respect to perturbations of size $\epsilon > 0$ on its input, we will ensure that each of our samples of f_a lies at least ϵ away from each of these jump discontinuities. Hence, we choose the samples from the set

$$\mathcal{S}_\epsilon = \bigcup_{k=a}^{K} \left(\frac{a}{k+1} + \epsilon, \frac{a}{k} - \epsilon \right). \tag{4}$$

The false structure

For the learning task we now consider two sets of size r,

$$\mathcal{T}_0^r = \{(x_1^{(i)}, 0)\}_{i=1}^r, \quad \mathcal{T}_\delta^r = \{(x_1^{(i)}, x_2^{(i)})\}_{i=1}^r,$$

$$x_2^{(i)} = \delta f_a(x_1^{(i)}), \qquad x_1^{(i)} \in \mathcal{S}_\epsilon,$$

where $0 < \delta < \epsilon$. Note that $\mathcal{T}_\delta^r$ gives rise to a false structure as the next proposition shows.

Proposition 5

Consider the predicates β_0 and β_1 defined by

$$\beta_0(x) = x_2 \text{ is } 0, \quad \beta_1(x) = x_2 \text{ is not } 0.$$

Define $g : \mathcal{M} \to \{0, 1\}$ by $g(x) = 1$ when $x_2 \neq 0$ and $g(x) = 0$ otherwise. Let $L' = \{\beta_0, \beta_1\}$. Then (g, L') is a false structure for (f_a, L) relative to $\mathcal{T}_\delta^r$.

The good minimiser exists

Proposition 6 (Existence of stable and accurate network)

Let $\sigma(x) = 1/(1+\exp(-x))$, $x \in \mathbb{R}$ be the sigmoid function, and let $C : \mathbb{R}^r \times \mathbb{R}^r \to \mathbb{R}$ be the cross entropy loss function for binary classification, that is

$$C(v, w) = \sum_{j=1}^{r} -w_j \log(\sigma(v_j)) - (1 - w_j) \log(1 - \sigma(v_j)). \qquad (5)$$

Let f_a be as in (3). Then, for any $\eta > 0$, there exists a two layer neural network $\Psi \in \mathcal{NN}_{\mathbf{N},2}$ with $\mathbf{N} = [1, 4K, 2]$ using the ReLU activation function, such that

$$f_a(x) = \lfloor \sigma \circ \Psi(x) \rceil, \quad x \in \mathcal{S}_\epsilon \times \mathbb{R},$$

where $\lfloor \cdot \rceil$ denotes rounding to the closest integer, and for any subset $\mathcal{T} = \{x^{(1)}, \ldots, x^{(r)}\} \subset \mathcal{S}_\epsilon \times \mathbb{R}$, $v = \{\Psi(x)\}_{x \in \mathcal{T}}$ and $w = \{f_a(x)\}_{x \in \mathcal{T}}$ we have $C(v, w) \leq \eta$.

Test: which structure does the network learn?

We fixed $a = 20$, $K = 26$, $\epsilon = 10^{-2}$ and $\delta = 10^{-4}$ and trained four neural networks $\Psi_1, \Psi_2, \Psi_3, \Psi_4 \in \mathcal{NN}_{\mathbf{N},2}$ with $\mathbf{N} = [1, 4K, 2]$ and ReLU activation function. The networks Ψ_i, $i = 1, \ldots, 4$ were trained on the sets $\mathcal{T}_\delta^7$, $\mathcal{T}_0^{5000}$, $\mathcal{T}_\delta^{5000}$ and $\mathcal{T}_0^{10000}$, respectively.

Define the test sets

$$\begin{aligned} \mathcal{C}_0 &= \{(x_1, 0) \,|\, x_1 \in [b, 1]\} \\ \mathcal{C}_\delta &= \{(x_1, x_2) \,|\, x_1 \in [b, 1], x_2 = \delta f_a(x_1)\}. \end{aligned} \tag{6}$$

Test: which structure does the network learn?

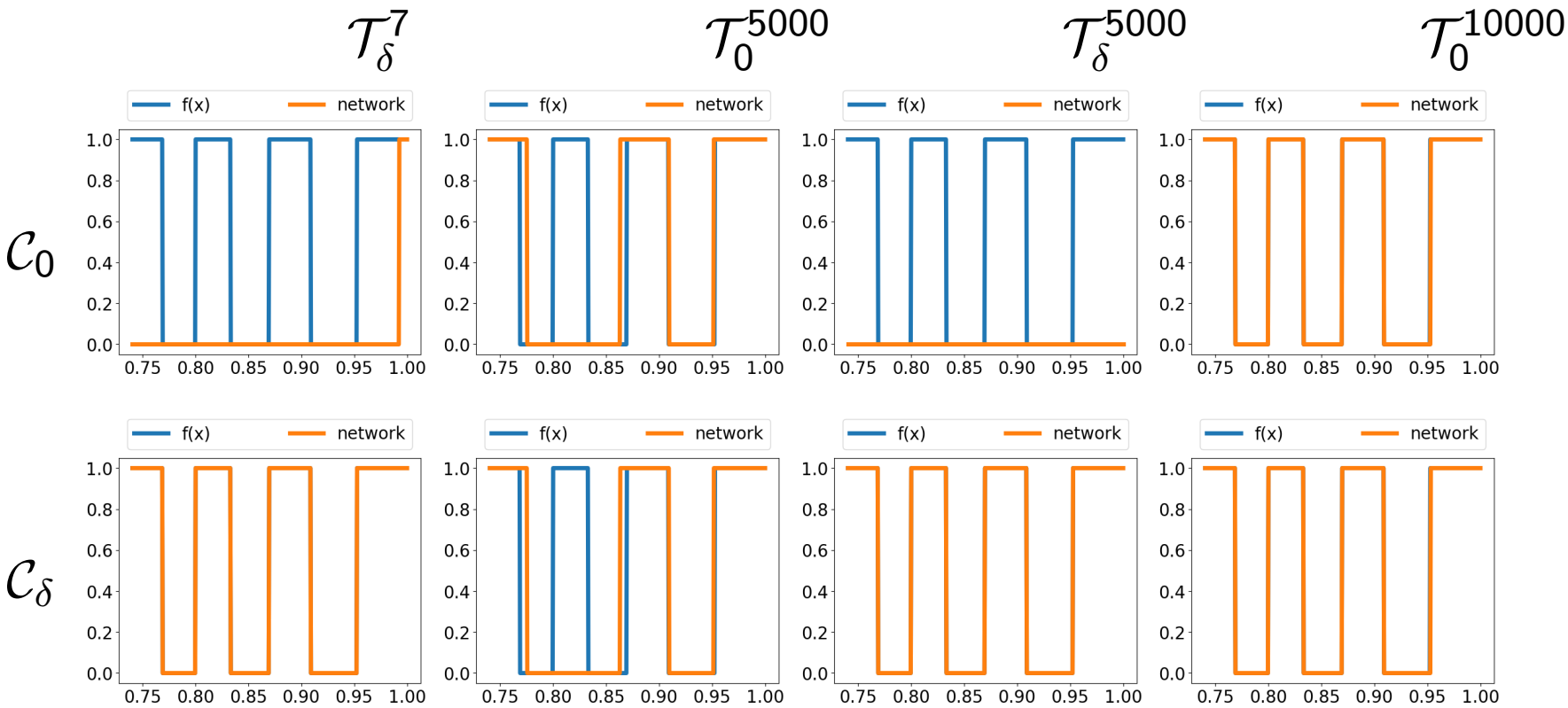


Figure : The graphs shows the output of $f_a(x)$ and $\lfloor \sigma(\Psi_i(x)) \rceil$ for x in the two sets $\mathcal{C}_0$ (top row) and $\mathcal{C}_\delta$ (bottom row). The networks Ψ_i, $i = 1, \ldots, 4$ have been trained on the sets $\mathcal{T}_\delta^7$, $\mathcal{T}_0^{5000}$, $\mathcal{T}_\delta^{5000}$ and $\mathcal{T}_\delta^{10000}$, respectively, and are shown from left to right.

Test: which structure does the network learn?

Conclusion:

- Ψ_1, trained on $\mathcal{T}_\delta^7$ (7 samples) learns the false structure g (up to a tiny interval).
- Ψ_2, trained on $\mathcal{T}_0^{5000}$, does not learn the original structure f_a nor the false structure g.
- Ψ_3, trained on $\mathcal{T}_\delta^{5000}$, learns the false structure g.
- Ψ_4, trained on $\mathcal{T}_\delta^{10000}$, learns the original structure f_a.

Examples of false structures: Case 2

Consider $\mathcal{M} =$

Figure : The collection $\mathcal{M}$ of images.

$\mathcal{M}$ is the collection of 32×32 grey scale images with a 3-pixel wide either horizontal or vertical light stripe on a dark background as shown in Figure 5. The colour code is as follows: -0.01 is black and $1 + 0.01$ is white. Hence, numbers between -0.01 and 0.01 yield variations of black and numbers between $1 - 0.01$ and $1 + 0.01$ give variations of white.

Examples of false structures: Case 2

Define the predicates

$$\alpha_0(x) = x \text{ has a light horizontal stripe,}$$

$$\alpha_1(x) = x \text{ has a light vertical stripe,}$$

and $L = \{\alpha_0, \alpha_1\}$. Define $f(x) = 0$ when $\alpha_0(x) = \text{true}$, and $f(x) = 1$ when $\alpha_1(x) = \text{true}$. We define (f, L) to be the original structure.

Testing the trained network

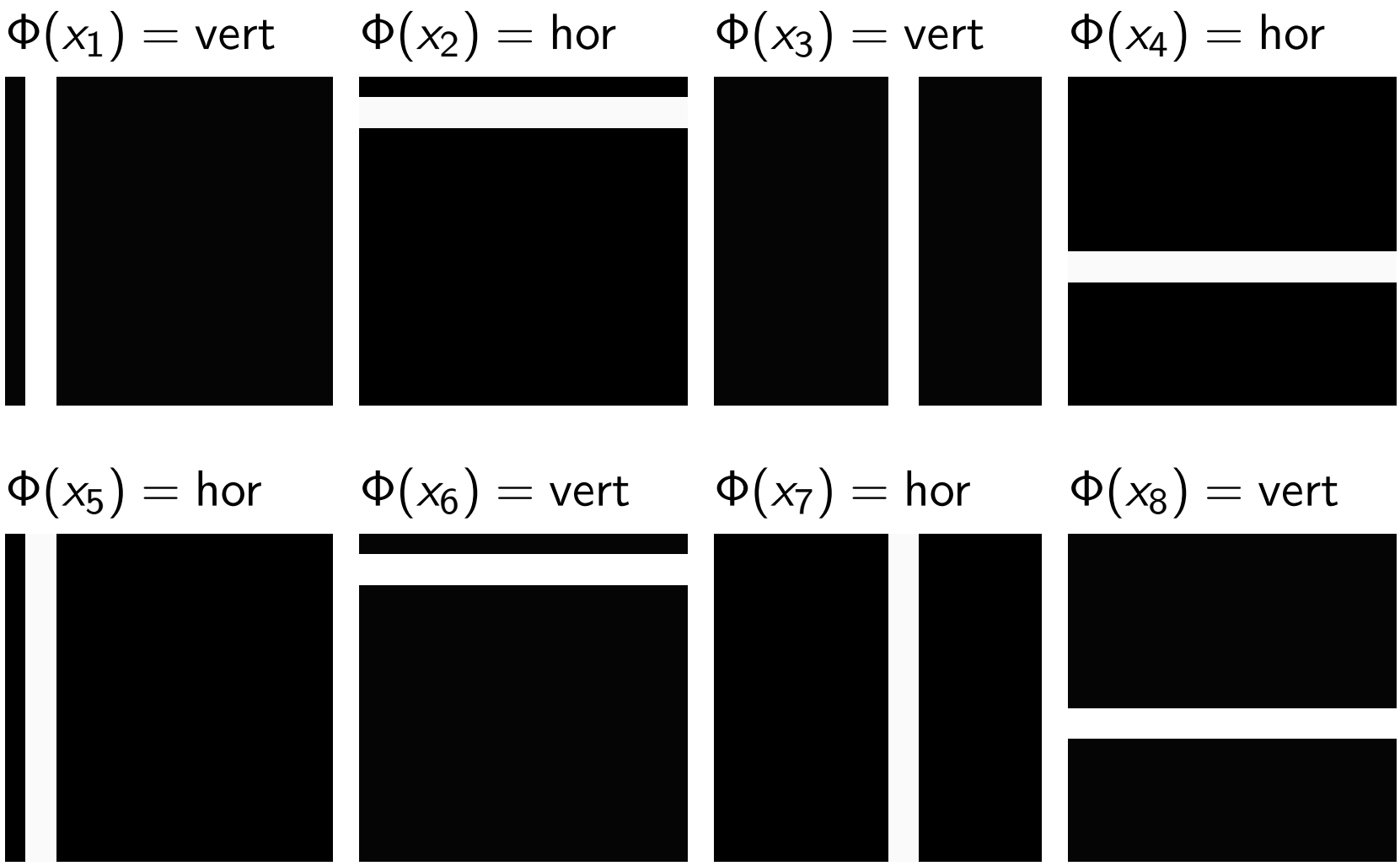


Figure : Test of a trained network.

Which false structure did the network learn?

Clearly the network Φ did not learn the original structure. The question is which false structure did it learn?

Examples of false structures: Case 2

Define $\tilde{\mathcal{M}}$ as follows:

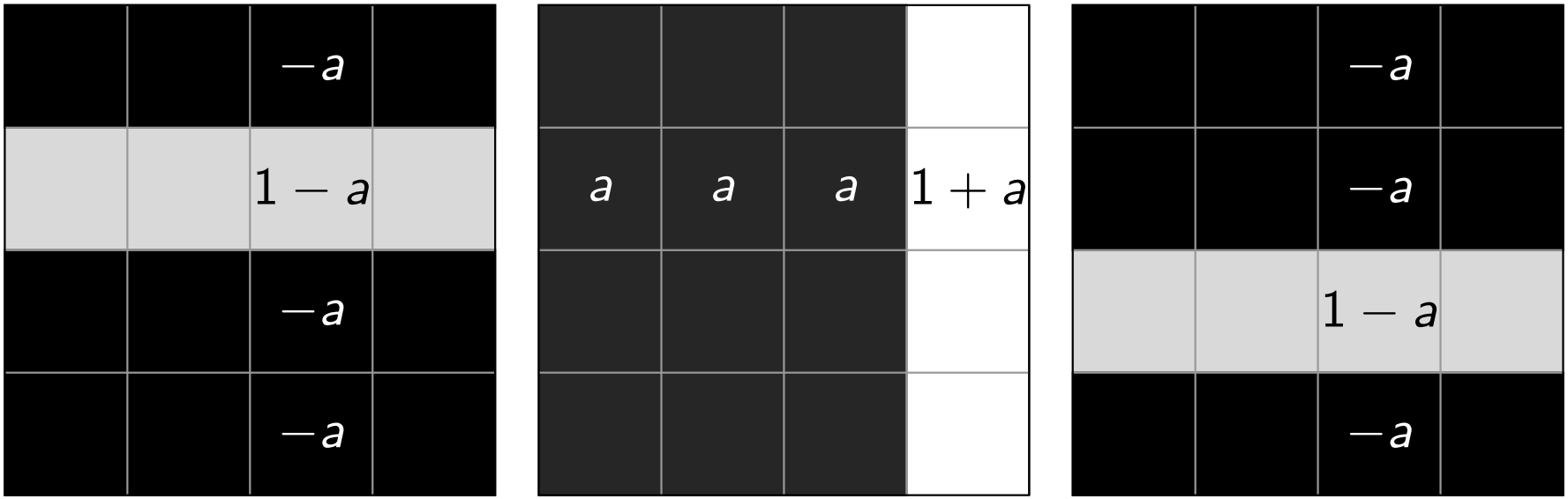


Figure : The colour code for the images defining $\tilde{\mathcal{M}} \subset \mathcal{M}$. The horizontal lines have colours $1 - a$ in the light area and $-a$ in the dark area. The vertical lines have a in the dark area and $1 + a$ in the light area.

Examples of false structures: Case 2

Proposition 7

Consider the predicates β_0 and β_1 on $\mathcal{M}$ defined by

$$\beta_0(x) = \textit{The sum of the pixel values of } x \textit{ are } \leq 96,$$

$$\beta_1(x) = \textit{The sum of the pixel values of } x \textit{ are } > 96,$$

and let $L' = \{\beta_0, \beta_1\}$. Let $g(x) = 0$ when $\beta_0(x) = $ true, and $g(x) = 1$ when $\beta_1(x) = $ true. Then (g, L') is a false structure for (f, L) relative to $\tilde{\mathcal{M}} \setminus \mathcal{O}$ where $\mathcal{O}$ denotes the images in $\mathcal{M}$ with colour code 0 in the dark area and colour code 1 in the light area.

Examples of false structures: Case 2

We have trained the network on

$$\mathcal{T} \subset \tilde{\mathcal{M}}$$

and tested the network on $\mathcal{C} \subset \tilde{\mathcal{M}}$ as well as on

$$\hat{\mathcal{C}} = \{x \,|\, x = \zeta(y), y \in \tilde{\mathcal{M}}\},$$

where $\zeta : [-0.01, 1 + 0.01] \setminus \{0, 1\} \to [-0.01, 1 + 0.01]$ is given as follows. Let $a > 0$ then $\zeta(a) = -a$, $\zeta(-a) = a$, $\zeta(1 - a) = 1 + a$, $\zeta(1 + a) = 1 - a$. Moreover, ζ acting on an image means pixelwise operations.

Examples of false structures: Case 2

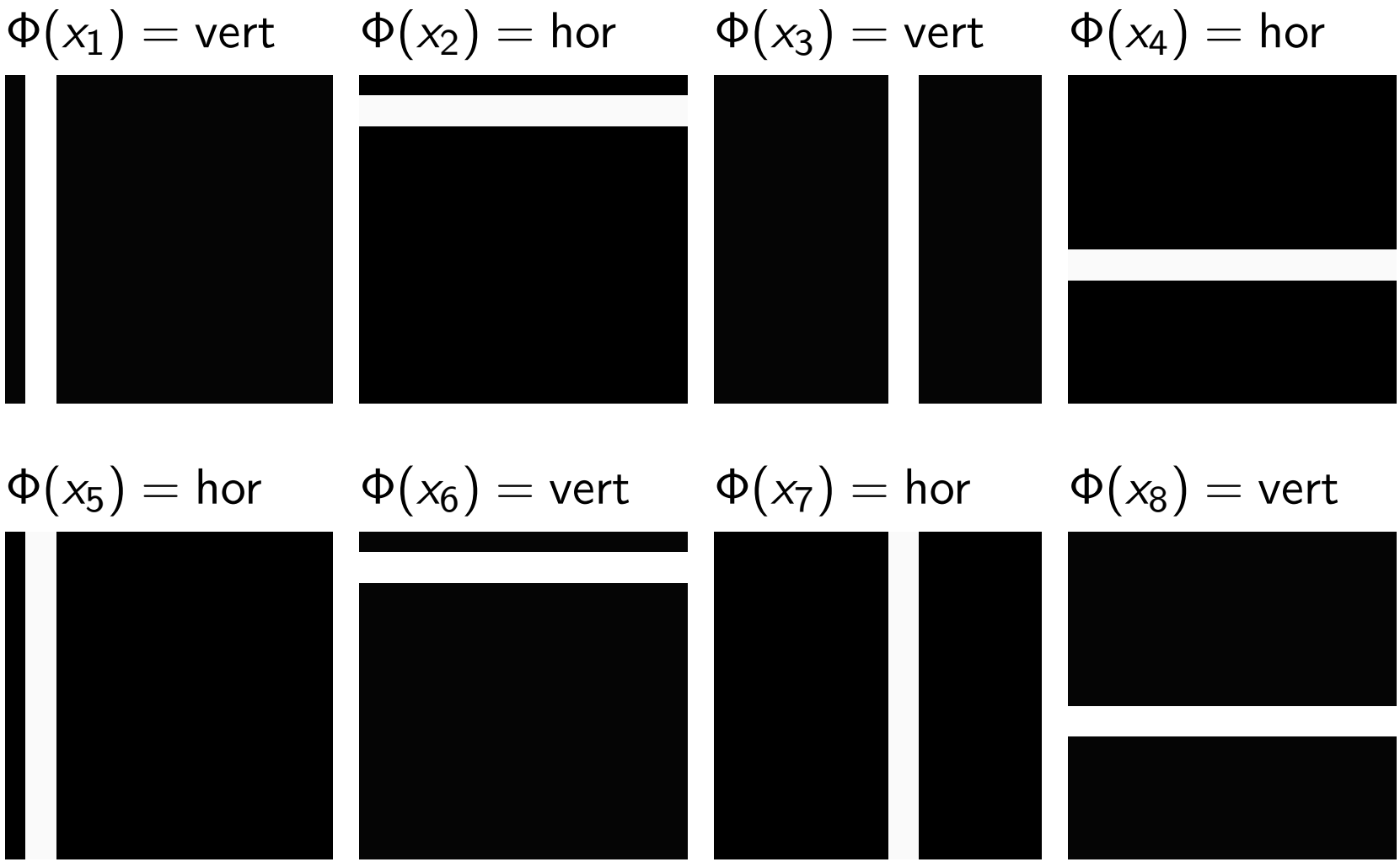
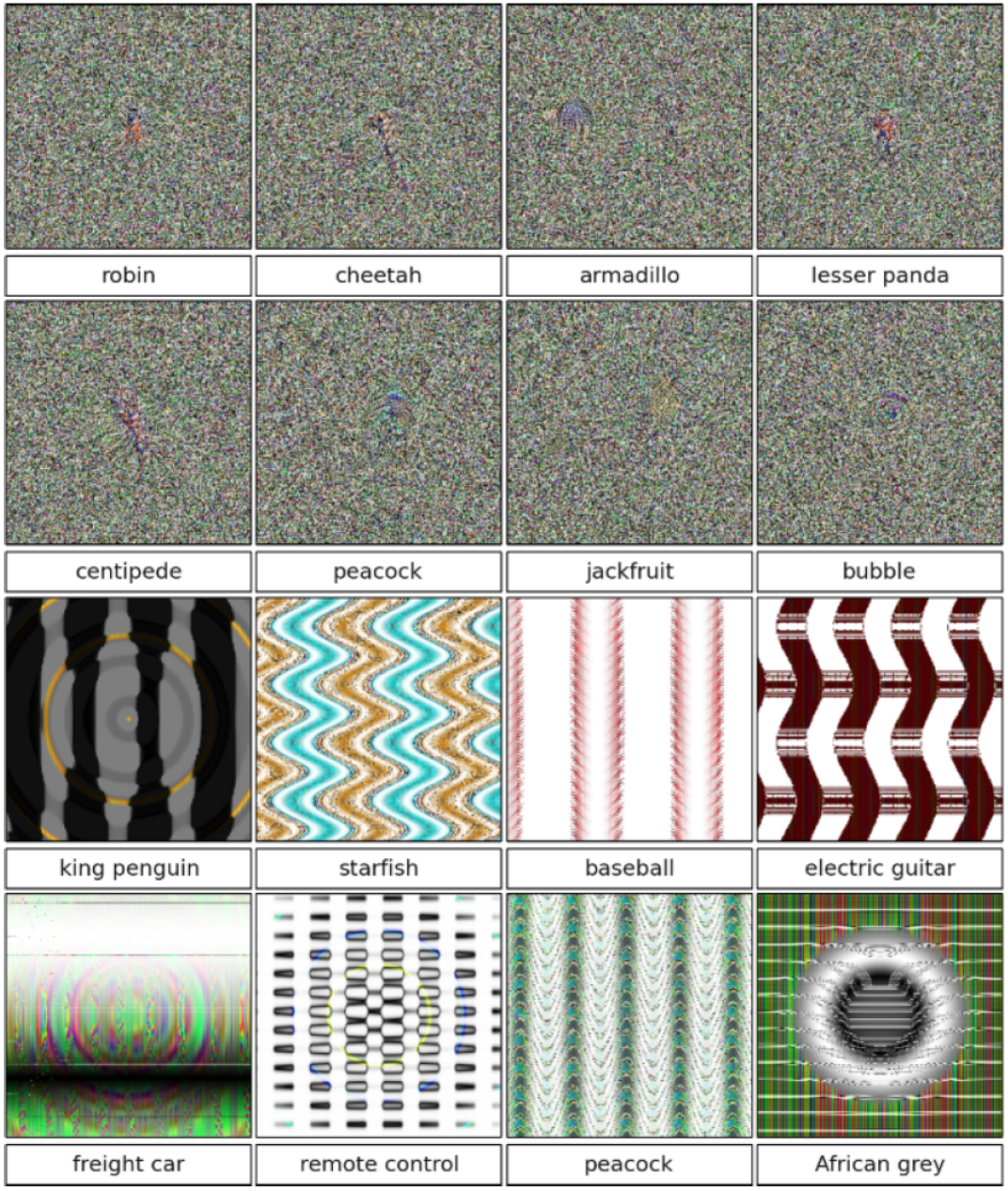


Figure : Upper row: Test of Φ on elements in $\mathcal{C}$. Lower row: Test of Φ on elements in $\tilde{\mathcal{C}}$. We have 100% success rate on $\mathcal{C}$ and 50% failure rate on $\tilde{\mathcal{C}}$.

What was the false structure that was learned?



AI replacing standard algorithms

-

Example: Inverse problems

The basic inverse problem

Image $x \in \mathbb{C}^N$, we are given access to measurements of the form

$$y = Ax + e, \tag{7}$$

where $A \in \mathbb{C}^{m \times N}$ represents sampling modality, $m \ll N$.

Task is to reconstruct x from the noisy measurements y.

Given a crappy images $x \in \mathbb{R}^d$, train a neural network $\phi \in \mathcal{NN}_{\mathbf{N},L,d}$ to get a good images

$$y = \phi(x).$$

In practice, one tries to learn the noise and use

$$y = x - \phi(x).$$

Denoising experiment with deep learning

Original

Noisy version

Denoised with Neur. Net.

DL in Inverse Problems: 1st Step

```
>> I = phantom(512); theta_1 = [0:1:179];
>> R = radon(I, theta_1);
>> imshow(I); imagesc(R)
```

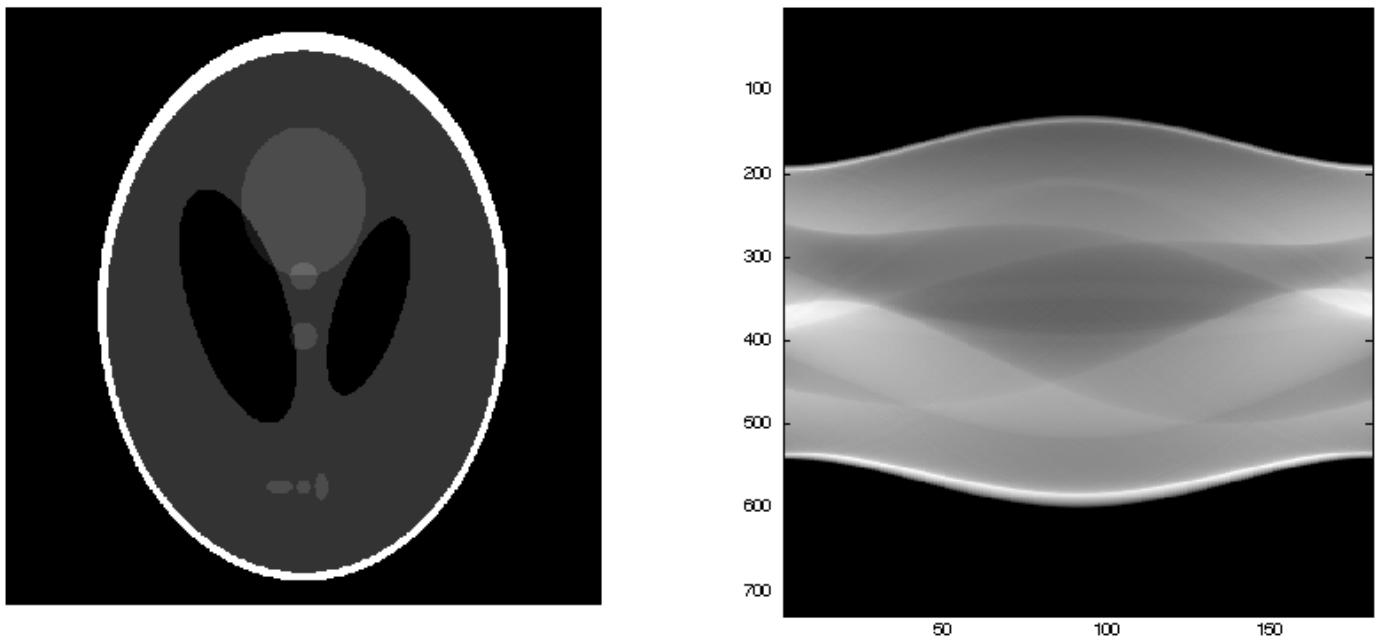


Figure : Left: Logan-Shepp Phantom. Right: The image under the Radon transform (sinogram)

DL in Inverse Problems: 1st Step

```
>> I = phantom(512); theta_3 = [0:3:179];
>> R = radon(I, theta_3); II = iradon(R,theta_3);
>> imshow(I); imagesc(II)
```

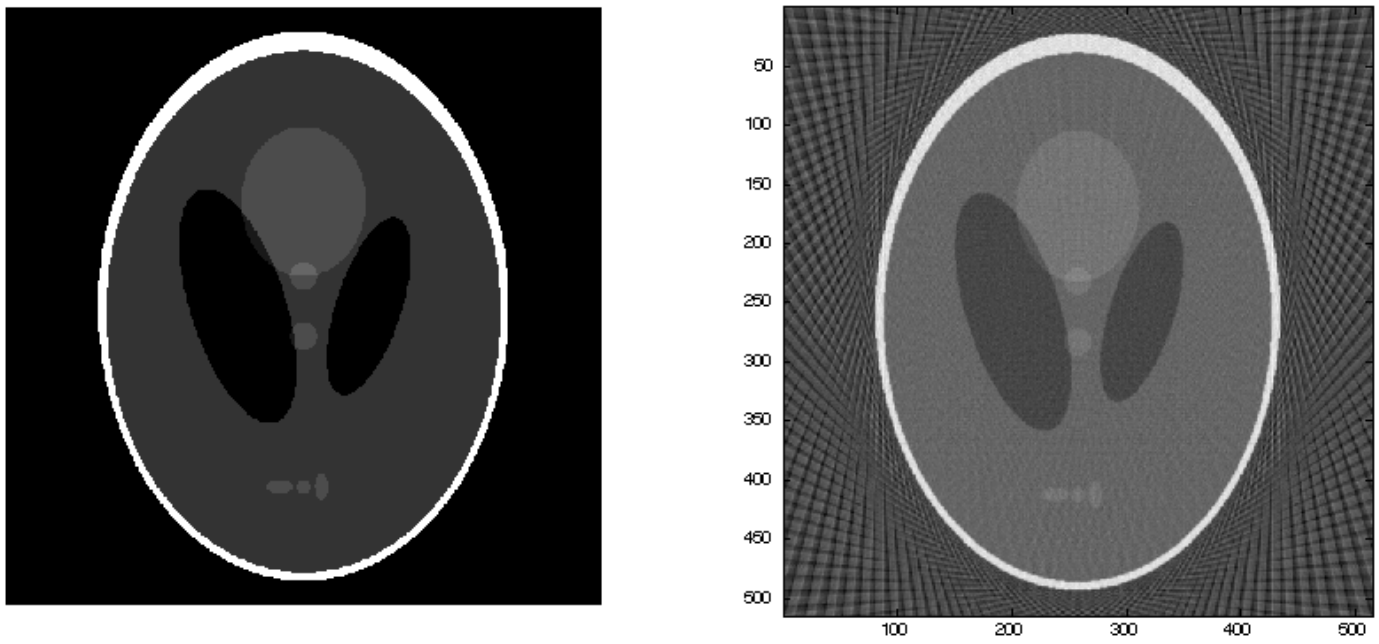


Figure : Left: Logan-Shepp Phantom. Right: Reconstruction with the filtered back projection using 60 lines.

DL in Inverse Problems: 1st Step

Crazy idea: The filtered back projection gives a noisy image.

Why don't we try deep learning to denoise the image. In particular, we train a neural network ϕ such that

$$x \approx \text{iradon}(\text{radon}(x)) - \phi(\text{iradon}(\text{radon}(x))$$

Deep Convolutional Neural Network for Inverse Problems in Imaging

Kyong Hwan Jin, Michael T. McCann, *Member, IEEE*, Emmanuel Froustey, Michael Unser, *Fellow, IEEE*

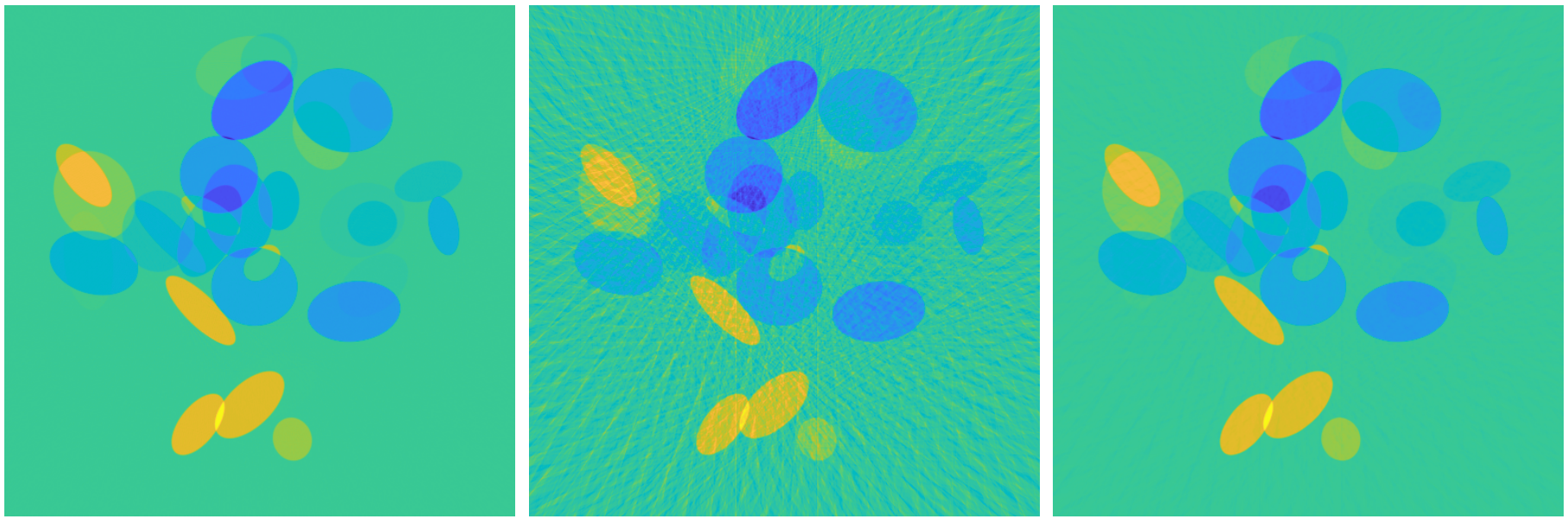
Abstract

In this paper, we propose a novel deep convolutional neural network (CNN)-based algorithm for solving ill-posed inverse problems. Regularized iterative algorithms have emerged as the standard approach to ill-posed inverse problems in the past few decades. These methods produce excellent results, but can be challenging to deploy in practice due to factors including the high computational cost of the forward and adjoint operators and the difficulty of hyper parameter selection. The starting point of our work is the observation that unrolled iterative methods have the form of a CNN (filtering followed by point-wise non-linearity) when the normal operator (H^*H, the adjoint of H times H) of the forward model is a convolution. Based on this observation, we propose using direct inversion followed by a CNN to solve normal-convolutional inverse problems. The direct inversion encapsulates the physical model of the system, but leads to artifacts when the problem is ill-posed; the CNN combines multiresolution decomposition and residual learning in order to learn to remove these artifacts while preserving image structure. We demonstrate the performance of the proposed network in sparse-view reconstruction (down to 50 views) on parallel beam X-ray computed tomography in synthetic phantoms as well as in real experimental sinograms. The proposed network outperforms total variation-regularized iterative reconstruction for the more realistic phantoms and requires less than a second to reconstruct a 512×512 image on the GPU.

DL in Inverse Problems: 1st Step (Experiments)

Computerised Tomography (CT) experiment with deep learning

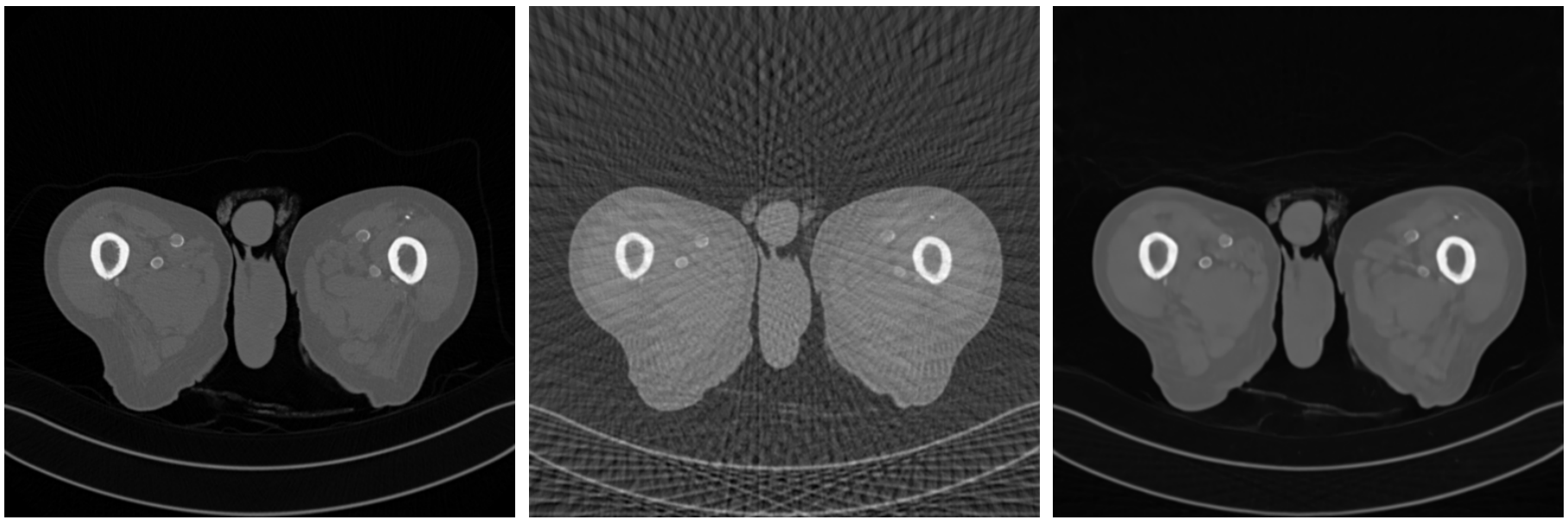
Original | Recon-FBP (50 lines) | Recon-NeurNet (50 lines)



DL in Inverse Problems: 1st Step (Experiments)

Computerised Tomography (CT) experiment with deep learning

Original | Recon-FBP (50 lines) | Recon-NeurNet (50 lines)



The Instability Test

Given a neural network $\Phi : \mathbb{R}^m \to \mathbb{R}^N$ that is able to reconstruct images from, for example, MRI (Fourier) data

$$y_{\text{data}} = A_{\text{scan}} x_{\text{image}}, \qquad A_{\text{scan}} \in \mathbb{C}^{m \times N}$$

$$\Phi(y_{\text{data}}) = x_{\text{image}},$$

find a perturbation x_δ with $\|x_\delta\| \leq \delta$, where $\delta > 0$ is small, such that

$$\Phi(y_{\text{data}} + A x_\delta) = x_{\text{image}} + x_{\text{artefact}},$$

or

$$\Phi(y_{\text{data}} + A x_\delta) = x_{\text{image}} + x_{\text{falsetumor}}.$$

Instability of DL in Inverse Problems - MRI

Experiment from "On instabilities of deep learning in image reconstruction - Does AI come at a cost?", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen

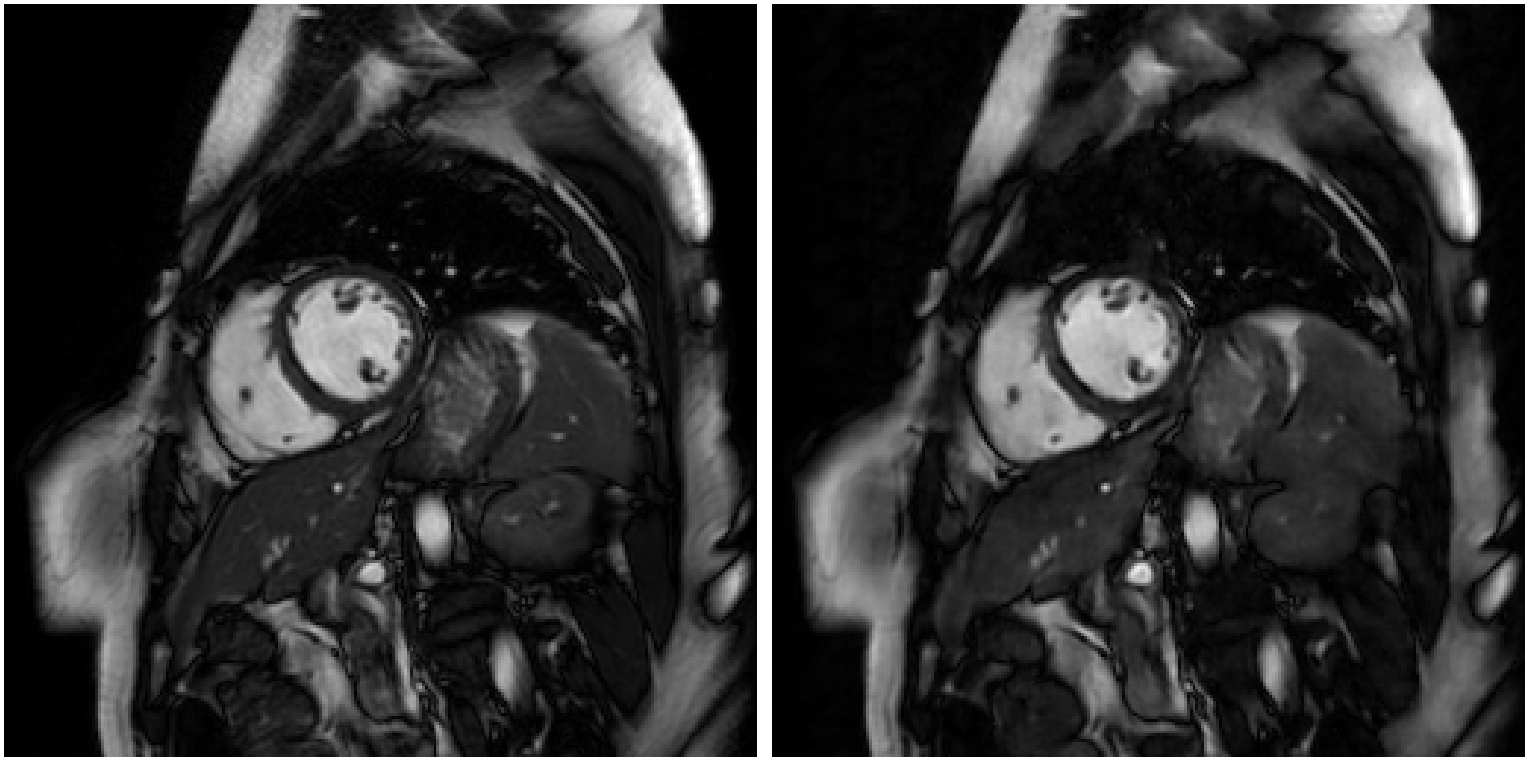


Figure : Left: Original image (no perturbation). Right: Reconstruction (25 % subsampling).

Neural net from "A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction", J. Schlemper, J. Caballero, J. Hajnal, A. Price, D. Rueckert *IEEE Trans. Med. Imag.* (to appear).

Instability of DL in Inverse Problems - MRI

Experiment from "On instabilities of deep learning in image reconstruction - Does AI come at a cost?", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen

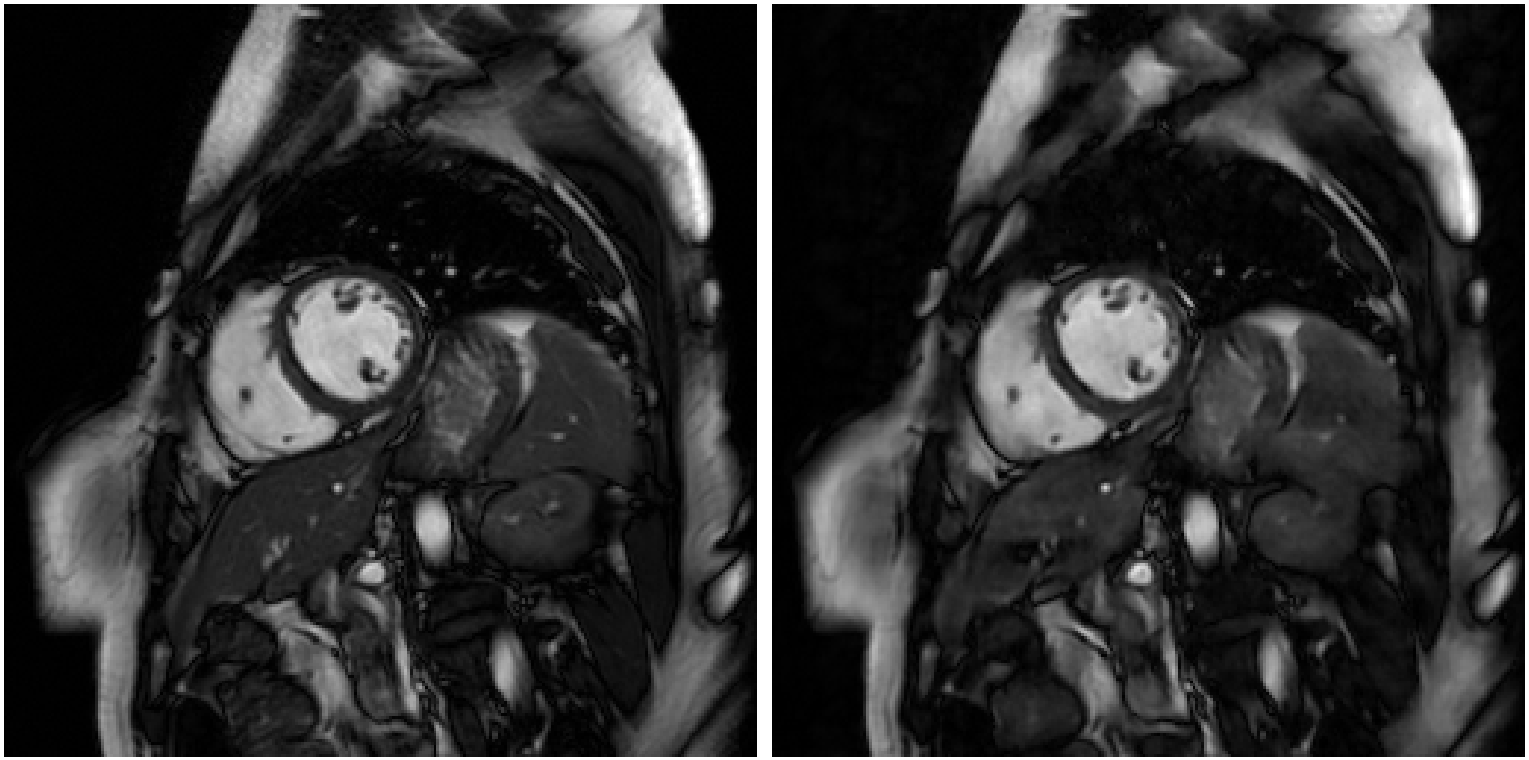


Figure : Left: Original image + tiny perturbation. Right: Reconstruction (25 % subsampling).

Neural net from "A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction", J. Schlemper, J. Caballero, J. Hajnal, A. Price, D. Rueckert *IEEE Trans. Med. Imag.* (to appear).

Instability of DL in Inverse Problems - MRI

Experiment from "On instabilities of deep learning in image reconstruction - Does AI come at a cost?", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen

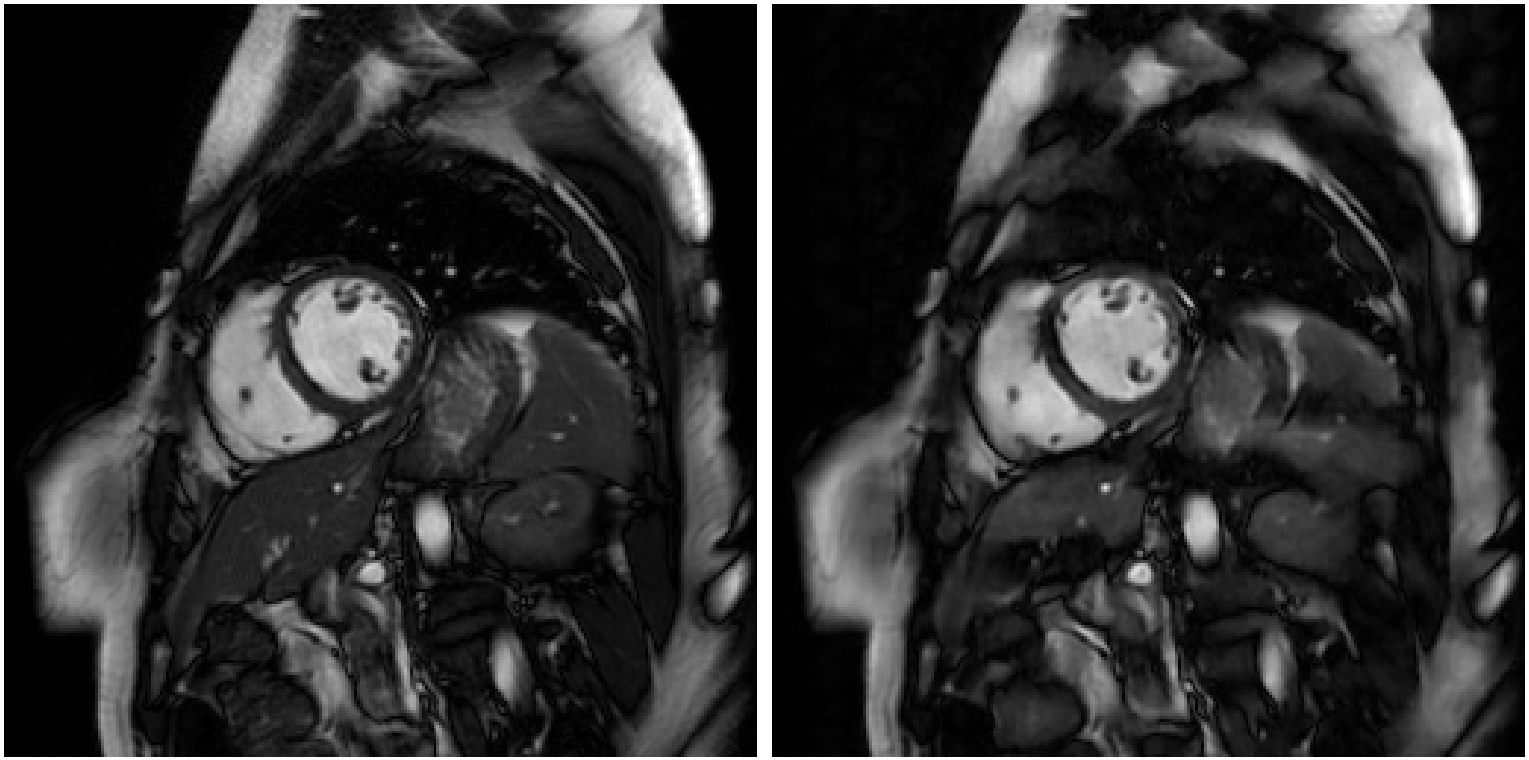


Figure : Left: Original image + tiny perturbation. Right: Reconstruction (25 % subsampling).

Neural net from "A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction", J. Schlemper, J. Caballero, J. Hajnal, A. Price, D. Rueckert *IEEE Trans. Med. Imag.* (to appear).

Instability of DL in Inverse Problems - MRI

Experiment from "On instabilities of deep learning in image reconstruction - Does AI come at a cost?", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen

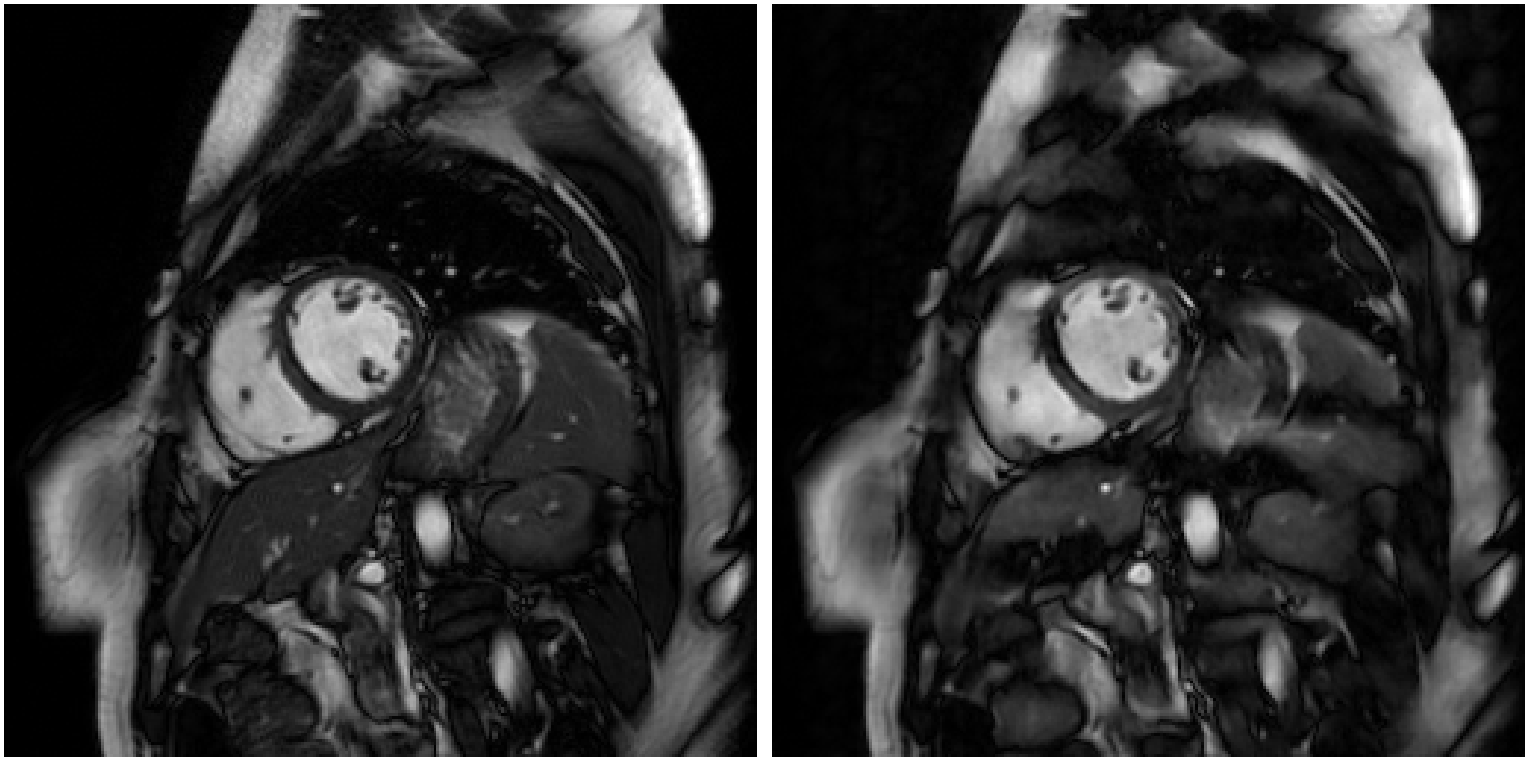


Figure : Left: Original image + tiny perturbation. Right: Reconstruction (25 % subsampling).

Neural net from "A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction", J. Schlemper, J. Caballero, J. Hajnal, A. Price, D. Rueckert *IEEE Trans. Med. Imag.* (to appear).

Instability of DL in Inverse Problems - MRI

Experiment from "On instabilities of deep learning in image reconstruction - Does AI come at a cost?", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen

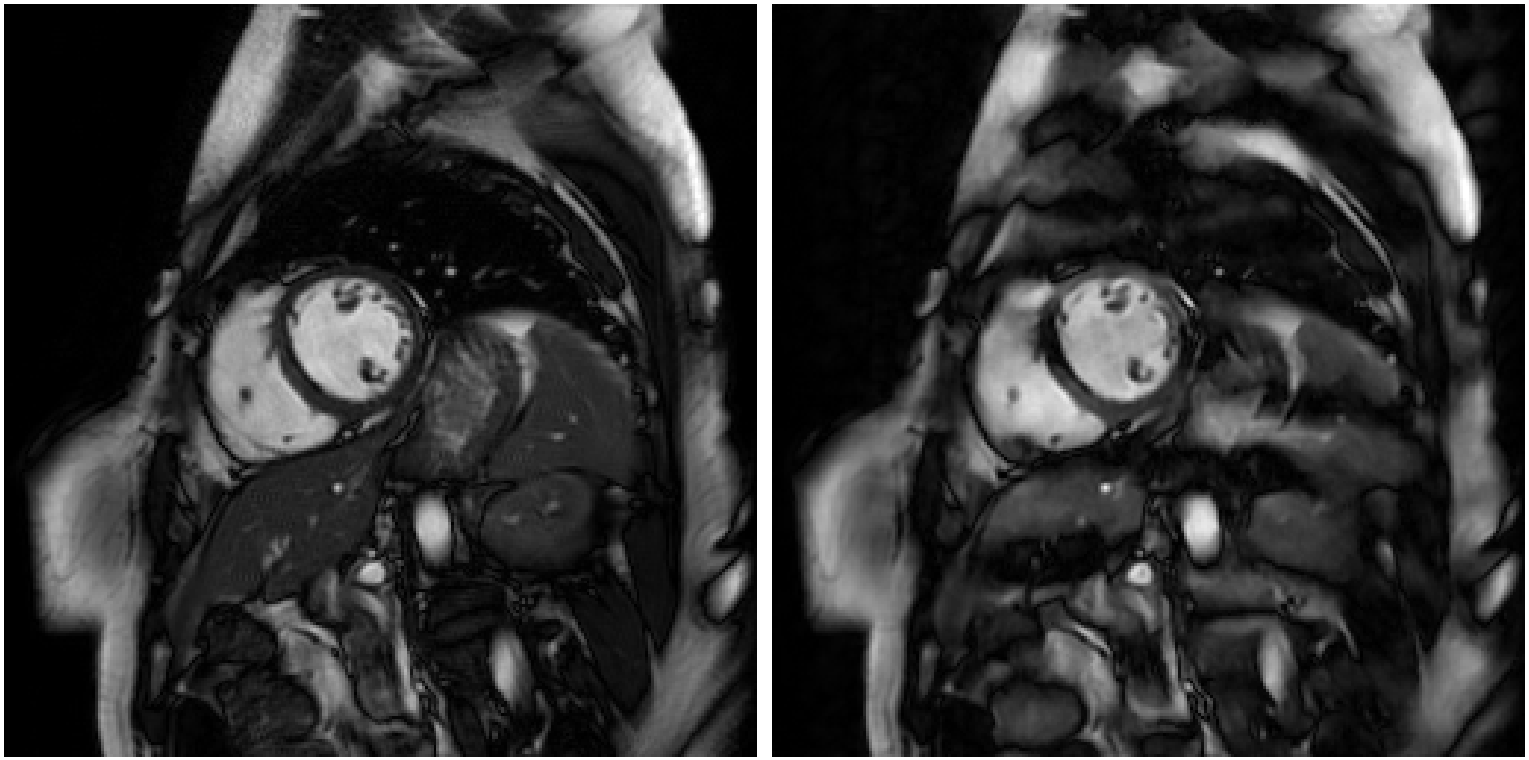


Figure : Left: Original image + tiny perturbation. Right: Reconstruction (25 % subsampling).

Neural net from "A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction", J. Schlemper, J. Caballero, J. Hajnal, A. Price, D. Rueckert *IEEE Trans. Med. Imag.* (to appear).

Instability of DL in Inverse Problems - MRI

Experiment from "On instabilities of deep learning in image reconstruction - Does AI come at a cost?", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen

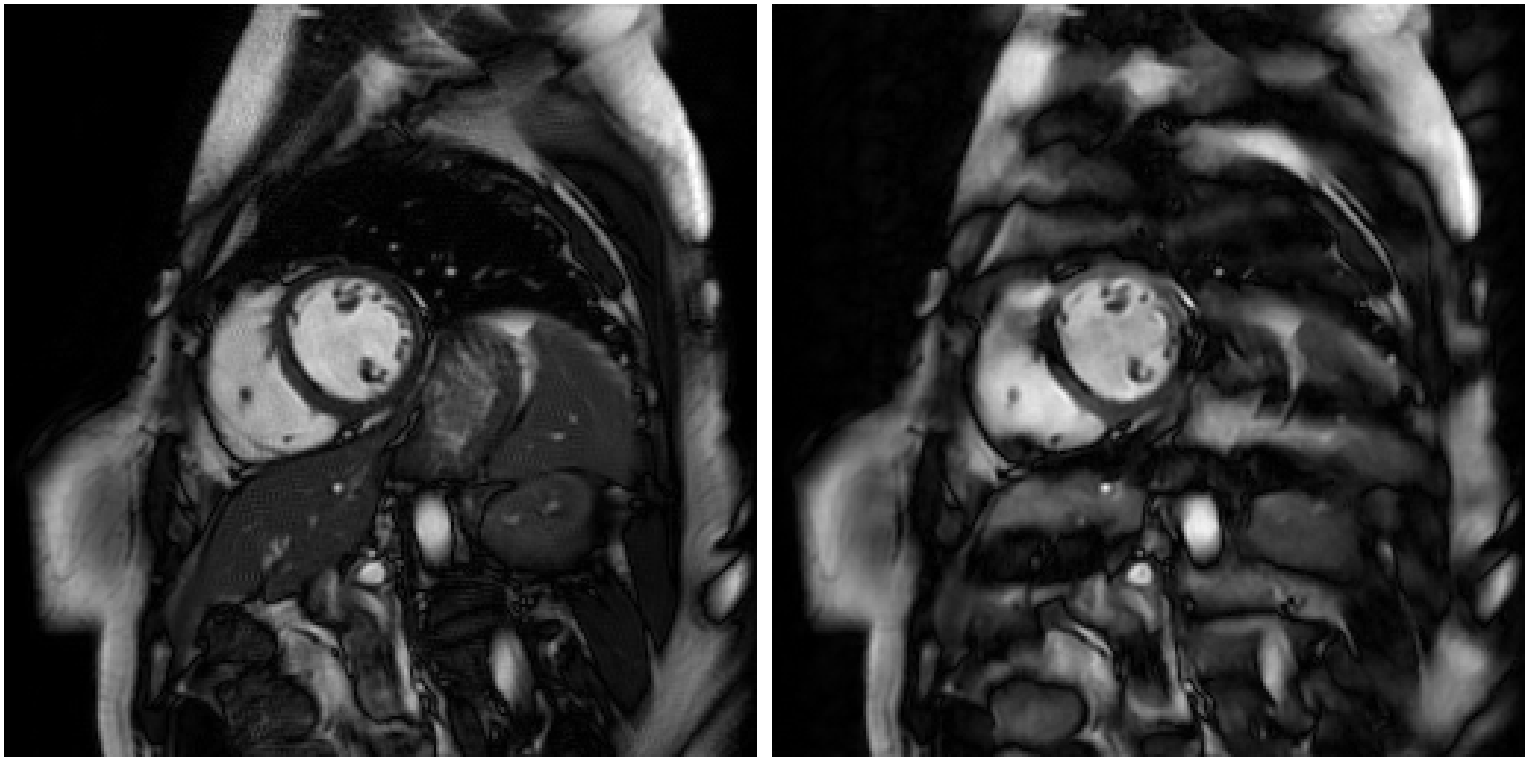


Figure : Left: Original image + tiny perturbation. Right: Reconstruction (25 % subsampling).

Neural net from "A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction", J. Schlemper, J. Caballero, J. Hajnal, A. Price, D. Rueckert *IEEE Trans. Med. Imag.* (to appear).

Instability of DL in Inverse Problems - MRI

Experiment from "On instabilities of deep learning in image reconstruction - Does AI come at a cost?", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen

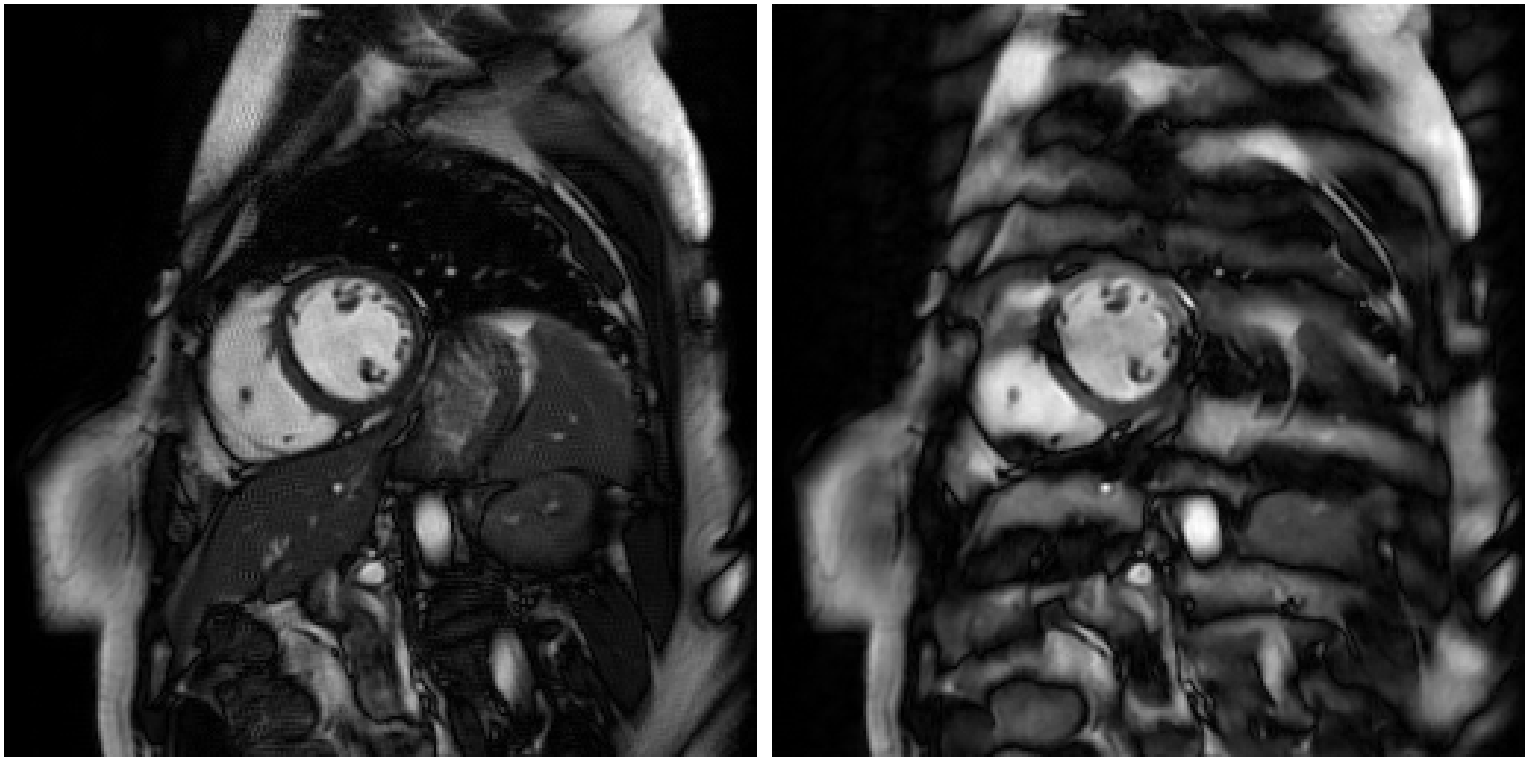


Figure : Left: Original image + tiny perturbation. Right: Reconstruction (25 % subsampling).

Neural net from "A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction", J. Schlemper, J. Caballero, J. Hajnal, A. Price, D. Rueckert *IEEE Trans. Med. Imag.* (to appear).

Instability of DL in Inverse Problems - MRI

Experiment from "On instabilities of deep learning in image reconstruction - Does AI come at a cost?", V. Antun, F. Renna, C. Poon, B. Adcock, A. Hansen

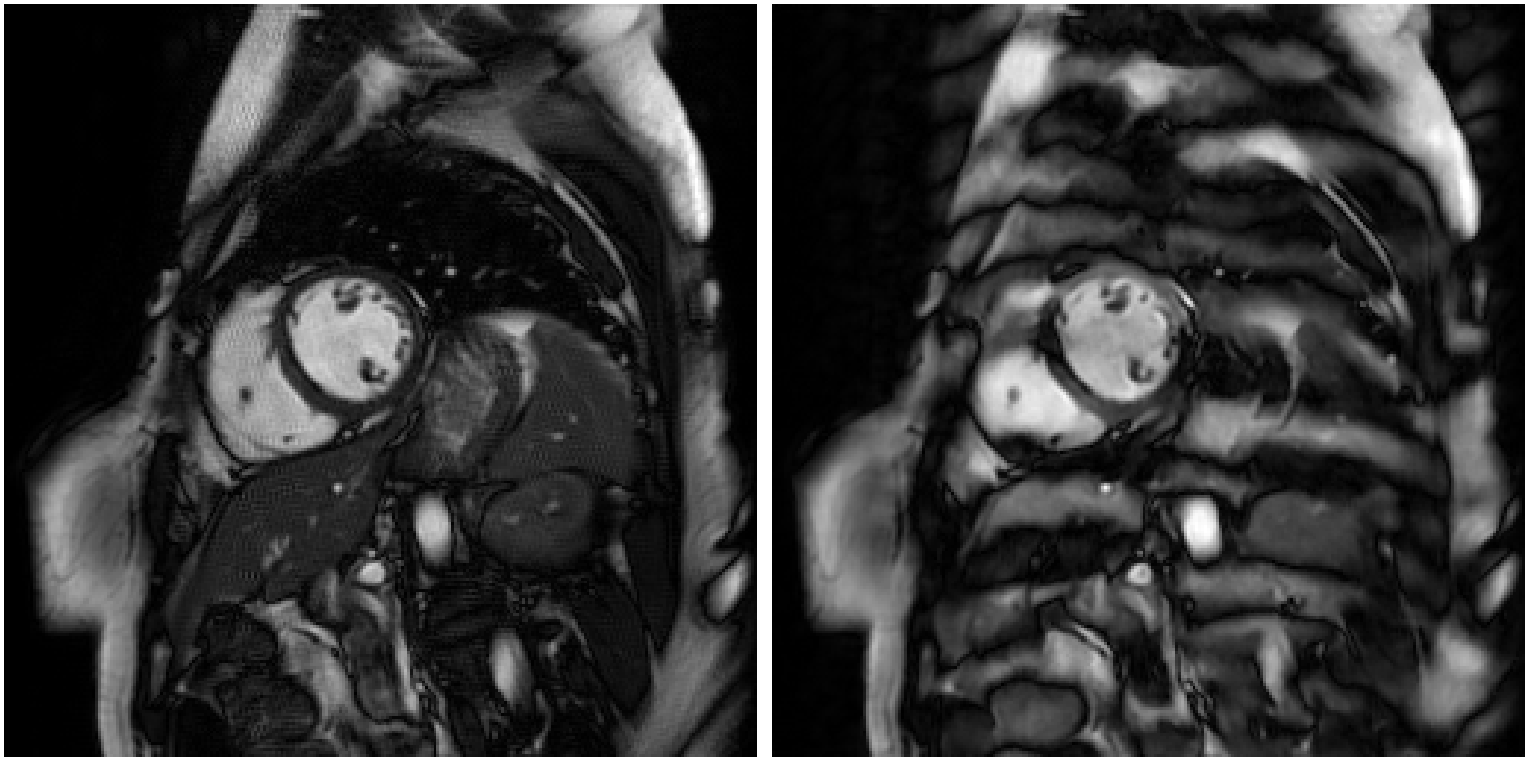


Figure : Left: Original image + tiny perturbation. Right: Reconstruction (25 % subsampling).

Neural net from "A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction", J. Schlemper, J. Caballero, J. Hajnal, A. Price, D. Rueckert *IEEE Trans. Med. Imag.* (to appear).

Instabilities in Inverse Problems

Original

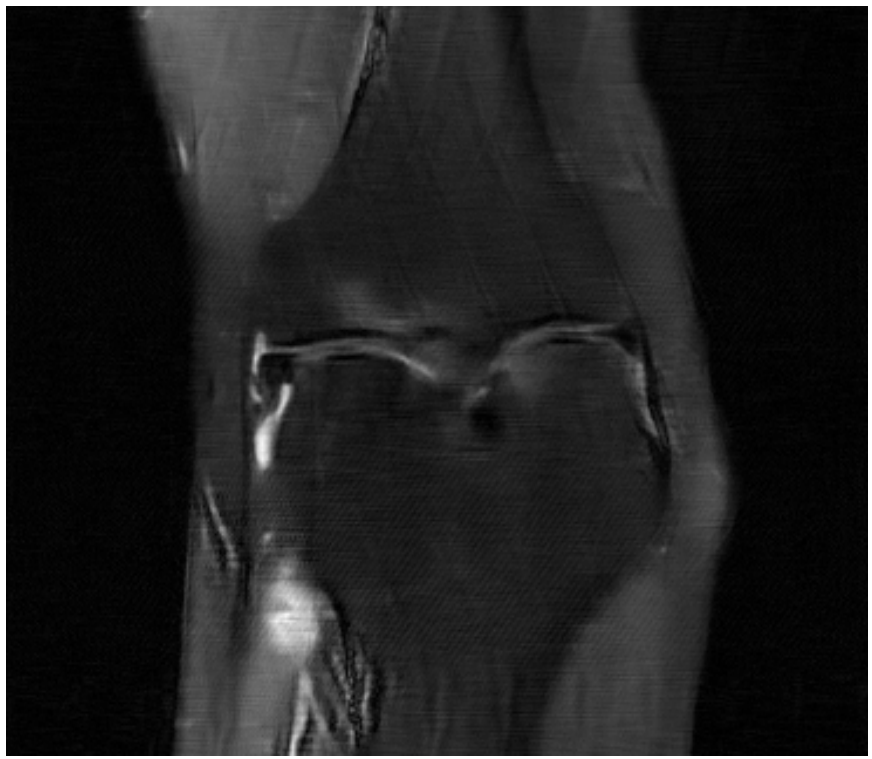
Network (15% sampling)

Neural net from "Learning a variational network for reconstruction of accelerated MRI data", K. Hammernik, T. Klatzer, E. Kobler, M. Recht, D. Sodickson, T. Pock, F. Knoll. *Magn. Reson. Imaging.* (Nov. 2017).

Instabilities in Inverse Problems

Original + tiny pert.

Network (15% sampling)



Neural net from "Learning a variational network for reconstruction of accelerated MRI data", K. Hammernik, T. Klatzer, E. Kobler, M. Recht, D. Sodickson, T. Pock, F. Knoll. *Magn. Reson. Imaging.* (Nov. 2017).

Theorem 8 (Antun, Gottschling, Hansen)

Let $A : \mathbb{C}^N \to \mathbb{C}^m$ be a linear sampling map and let $\Psi : \mathbb{C}^m \to \mathbb{C}^N$. Suppose that there are $\eta, x, y, \xi_x, \xi_y \in \mathbb{C}^N$ with $\|\xi_x\|, \|\xi_x\| \leq \delta \in (0,1)$ such that

$$\Psi(A(\eta + x)) = \eta + x + \xi_x, \quad \Psi(A(\eta + y)) = \eta + y + \xi_y,$$

where $x \in \mathcal{N}(A)$, the nullspace of A, and $\|Ay\| = \delta > 0$. Then we have the following.

(i) *(Instabilities) If $\|x\| = \|y\| = 1$ and $\Re\langle x, y\rangle \in [-1, 1)$ then the local Lipschitz constant of Ψ at $A\eta$, defined for $\epsilon \geq \delta > 0$, satisfies*

$$L_\eta^\epsilon = \sup_{0<\|Az\|\leq\epsilon} \frac{\|\Psi(A\eta + Az) - \Psi(A\eta)\|}{\|Az\|} \geq \frac{1}{\epsilon}\left(\sqrt{2(1 - \Re\langle x, y\rangle)} - 2\delta\right).$$

(ii) *(False positives) Moreover, there exists a perturbation $r \in \mathbb{C}^m$ with $\|r\| = \delta$ such that*

$$\|\Psi(r + A(\eta + y)) - (\eta + x)\| \leq \delta.$$

(iii) *(False negatives) If $x = 0$ above, then there exists a perturbation $r \in \mathbb{C}^m$ with $\|r\| = \delta$ such that*

$$\|\Psi(r + A(\eta + y)) - \eta\| \leq \delta.$$

Kernel awareness

Theorem 8 says that to avoid instability there must be some "kernel awareness" in the algorithm.

Training neural nets for inverse problems

Given an inverse problem

$$Ax = y, \quad A \in \mathbb{C}^{m\times N}, \quad m < N$$

a training set $\mathcal{T} = \{y^1 = Ax^1, \ldots, y^r = Ax^r\} \subset \mathbb{R}^d$, and a cost function $C : \mathbb{R}^{dr} \times \mathbb{R}^{dr} \to \mathbb{R}_+$, compute

$$\phi \in \underset{\tilde{\phi}\in\mathcal{NN}_{\mathbf{N},L,d}}{\operatorname{argmin}} \; C(v, w),$$

with

$$v = \{\tilde{\phi}(y^j)\}_{j=1}^r, \quad w = \{x^j\}_{j=1}^r.$$

Definition 9 (Robust Nullspace Property)

A matrix $U \in \mathbb{C}^{m \times n}$ satisfies the ℓ^2 *robust nullspace property of order* s if there is a $\rho \in (0, 1)$ and a $\tau > 0$ such that

$$\|v_S\|_2 \leq \frac{\rho}{\sqrt{s}} \|v_{S^c}\|_1 + \tau \|Uv\|_2 \tag{8}$$

for all s-sparse sets S and vectors $v \in \mathbb{C}^n$.

Theorem 10 (Robust Nullspace Property)

Suppose that a matrix $A \in \mathbb{C}^{m \times N}$ *satisfies the robust null space property of order* s *with constants* $0 < \rho < 1$ *and* $\tau > 0$. *Then, for any* s*-sparse* $x \in \mathbb{C}^N$ *and* $y = Ax$, *a solution*

$$\tilde{x} \in \arg\min ||z||_1 \text{ subject to } ||Az - y||_2 \leq \delta$$

satisfies

$$\|x - \tilde{x}\|_1 \leq \frac{4\tau}{1-\rho}\delta.$$

Can we always compute a neural network that approximates the function we want?

Approximation qualities of neural nets

The universal approximation theorem:

Theorem 11 (Pinkus, Acta Numerica 1999)

Let $\rho \in C(\mathbb{R})$. Then the set of neural networks is dense in $C(\mathbb{R}^d)$ in the topology of uniform convergence on compact sets, if and only if ρ is not a polynomial.

The interpolation theorem:

Theorem 12 (Pinkus, Acta Numerica 1999)

Let $\rho \in C(\mathbb{R})$ and assume that ρ is not a polynomial. For any k distinct points $\{x_j\}_{j=1}^k \subset \mathbb{R}^d$ and associated data $\{\alpha_j\}_{j=1}^k \subset \mathbb{R}$. Then there exists a neural network ϕ such that

$$\phi(x_j) = \alpha_j, \qquad j = 1, \ldots, k.$$

Theorem 13 (Bastounis, Hansen, Vlacic)

There is an uncountable family of classification functions $f : \mathbb{R}^{N_0} \to \{0,1\}$ *such that for any neural network dimensions* $\mathbf{N} = (N_L, N_{L-1}, \ldots, N_1, N_0)$ *with* $N_0, L \geq 2$ *and any* $0 < \epsilon < 1/(K+M)$ *where* M *is arbitrarily large and* $K \geq 3(N_1+1)\cdots(N_{L-1}+1)$ *we have the following. There exist uncountably many training sets* $\mathcal{T} = \{x^1, \ldots, x^K\}$ *and uncountably many classification sets* $\mathcal{C} = \{y^1, \ldots, y^M\}$ *such that there is a*

$$\tilde{\Phi} \in \operatorname*{argmin}_{\Phi \in \mathcal{NN}_{\mathbf{N},L}} C(v,w), \quad v_j = \Phi(x^j), \quad w_j = f(x^j),$$

where $1 \leq j \leq K$, *and* $C(v,w) = 0$ *iff* $v = w$, *such that*

$$\tilde{\Phi}(x) = f(x) \quad \forall x \in \mathcal{T} \cup \mathcal{C}.$$

However, there exists uncountably many $v \in \mathbb{R}^{N_0}$ *such that*

$$|\tilde{\Phi}(v) - f(v)| \geq 1/2, \qquad \|v - x\|_\infty \leq \epsilon \text{ for some } x \in \mathcal{T}.$$

Moreover, there is another neural network $\hat{\Phi}$ *such that*

$$\hat{\Phi}(x) = f(x) \quad \forall x \in \mathcal{B}_\epsilon^\infty(\mathcal{T} \cup \mathcal{C}).$$

Computing neural nets: What could go wrong?

(i) There does not exist a neural network that approximates the function we are interested in.

(ii) There does exist a neural network that approximates the function, however, there does not exist an algorithm that can construct the neural network.

(iii) There does exist a neural network that approximates the function, and an algorithm to construct it. However, the algorithm will need prohibitively many samples.

Both of these last two can happen!

Classical mappings in inverse problems

(iii) Basis Pursuit (BP)

$$z \in \operatorname*{argmin}_x \|x\|_1 \text{ subject to } \|Ax - y\|_2 \leq \delta, \qquad \delta \geq 0, \quad (9)$$

(iv) Unconstrained Lasso (UL)

$$z \in \operatorname*{argmin}_x \|Ax - y\|_2^2 + \lambda \|x\|_1, \qquad \lambda > 0, \quad (10)$$

Neural networks are FANTASTIC approximators!

Consider the following mapping $\varphi_{A,\nu} : \mathcal{M} \to \mathbb{R}^N$ where

$$\mathcal{M} = \{y_j\}_{j=1}^r \subset \mathbb{R}^m, \quad r < \infty,\ m < N \tag{11}$$

given by

$$\varphi_{A,\nu}(y) = w, \quad w \in \operatorname*{argmin}_z \|z\|_1 \text{ subject to } \|Az - y\|_2 \leq \nu. \tag{12}$$

Theorem 14

Let $\nu, \delta \geq 0$. If the non-linear function ρ in each layer is not a polynomial, there exists a neural network Φ, depending on A and $\mathcal{M}$, such that

$$\|\Phi(y) - \varphi_{A,\nu}(y)\|_2 \leq \delta, \quad \forall y \in \mathcal{M}.$$

But: need a constructive training model.

Constructive?

In reality given approximations: $\{y_{j,n}\}_{j=1}^{r}$, $\{\phi_{j,n}\}_{j=1}^{r}$ and A_n such that:

$$\|y_{j,n} - y_j\|, \ \|\phi_{j,n} - \varphi_{A_n,\nu}(y_{j,n})\|, \ \|A_n - A\| \leq 2^{-n}.$$

This is what we can store on a computer in real life, models irrational A etc.

Training set must be

$$\mathcal{T} := \{(y_{j,n}, \phi_{j,n}, A_n) \,|\, j = 1, \ldots, r, n \in \mathbb{N}\}.$$

Can we train a neural network that can approximate Φ based on the training set $\mathcal{T}$?

Theorem 15 (Impossible in general)

Let $K > 2, L \in \mathbb{N}$ and d be any metric on $\mathbb{R}^N$ where $N \geq 6$. Then there exists a well conditioned class Ω of elements $(A, \mathcal{M})$, such that we have the following three conditions. Consider the neural network Φ from Theorem 1.

(i) *There does not exist any algorithm taking elements from $\mathcal{T}$ as input and producing a neural network Ψ such that Ψ approximates Φ on $\mathcal{M}$ to K correct digits in the metric d for all $(A, \mathcal{M}) \in \Omega$.*

(ii) *There exists an algorithm taking elements from $\mathcal{T}$ as input that produces a neural network Ψ that approximates Φ on $\mathcal{M}$ to $K - 1$ correct digits in the metric d for all $(A, \mathcal{M}) \in \Omega$. However, any algorithm producing such a network will need arbitrary many samples of elements from $\mathcal{T}$, where accessing $(y_{j,n}, \phi_{j,n}, A_n)$ for one j and n counts as one sample.*

(iii) *There exists an algorithm using L samples from $\mathcal{T}$ as input that produces a neural network Ψ that approximates Φ on $\mathcal{M}$ to $K - 2$ correct digits in the metric d for all $(A, \mathcal{M}) \in \Omega$.*

Well conditioned

- Condition of a matrix $\mathrm{Cond}(A) = \|A\|\|A^{-1}\|$.
- Condition of the mapping $\Psi : \Omega \subset \mathbb{C}^n \to \mathbb{C}^m$, linear or non-linear, is often given by

$$\mathrm{Cond}(\Psi) = \sup_{x\in\Omega} \lim_{\epsilon\to 0^+} \sup_{\substack{x+z\in\Omega \\ 0<\|z\|\leq\epsilon}} \frac{\mathrm{dist}(\Psi(x+z), \Psi(x))}{\|z\|},$$

where we allow for multivalued functions by defining $\mathrm{dist}(\Psi(x), \Psi(z)) = \min_{\tilde{x}\in\Psi(x),\tilde{z}\in\Psi(z)} \|\tilde{x} - \tilde{z}\|$.

Well conditioned

- If Ψ denotes the solution map to our problem (in this example basis pursuit) with domain Ω, we define

$$\rho(A, y) = \sup\{\delta \mid \|\tilde{A}\|, \|\tilde{y}\| \leq \delta \Rightarrow (A + \tilde{A}, y + \tilde{y}) \in \Omega \text{ are feasible}\},$$

and this yields the Feasibility Primal (FP) condition number

$$C_{\mathrm{FP}}(A, y) := \frac{\max(\|A\|, \|y\|)}{\rho(A, y)}.$$

Do there exist stable neural networks, for inverse problems, that are recursive?

(that also come with recovery guarantees)

Adaptive Neural Networks

An adaptive neural network is a network for which the weights may depend on the input.

In particular, given $y \in \mathbb{C}^N$ and $\mathbf{N} = (N_L, N_{L-1}, \ldots, N_1, N)$ there is a neural network $\Phi_y \in \mathcal{NN}_{\mathbf{N},L}$, where the weights depend on y.

The output of the neural network is given by $\Phi_y(y)$.

In addition the mapping $y \mapsto \Phi_y$ is recursive.

Definition 16 (Sparsity in levels)

For $1 \leq r \leq N$, let $\mathbf{M} = (M_1, \ldots, M_r)$, where $1 \leq M_1 < \ldots < M_r = N$, and $\mathbf{s} = (s_1, \ldots, s_r)$, where $s_k \leq M_k - M_{k-1}$ for $k = 1, \ldots, r$ and $M_0 = 0$. A vector $x \in \mathbb{C}^N$ is $(\mathbf{s}, \mathbf{M})$-*sparse in levels* if

$$|\mathrm{supp}(x) \cap \{M_{k-1} + 1, \ldots, M_k\}| \leq s_k, \quad k = 1, \ldots, r.$$

The *total sparsity* of x is the quantity $s = s_1 + \ldots + s_r$ and the set of $(\mathbf{s}, \mathbf{M})$-sparse vectors is denoted by $\Sigma_{\mathbf{s},\mathbf{M}}$.

Define also

$$\sigma_{\mathbf{s},\mathbf{M}}(x)_{\ell^p} = \min\{\|x - z\|_{\ell^p} : z \in \Sigma_{\mathbf{s},\mathbf{M}}\}$$

Definition 17

Let $r \in \mathbb{N}$, $\mathbf{N} = (N_1, \dots, N_r) \in \mathbb{N}^r$ with $1 \leq N_1 < \dots < N_r$, $\mathbf{m} = (m_1, \dots, m_r) \in \mathbb{N}^r$, with $m_k \leq N_k - N_{k-1}$, $k = 1, \dots, r$, and suppose that

$$\Omega_k \subseteq \{N_{k-1} + 1, \dots, N_k\}, \quad |\Omega_k| = m_k, \quad k = 1, \dots, r,$$

are chosen uniformly at random, where $N_0 = 0$. We refer to the set

$$\Omega = \Omega_{\mathbf{N},\mathbf{m}} := \Omega_1 \cup \dots \cup \Omega_r.$$

as an $(\mathbf{N}, \mathbf{m})$-multilevel sampling scheme.

Stable and Recursive Adaptive NNs Exist

Let $N = 2^j$ for some $j \in \mathbb{N}$ and $\delta > 0$. Then there exist $\mathbf{N} = (N_L, N_{L-1}, \dots, N_1, N)$, where $\mathbf{N}$ and L depend on δ, and a recursive mapping

$$\mathbb{Q}^N \ni z \mapsto \Phi_z \in \mathcal{NN}_{\mathbf{N}}$$

such that we have the following. Let $\mathbf{M} = (M_1, \dots, M_r)$ and $\mathbf{s} = (s_1, \dots, s_r)$ be sparsity levels and local sparsities respectively, where the M_js are the sparsity levels corresponding to the scales of the Haar wavelet. Let $\epsilon > 0$, $\mathbf{N} = \mathbf{M}$ and let $\Omega = \Omega_{\mathbf{N},\mathbf{m}}$ be the $(\mathbf{N}, \mathbf{m})$-multilevel sampling pattern with

$$m_k \gtrsim \left(s_k + \sum_{l=1}^{k-1} s_l 2^{-(k-l)} + \sum_{l=k+1}^{r} s_l 2^{-3(l-k)} \right) \cdot L, \quad k = 1, \dots, r,$$

where

$$L = \log^3(N) \cdot \log(m) \cdot \log^2(\log(N)s) + \log(N) \cdot \log(\epsilon^{-1}).$$

If $z = P_\Omega y$, for any y given by

$$y = U_{\mathrm{df}} x + e, \qquad \|e\|_2 \leq \delta,$$

where $U_{\mathrm{df}} \in \mathbb{C}^{N \times N}$ denotes the discrete Fourier transform and $x, e \in \mathbb{C}^N$, then, with probability greater than $1 - \epsilon$,

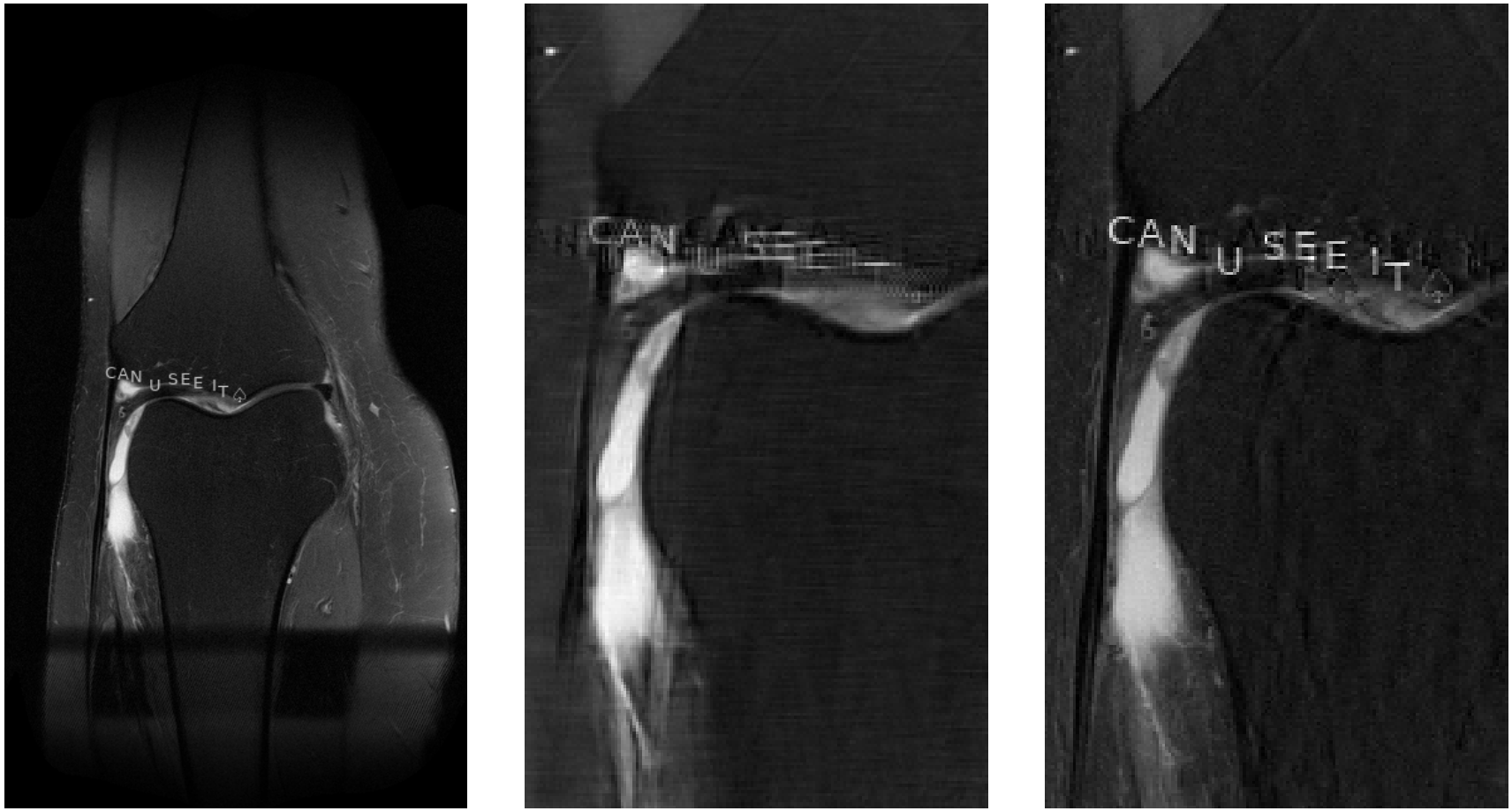
$$\|x - \Phi_z(z)\|_{\ell^2} \lesssim r^{1/4} \left(\sigma_{\mathbf{s},\mathbf{M}}(Wx)_{\ell^1} / \sqrt{rs} + \delta \right),$$

where $W \in \mathbb{C}^{N \times N}$ is the discrete Haar transform.

Deep learning vs Untrained Adaptive NNs

MRI experiment with 15% subsampling.

Original **Trained Network** **Untrained Network**



Trained neural net from "Learning a variational network for reconstruction of accelerated MRI data", K. Hammernik, T. Klatzer, E. Kobler, M. Recht, D. Sodickson, T. Pock, F. Knoll. *Magn. Reson. Imaging.* (Nov. 2017).

How should we test AI?

Smale's 18th problem

What are the limits of intelligence, both artificial and human?

Smale's 18th problem

Smale:

"Penrose (1991) attempts to show some limitations of artificial intelligence. Involved in his argumentation is the interesting question, is the Mandelbrot set decidable? (see problem 14) and implications of the Gödel incompleteness theorem. However a broader study is called for, one which involves deeper models of the brain, and of the computer, in a search of what artificial and human intelligence have in common, and how they differ."

"Finally problem solving as exemplified by Turing and real number machines is only part of the story of intelligence. Continual interaction with the environment must be incorporated into a good model. Learning is a part of human intelligent activity. The corresponding mathematics is suggested by the theory of repeated games, neural nets and genetic algorithms."

Turing's imitation game

Turing:

"I propose to consider the question, "Can machines think?" This should begin with definitions of the meaning of the terms "machine" and "think." The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous, If the meaning of the words "machine" and "think" are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, "Can machines think?" is to be sought in a statistical survey such as a Gallup poll. But this is absurd. Instead of attempting such a definition I shall replace the question by another, which is closely related to it and is expressed in relatively unambiguous words."

Turing's imitation game

Turing:

"The new form of the problem can be described in terms of a game which we call the 'imitation game." It is played with three people, a man (A), a woman (B), and an interrogator (C) who may be of either sex. The interrogator stays in a room apart front the other two. The object of the game for the interrogator is to determine which of the other two is the man and which is the woman. He knows them by labels X and Y, and at the end of the game he says either "X is A and Y is B" or "X is B and Y is A." The interrogator is allowed to put questions to A and B."

" We now ask the question, "What will happen when a machine takes the part of A in this game?" Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman? These questions replace our original, "Can machines think?""

The imitation game and deep learning

How would AI, created by deep learning, perform in the Turing test?

Worst case v.s. average performance

What are the appropriate tests for AI?

Tests will depend on application and community:

- Defence and intelligence community
- Healthcare community
- Public sector management community