On the barriers of AI and the trade-off between stability and accuracy in deep learning

Vegard Antun (Oslo, vegarant@math.uio.no)
Matthew J. Colbrook (Cambridge, m.colbrook@damtp.cam.ac.uk)

Joint work with:

Ben Adcock (SFU), Nina Gottschling (Cambridge), Anders Hansen (Cambridge), Clarice Poon (Bath), Francesco Renna (Porto)

Geilo Winter School, January 2021

MAIN GOAL

Determine the barriers of computations in deep learning
(i.e. what is and what is not possible)

$\Downarrow$

Stability and Accuracy in AI

Outline of lectures

DAY I	DAY II	Day III
Gravity of AI	Inverse Problems	Achieving Kernel Awareness
Image Classification	Instabilities & Kernel Awareness	FIRENETs
Need for Foundations	Intriguing Barriers	Imaging Applications
AI for Image Reconstruction	Algorithm Unrolling	Numerical Examples

Slides will be hosted at http://www.damtp.cam.ac.uk/user/mjc249/Talks.html.
Useful references for further reading in grey boxes.
Comments and suggestions welcome! (vegarant@math.uio.no, m.colbrook@damtp.cam.ac.uk)

Can we improve image reconstruction?

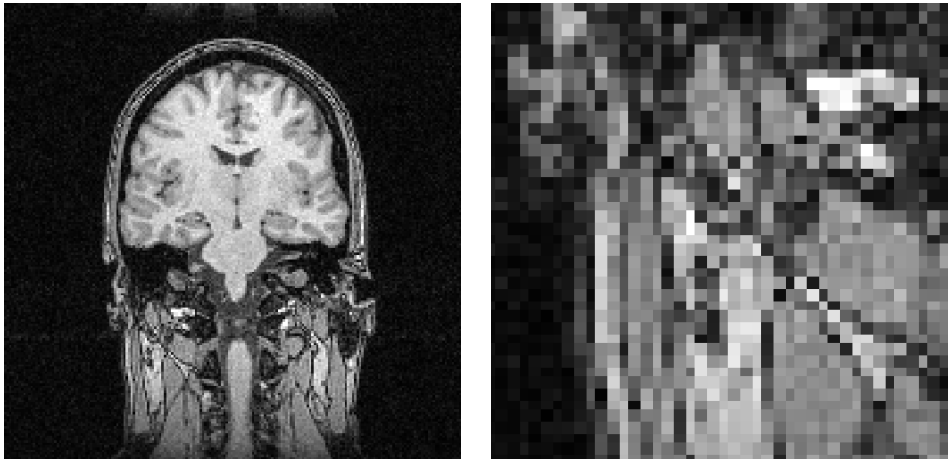


Figure: Standard 3D MRI headscan. Scanning time = 15 min (our experiment done with data from Cambridge University Hospital).

Experiment and data from Bogdan Roman and Anders C. Hansen

Can we improve image reconstruction?

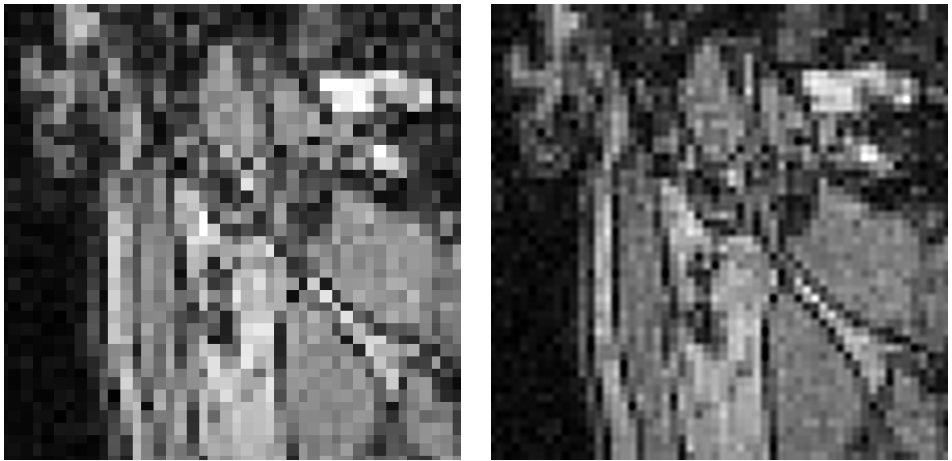


Figure: Left: Standard full sampling. Right: One type of compressed sensing approaches to resolution enhancing. Scanning time for both = 15 min.

Experiment and data from Bogdan Roman and Anders C. Hansen

Can we improve image reconstruction?

Figure: Two different compressed sensing approaches to resolution enhancing. Scanning time = 15 min

Experiment and data from Bogdan Roman and Anders C. Hansen

Can we improve image reconstruction?

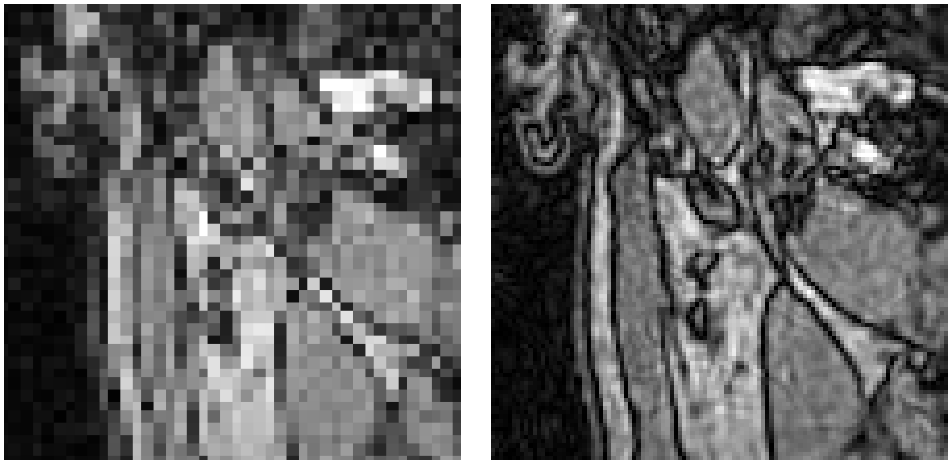


Figure: Left: Standard full sampling. Right: The correct type of compressed sensing approaches to resolution enhancing. Scanning time for both = 15 min

Experiment and data from Bogdan Roman and Anders C. Hansen

A discrete linear inverse problem

Given measurements $y = Ax + e$, of $x \in \mathcal{M}_1 \subset \mathbb{R}^N$, recover x.

- Here $A \in \mathbb{R}^{m \times N}$ is a (underdetermined) matrix with $m < N$,
- x is the **unknown** signal of interest,
- and e is noise or perturbations.

A discrete linear inverse problem

Given measurements $y = Ax + e$, of $x \in \mathcal{M}_1 \subset \mathbb{R}^N$, recover x.

- Here $A \in \mathbb{R}^{m \times N}$ is a (underdetermined) matrix with $m < N$,
- x is the **unknown** signal of interest,
- and e is noise or perturbations.

Examples of sets $\mathcal{M}_1$:

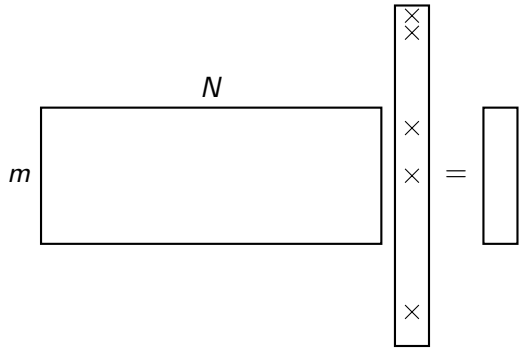
- $\mathcal{M}_1 =$ 'Set of natural images'
- $\mathcal{M}_1 =$ Set of s-sparse vectors
- $\mathcal{M}_1 = \mathcal{N}(A)^{\perp}$
- $\mathcal{M}_1 =$ Union of subspaces

Standard algorithms

–

Sparse solutions of linear systems and its' relation to imaging

Sparse linear systems



We say that a vector $x \in \mathbb{C}^N$ is s-sparse, if it has at most s non-zero components.

Sparse solutions of underdetermined systems have many applications!

- Linear regression in statistics – The LASSO
- Medical imaging - MRI, CT, microscopy ...
- Non-linear function approximation
- Error correction
- Explainable AI - LIME
- Dictionary learning and sparse coding
- Classification

How do we find sparse solutions?

Solve one of the problems:

Quadratically constrained basis pursuit (QCBP):

$$\min_{z\in\mathbb{C}^N} \|z\|_{l^1} \quad \text{subject to} \quad \|Az-y\|_{l^2} \leq \eta \tag{P_1}$$

Unconstrained LASSO (U-LASSO):

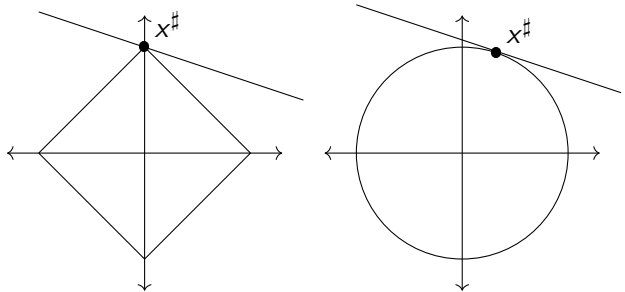
$$\min_{z\in\mathbb{C}^N} \|Az-y\|_{l^2}^2 + \lambda\|z\|_{l^1} \tag{P_2}$$

Square-root LASSO (SR-LASSO):

$$\min_{z\in\mathbb{C}^N} \|Az-y\|_{l^2} + \lambda\|z\|_{l^1} \tag{P_3}$$

We let $\Xi_j(y,A)$ denote the set of minimizers for (P_j), given input $A\in\mathbb{C}^{m\times N}, y\in\mathbb{C}^m$.

Why do we get sparse solutions?



The optimal solution

$$x^\sharp \in \operatorname{argmin}_{z \in \mathbb{R}^2} \|z\|_{l^p} \quad \text{subject to} \quad Az = y$$

for different values of p.

J. R. Statist. Soc. B (1996)
58, *No.* 1, *pp.* 267–288

Regression Shrinkage and Selection via the Lasso

By ROBERT TIBSHIRANI†

University of Toronto, Canada

[Received January 1994. Revised January 1995]

SUMMARY

We propose a new method for estimation in linear models. The 'lasso' minimizes the residual sum of squares subject to the sum of the absolute value of the coefficients being less than a constant. Because of the nature of this constraint it tends to produce some coefficients that are exactly 0 and hence gives interpretable models. Our simulation studies suggest that the lasso enjoys some of the favourable properties of both subset selection and ridge regression. It produces interpretable models like subset selection and exhibits the stability of ridge regression. There is also an interesting relationship with recent work in adaptive function estimation by Donoho and Johnstone. The lasso idea is quite general and can be applied in a variety of statistical models: extensions to generalized regression models and tree-based models are briefly described.

Keywords: QUADRATIC PROGRAMMING; REGRESSION; SHRINKAGE; SUBSET SELECTION

1. INTRODUCTION

Consider the usual regression situation: we have data $(\mathbf{x}^i, y_i)$, $i = 1, 2, \ldots, N$, where $\mathbf{x}^i = (x_{i1}, \ldots, x_{ip})^T$ and y_i are the regressors and response for the ith observation.

Google Scholar

regression shrinkage and selection via lasso

Artikler

Omtrent 50 400 resultater (**0,08** sek)

Når som helst
Etter 2021
Etter 2020
Etter 2017
Egendefinert periode

Regression shrinkage and **selection via** the **lasso**

R Tibshirani - Journal of the Royal Statistical Society: Series B …, 1996 - Wiley Online Library

We propose a new method for estimation in linear models. The '**lasso**'minimizes the residual sum of squares subject to the sum of the absolute value of the coefficients being less than a constant. Because of the nature of this constraint it tends to produce some coefficients that …

☆ Sitert av 37627 Beslektede artikler Alle 55 versjoner

Robust Uncertainty Principles: Exact Signal Reconstruction From Highly Incomplete Frequency Information

Emmanuel J. Candès, Justin Romberg, *Member, IEEE*, and Terence Tao

***Abstract*—This paper considers the model problem of reconstructing an object from incomplete frequency samples. Consider a discrete-time signal $f \in \mathbf{C}^N$ and a randomly chosen set of frequencies Ω. Is it possible to reconstruct f from the partial knowledge of its Fourier coefficients on the set Ω?**

A typical result of this paper is as follows. Suppose that f is a superposition of $|T|$ spikes $f(t) = \sum_{\tau \in T} f(\tau)\delta(t-\tau)$ obeying

$$|T| \leq C_M \cdot (\log N)^{-1} \cdot |\Omega|$$

for some constant $C_M > 0$. We do not know the locations of the spikes nor their amplitudes. Then with probability at least $1 - O(N^{-M})$, f can be reconstructed exactly as the solution to the ℓ_1 minimization problem

$$\min_g \sum_{t=0}^{N-1} |g(t)|, \quad \text{s.t. } \hat{g}(\omega) = \hat{f}(\omega) \text{ for all } \omega \in \Omega.$$

In short, exact recovery may be obtained by solving a convex optimization problem. We give numerical values for C_M which depend on the desired probability of success. Our result may be interpreted as a novel kind of nonlinear sampling theorem. In effect, it says that any signal made out of $|T|$ spikes may be recovered by convex programming from almost every set of frequencies of size $O(|T| \cdot \log N)$. Moreover, this is nearly optimal in the sense that

I. Introduction

In many applications of practical interest, we often wish to reconstruct an object (a discrete signal, a discrete image, etc.) from incomplete Fourier samples. In a discrete setting, we may pose the problem as follows; let $\hat{f}$ be the Fourier transform of a discrete object $f(t)$, $t = (t_1, \ldots, t_d) \in \mathbb{Z}_N^d := \{0, 1, \ldots, N-1\}^d$

$$\hat{f}(\omega) = \sum_{t \in \mathbb{Z}_N^d} f(t) e^{-2\pi i(\omega_1 t_1 + \cdots + \omega_d t_d)/N}.$$

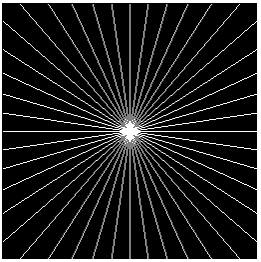
The problem is then to recover f from partial frequency information, namely, from $\hat{f}(\omega)$, where $\omega = (\omega_1, \ldots, \omega_d)$ belongs to some set Ω of cardinality less than N^d—the size of the discrete object.

In this paper, we show that we can recover f *exactly* from observations $\hat{f}|_\Omega$ on small set of frequencies provided that f is *sparse*. The recovery consists of solving a straightforward optimization problem that finds $f^\sharp$ of minimal complexity with $\hat{f}^\sharp(\boldsymbol{\omega}) = \hat{f}(\boldsymbol{\omega}), \forall\, \boldsymbol{\omega} \in \Omega$.

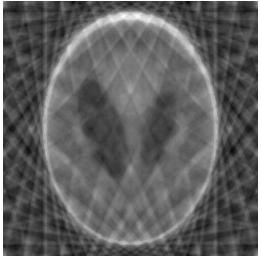
A. A Puzzling Numerical Experiment

A puzzling experiment

(a) Original



(b) Sampling map



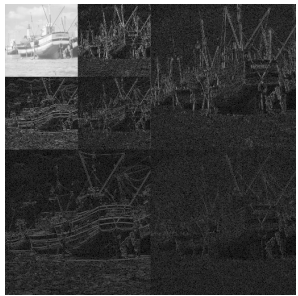
(c) Classical recovery (linear)

(d) Compressed sensing recovery

Images are sparse in transformed domains

Image x

Wx
$W =$ Wavelets



Wx
$W = \nabla$

In sparse regularization we use

$$\hat{x} \in \operatorname{argmin}_{z \in \mathbb{C}^N} \|Wz\|_{l^1} \quad \text{subject to} \quad \|Az - y\|_{l^2} \leq \eta$$

as our solution to the inverse problem.

Robust null space property

Notation: Let $\Omega \subset \{1, \dots, N\}$ and let $P_\Omega \in \mathbb{R}^{N \times N}$ be the projection

$$P_\Omega x = \begin{cases} x_i & i \in \Omega \\ 0 & \text{otherwise} \end{cases}.$$

Definition (Robust Null Space Property)

A matrix $A \in \mathbb{C}^{m \times N}$ satisfies the robust Null Space Property (rNSP) of order $1 \leq s \leq N$ with constants $0 < \rho < 1$ and $\gamma > 0$ if

$$\|P_\Omega x\|_{\ell^2} \leq \frac{\rho}{\sqrt{s}} \|P_\Omega^\perp x\|_{\ell^1} + \gamma \|Ax\|_{\ell^2},$$

for all $x \in \mathbb{C}^N$ and any $\Omega \subseteq \{1, \dots, N\}$ with $|\Omega| \leq s$.

Typical compressive sensing theorem

Theorem 1

Let $A \in \mathbb{C}^{m \times N}$ with $m < N$ and let $W \in \mathbb{C}^{N \times N}$ be unitary. Suppose that AW^{-1} has the rNSP of order s with constants $0 < \rho < 1$ and $\gamma > 0$. Let $y = Ax + e$ and let $0 < \lambda \leq C_1/(\sqrt{s}C_2)$. Then every minimizer $\hat{x} \in \mathbb{C}^N$ of the problem

$$\min_{z \in \mathbb{C}^N} \lambda \|Wz\|_{l^1} + \|Az - y\|_{l^2} \tag{P_3}$$

satisfies

$$\|\hat{x} - x\|_{l^2} \leq 2C_1 \frac{\sigma_s(Wx)_{l^1}}{\sqrt{s}} + \left(\frac{C_1}{\sqrt{s}\lambda} + C_2 \right) \|e\|_{l^2},$$

where C_1 and C_2 are the constants in (10), and

$$\sigma_s(z)_{l^1} := \inf\{\|z - t\|_{l^1} : t \text{ is a } s\text{-sparse vector}\}$$

denotes the distance to a s-sparse vector.

Reading material

- Adcock, B., & Hansen, A. C., '*Compressive Imaging: Structure, Sampling, Learning*', Cambridge University Press, 2021 (to appear). `https://www.compressiveimagingbook.com`
- Foucart, S., & Rauhut, H., '*A Mathematical Introduction to Compressive Sensing*', birkhäuser, 2013.

AI replacing standard algorithms in inverse problems

The basic inverse problem – Image denoising

Clear image $x \in \mathbb{C}^N$ is contaminated by unknown noise e, and we are given access to measurements of the form

$$y = x + e,$$

The task is to reconstruct x from the noisy measurements y.

The Basics of Deep Learning in Denoising

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

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

In practice, one tries to learn the noise and use

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

The Basics of Deep Learning in Denoising

Denoising experiment with deep learning

Original

Noisy version

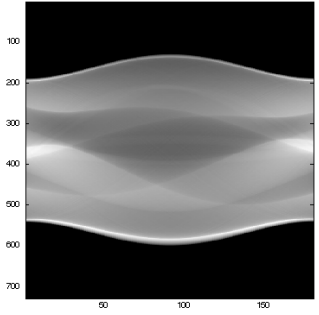
Denoised with Neur. Net.

DL in Inverse Problems: 1st Step

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

Logan-Shepp Phantom

The image under the Radon transform (sinogram)

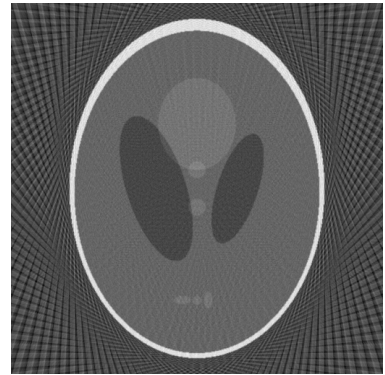


DL in Inverse Problems: 1st Step

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

Logan-Shepp Phantom

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

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

Deep Convolutional Neural Network for Inverse Problems in Imaging

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

Abstract

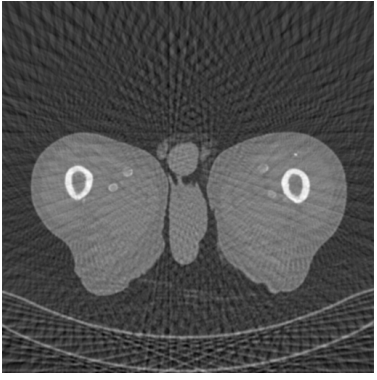
In this paper, we propose a novel deep convolutional neural network (CNN)-based algorithm for solving ill-posed inverse problems. Regularized iterative algorithms have emerged as the standard approach to ill-posed inverse problems in the past few decades. These methods produce excellent results, but 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

DL in Inverse Problems: 1st Step (Experiments)

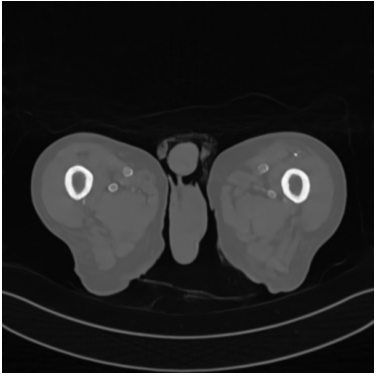
Computerised Tomography (CT) experiment with deep learning

Original

Recon-FBP (50 lines)



Recon-NeurNet (50 lines)



Can neural networks for image reconstruction be unstable?

The Instability Test

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

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

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

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

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

or

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

How do we compute worst-case perturbations?

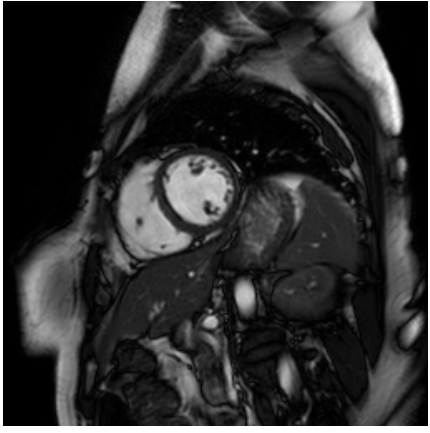
To find a worst case perturbation for a neural network $\Psi\colon \mathbb{C}^m \to \mathbb{C}^N$ we seek to maximize

$$Q(r) \coloneqq \|\Psi(A(x+r)) - \Psi(Ax)\|_{l^2}^2 - \lambda\|r\|_{l^2}^2$$

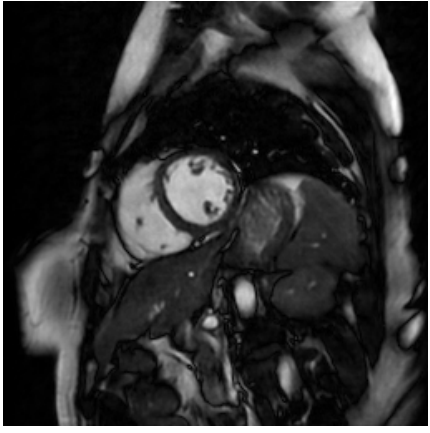
using a gradient based optimization method.

AI generated hallucinations – Instabilities

$|x|$



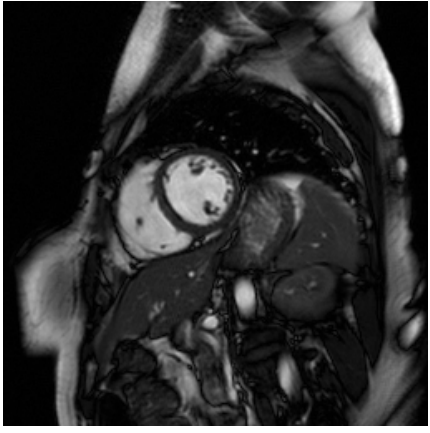
$|\Psi(Ax)|$



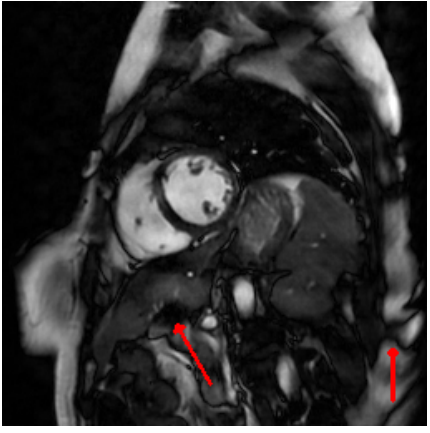
Network from: J. Schlemper, J. Caballero, J. V. Hajnal, A. Price and D. Rueckert, '*A deep cascade of convolutional neural networks for MR image reconstruction*', in International conference on information processing in medical imaging, Springer, 2017, pp. 647–658.
Figures from: Antun, V., Renna, F., Poon, C., Adcock, B., & Hansen, A. C., '*On instabilities of deep learning in image reconstruction and the potential costs of AI*'. Proc. Natl. Acad. Sci. USA, 2020..

AI generated hallucinations – Instabilities

$|x + r_1|$



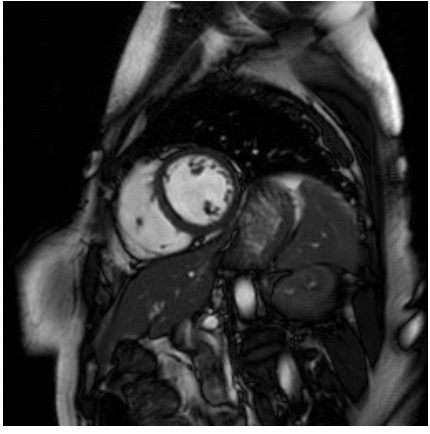
$|\Psi(A(x + r_1))|$



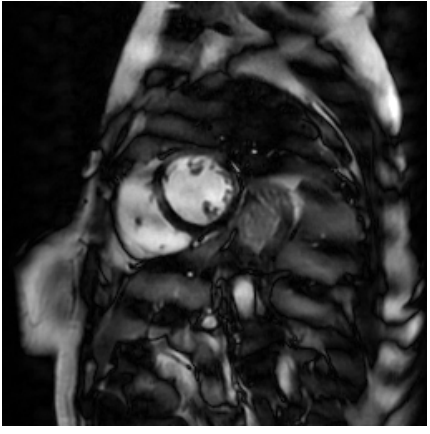
Network from: J. Schlemper, J. Caballero, J. V. Hajnal, A. Price and D. Rueckert, '*A deep cascade of convolutional neural networks for MR image reconstruction*', in International conference on information processing in medical imaging, Springer, 2017, pp. 647–658.
Figures from: Antun, V., Renna, F., Poon, C., Adcock, B., & Hansen, A. C., '*On instabilities of deep learning in image reconstruction and the potential costs of AI*'. Proc. Natl. Acad. Sci. USA, 2020..

AI generated hallucinations – Instabilities

$|x + r_2|$



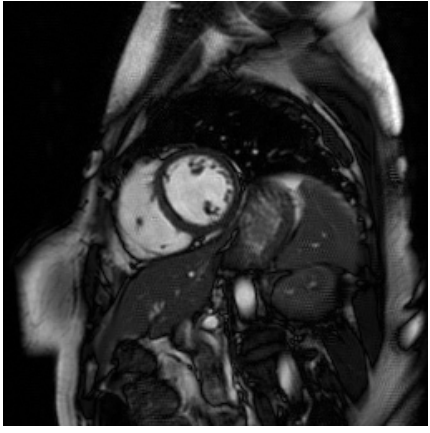
$|\Psi(A(x + r_2))|$



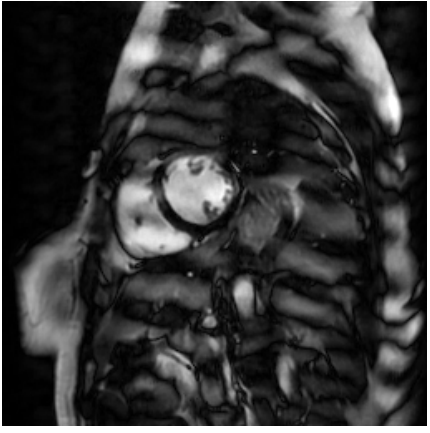
Network from: J. Schlemper, J. Caballero, J. V. Hajnal, A. Price and D. Rueckert, '*A deep cascade of convolutional neural networks for MR image reconstruction*', in International conference on information processing in medical imaging, Springer, 2017, pp. 647–658.
Figures from: Antun, V., Renna, F., Poon, C., Adcock, B., & Hansen, A. C., '*On instabilities of deep learning in image reconstruction and the potential costs of AI*'. Proc. Natl. Acad. Sci. USA, 2020..

AI generated hallucinations – Instabilities

$|x + r_3|$



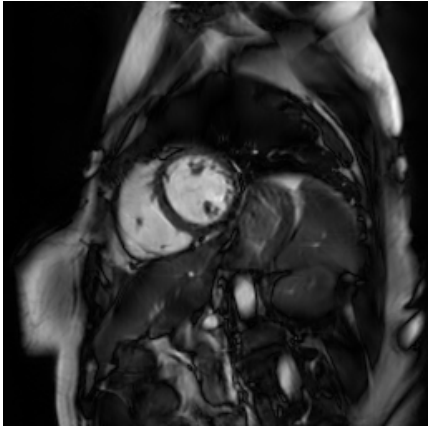
$|\Psi(A(x + r_3))|$



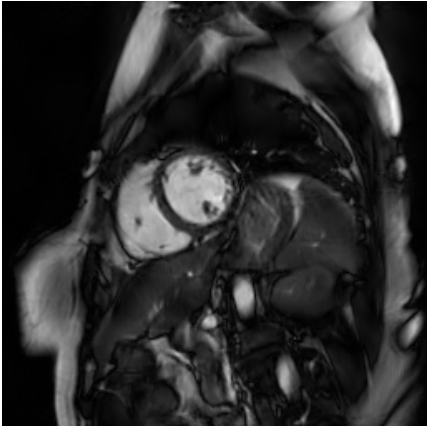
Network from: J. Schlemper, J. Caballero, J. V. Hajnal, A. Price and D. Rueckert, '*A deep cascade of convolutional neural networks for MR image reconstruction*', in International conference on information processing in medical imaging, Springer, 2017, pp. 647–658.
Figures from: Antun, V., Renna, F., Poon, C., Adcock, B., & Hansen, A. C., '*On instabilities of deep learning in image reconstruction and the potential costs of AI*'. Proc. Natl. Acad. Sci. USA, 2020..

Reconstruction using state-of-the-art standard methods

SoA from Ax

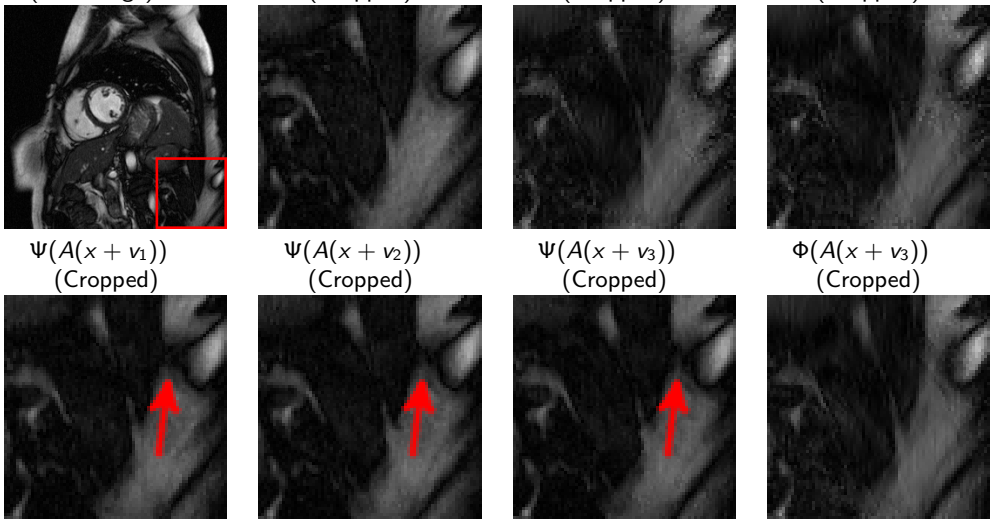


SoA from $A(x + r_3)$



Figures from: Antun, V., Renna, F., Poon, C., Adcock, B., & Hansen, A. C., *'On instabilities of deep learning in image reconstruction and the potential costs of AI'*. Proc. Natl. Acad. Sci. USA, 2020..

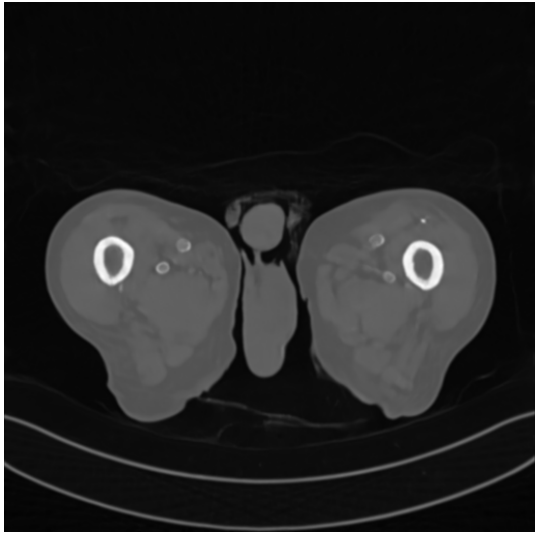
AI generated hallucinations – Random noise



AI generated hallucinations – Instabilities

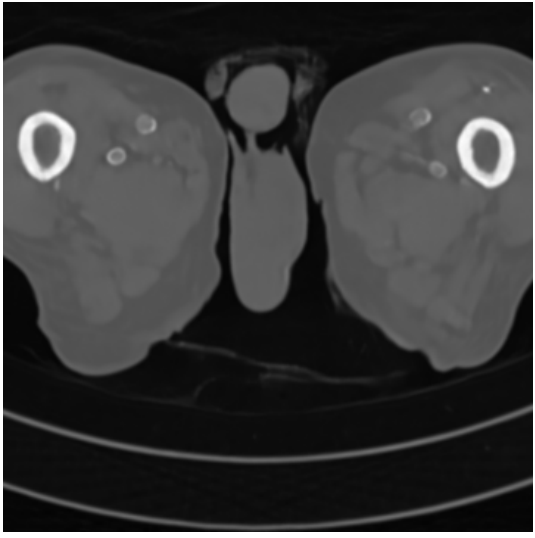
x

$\Psi(Ax))$

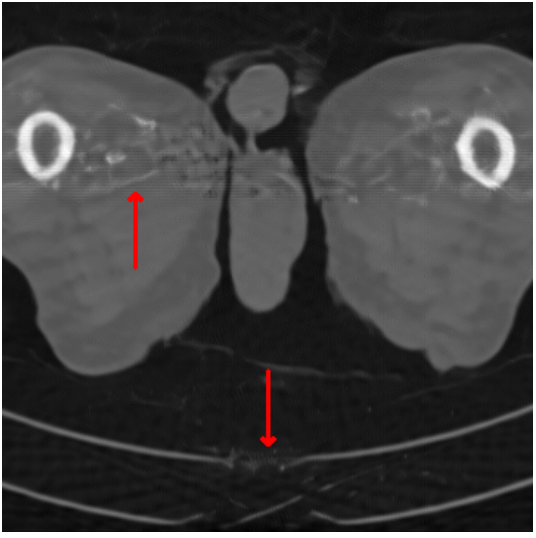


AI generated hallucinations – Instabilities

$x + r$

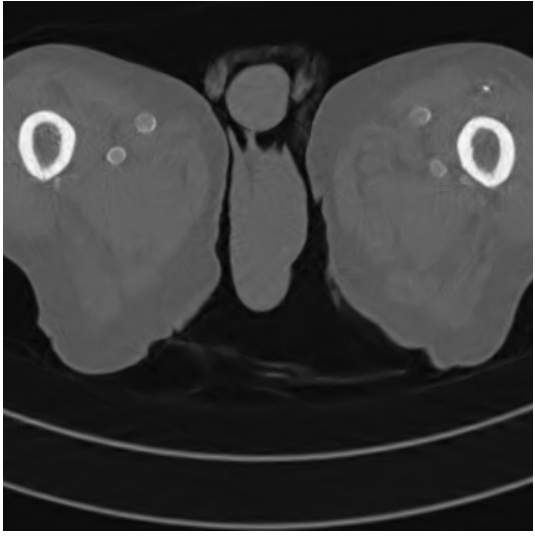


$\Psi(A(x + r))$

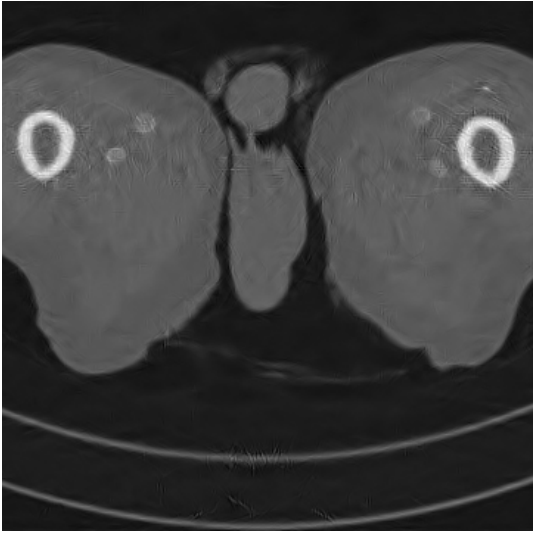


Testing on standard methods

State-of-the-art from Ax



State-of-the-art from $A(x + r)$



AI generated hallucinations – Instabilities

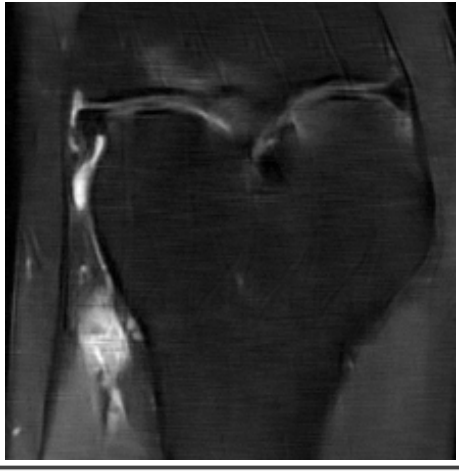
x

$\Psi(Ax)$

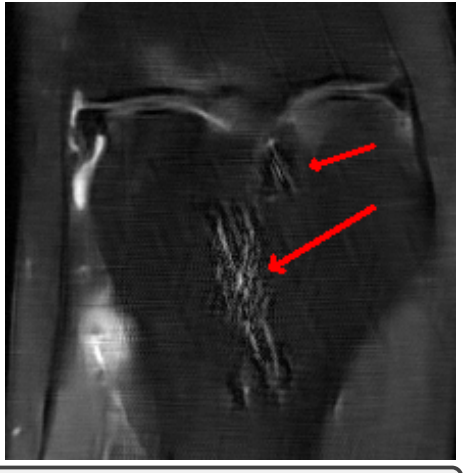
Network architecture from: Hammernik, K., Klatzer, T., Kobler, E., Recht, M. P., Sodickson, D. K., Pock, T., & Knoll, F., '*Learning a variational network for reconstruction of accelerated MRI data*'. Magnetic resonance in medicine, 79(6), 3055-3071.
Figures from: Antun, V., Renna, F., Poon, C., Adcock, B., & Hansen, A. C., '*On instabilities of deep learning in image reconstruction and the potential costs of AI*'. Proc. Natl. Acad. Sci. USA, 2020..

AI generated hallucinations – Instabilities

$x + r$



$\Psi(A(x + r))$



Network architecture from: Hammernik, K., Klatzer, T., Kobler, E., Recht, M. P., Sodickson, D. K., Pock, T., & Knoll, F., '*Learning a variational network for reconstruction of accelerated MRI data*'. Magnetic resonance in medicine, 79(6), 3055-3071.
Figures from: Antun, V., Renna, F., Poon, C., Adcock, B., & Hansen, A. C., '*On instabilities of deep learning in image reconstruction and the potential costs of AI*'. Proc. Natl. Acad. Sci. USA, 2020..

Testing on standard methods

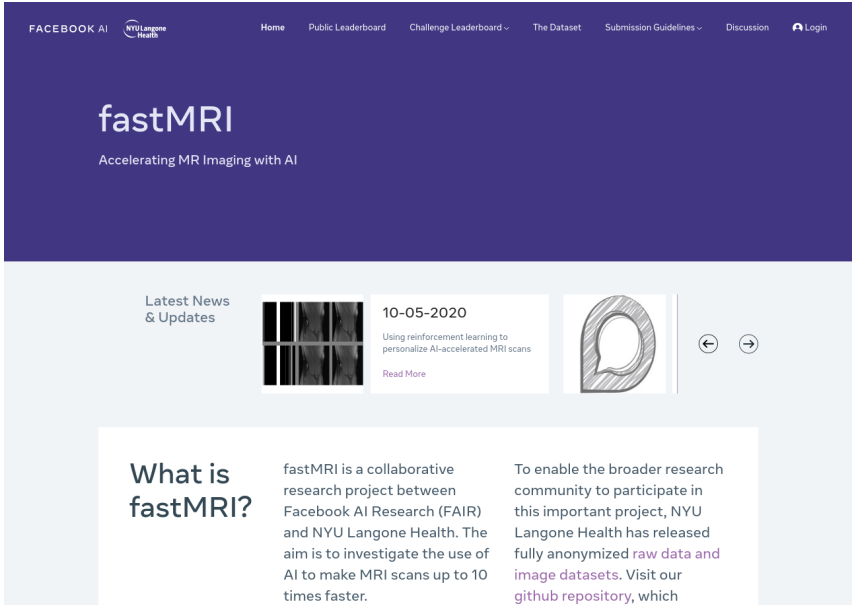
State-of-the-art from Ax

State-of-the-art from $A(x + r)$

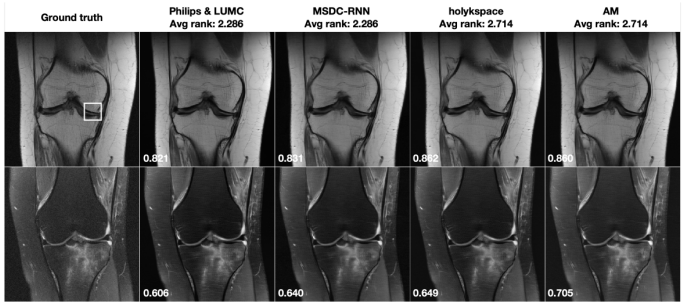
Figures from: Antun, V., Renna, F., Poon, C., Adcock, B., & Hansen, A. C., *'On instabilities of deep learning in image reconstruction and the potential costs of AI'*. Proc. Natl. Acad. Sci. USA, 2020..

We are not only striving for stable methods, but also accurate methods.

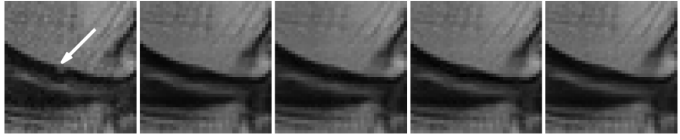
Accelerating MR Imaging with AI – fastMRI



False negatives – 4 x Speedup



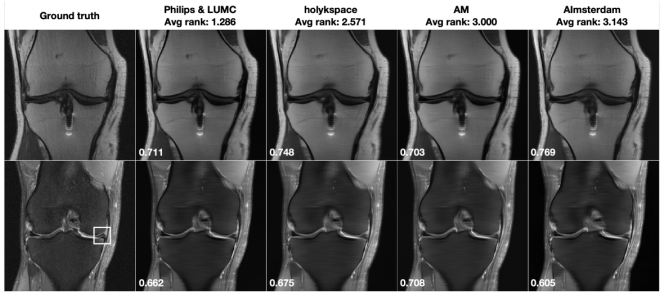
(a) Top row: Results for one slice from an acquisition without fat suppression. This case shows subtle pathology in the ROI indicated by a white rectangle in the ground truth image. Bottom row: One slice from an acquisition with fat suppression.



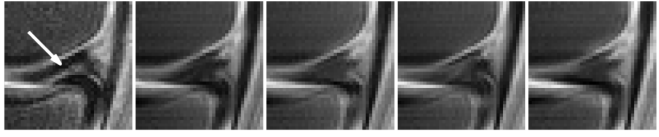
(b) Zoomed view of the ROI that shows a subchondral osteophyte (highlighted by a white arrow in the ground truth reconstruction). This pathology is not visible in any of the accelerated reconstructions.

Figure from: Knoll, Florian, et al. '*Advancing machine learning for MR image reconstruction with an open competition: Overview of the 2019 fastMRI challenge*'. Magnetic Resonance in Medicine (2020).

False negatives – 8 x Speedup



(a) Top row: Results for one slice from an acquisition without fat suppression. This case shows shows moderate artifact from a metal implant. Bottom row: One slice from an acquisition with fat suppression. This case shows shows a meniscal tear in the ROI indicated by a white rectangle in the ground truth image.



(b) Zoomed view of the ROI that shows a meniscal tear (highlighted by a white arrow in the ground truth reconstruction). This

Figure from: Knoll, Florian, et al. '*Advancing machine learning for MR image reconstruction with an open competition: Overview of the 2019 fastMRI challenge*'. Magnetic Resonance in Medicine (2020).

The Accuracy Test

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

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

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

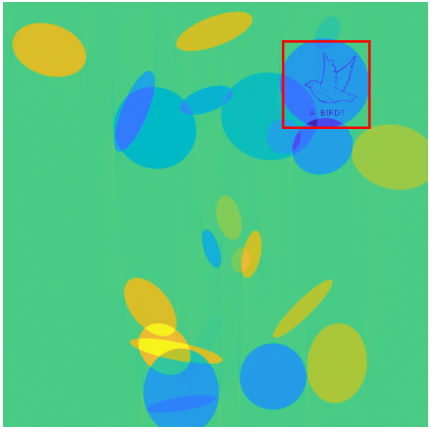
Check if the network can reconstruct unseen detail x_{detail} with $\|x_{\text{detail}}\| \leq \delta$, where $\delta > 0$ is small, such that

$$\Psi(y_{\text{data}} + A x_{\text{detail}}) = x_{\text{image}} + x_{\text{detail}},$$

NB: To make the test fair, we make the detail just large enough for a standard method to capture it.

AI generated hallucinations – Lack of accuracy

(Image + detail) $x + r$



Cropped version

AI generated hallucinations – Lack of accuracy

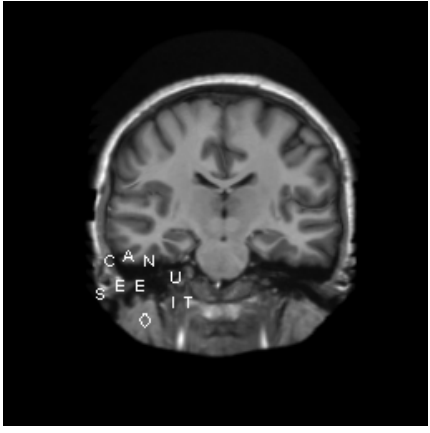
Network rec. $\Psi(A(x+r))$ (cropped)

State-of-the-art: $\Phi(A(x+r))$ (cropped)

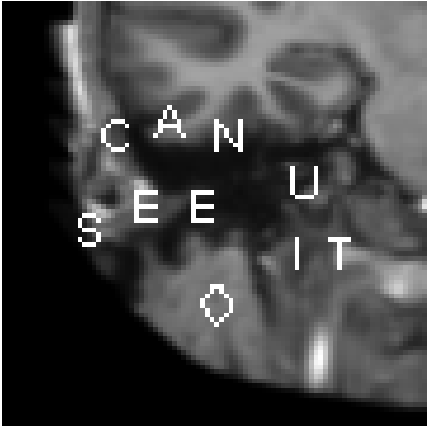
Network from: K. H. Jin, M. T. McCann, E. Froustey and M. Unser, '*Deep convolutional neural network for inverse problems in imaging*', IEEE Transactions on Image Processing, vol. 26, no. 9, pp. 4509–4522, 2017.

AI generated hallucinations – Lack of accuracy

(Image+detail) $x + r$



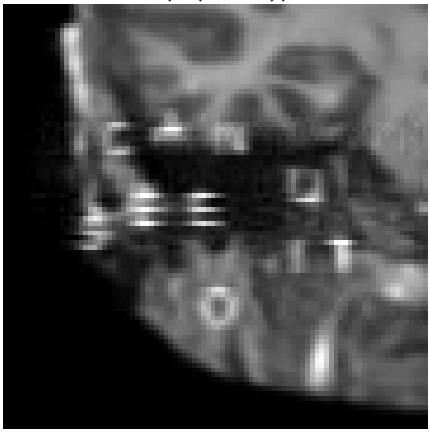
(cropped version)



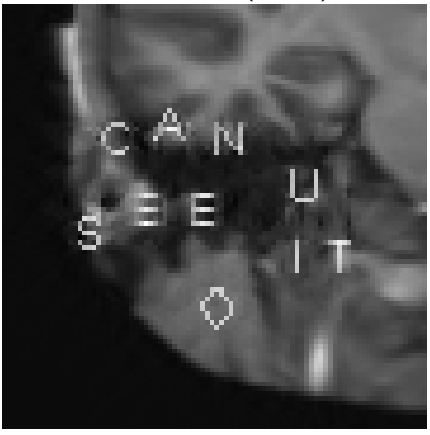
Figures from: Antun, V., Renna, F., Poon, C., Adcock, B., & Hansen, A. C., '*On instabilities of deep learning in image reconstruction and the potential costs of AI*'. Proc. Natl. Acad. Sci. USA, 2020..

AI generated hallucinations – Lack of accuracy

$\Psi(A(x+r))$



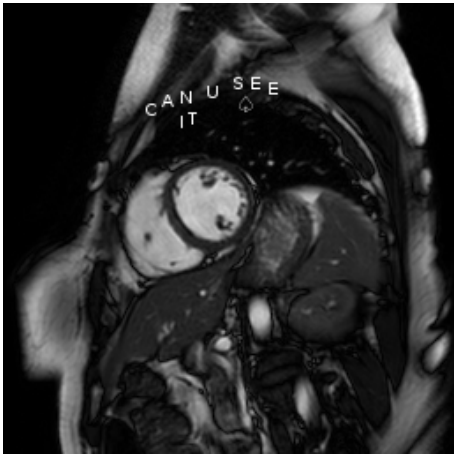
SoA from $A(x+r)$



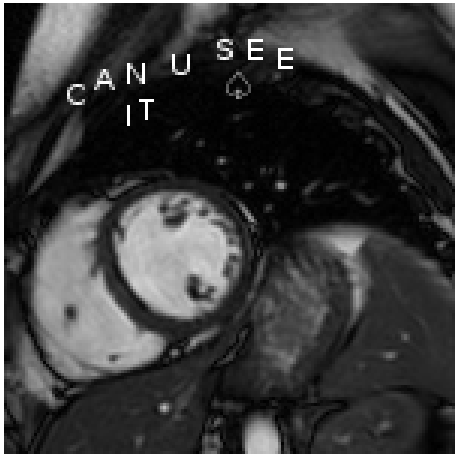
Network from: G. Yang, S. Yu, H. Dong, G. Slabaugh, P. L. Dragotti, X. Ye, F. Liu, S. Arridge, J. Keegan, Y. Guo and D. Firmin, *DAGAN: Deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction*, IEEE Transactions on Medical Imaging, 2017.
Figures from: Antun, V., Renna, F., Poon, C., Adcock, B., & Hansen, A. C., *'On instabilities of deep learning in image reconstruction and the potential costs of AI'*. Proc. Natl. Acad. Sci. USA, 2020..

AI generated hallucinations – Lack of accuracy

(Image+detail) $x + r$



(Cropped version)



Figures from: Antun, V., Renna, F., Poon, C., Adcock, B., & Hansen, A. C., '*On instabilities of deep learning in image reconstruction and the potential costs of AI*'. Proc. Natl. Acad. Sci. USA, 2020..

AI generated hallucinations – Lack of accuracy

$\Psi(A(x+r))$

SoA from $A(x+r)$

Network from: J. Schlemper, J. Caballero, J. V. Hajnal, A. Price and D. Rueckert, '*A deep cascade of convolutional neural networks for MR image reconstruction*', in International conference on information processing in medical imaging, Springer, 2017, pp. 647–658.
Figures from: Antun, V., Renna, F., Poon, C., Adcock, B., & Hansen, A. C., '*On instabilities of deep learning in image reconstruction and the potential costs of AI*'. Proc. Natl. Acad. Sci. USA, 2020..

Why does the AI halucinate?

There is a trade-off between stability and accuracy.

Why do we see instabilites?

The troublesome kernel: why deep learning for inverse problems is typically unstable

Nina M. Gottschling* Vegard Antun† Ben Adcock‡ Anders C. Hansen§

January 7, 2020

Abstract

There is overwhelming empirical evidence that Deep Learning (DL) leads to unstable methods in applications ranging from image classification and computer vision to voice recognition and automated diagnosis in medicine. Recently, a similar instability phenomenon has been discovered when DL is used to solve certain problems in computational science, namely, inverse problems in imaging. In this paper we present a comprehensive mathematical analysis explaining the many facets of the instability phenomenon in DL for inverse problems. Our main results not only explain why this phenomenon occurs, they also shed light as to why finding a cure for instabilities is so difficult in practice. Additionally, these theorems show that instabilities are typically not rare events – rather, they can occur even when the measurements are subject to completely random noise – and consequently how easy it can be to destabilise certain trained neural networks. We also examine the delicate balance between reconstruction performance and stability, and in particular, how DL methods may outperform state-of-the-art sparse regularization methods, but at the cost of instability. Finally, we demonstrate a counterintuitive phenomenon: training a neural network may generically not yield an optimal reconstruction method for an inverse problem.

Keywords: Deep learning, stability, inverse problems, imaging, sparse regularization

Mathematics Subject Classification (2010): 65R32, 94A08, 68T05, 65M12

Overperformance yields instabilities

Theorem 2 (Universal Instability Theorem)

Let $A \in \mathbb{C}^{m\times N}$, where $m < N$, and let $\Psi : \mathbb{C}^m \to \mathbb{C}^N$ be a continuous map. Suppose there are $x, x' \in \mathbb{C}^N$ and $\eta > 0$ such that

$$\|\Psi(Ax) - x\| < \eta, \quad \text{and} \quad \|\Psi(Ax') - x'\| < \eta, \tag{1}$$

and

$$\|Ax - Ax'\| < \eta. \tag{2}$$

We then have the following:

(i) **(Instability with respect to worst-case perturbations)** *Then the local ε-Lipschitz constant at $y = Ax$ satisfies*

$$L^{\varepsilon}(\Psi, y) := \sup_{0<\|z-y\|\leq\varepsilon} \frac{\|\Psi(z) - \Psi(y)\|}{\|z - y\|} \geq \frac{1}{\varepsilon}\left(\|x - x'\| - 2\eta\right), \qquad \forall \varepsilon \geq \eta. \tag{3}$$

Overperformance yields instabilities

Theorem 2 (Universal Instability Theorem)

Let $A \in \mathbb{C}^{m\times N}$, where $m < N$, and let $\Psi : \mathbb{C}^m \to \mathbb{C}^N$ be a continuous map. Suppose there are $x, x' \in \mathbb{C}^N$ and $\eta > 0$ such that

$$\|\Psi(Ax) - x\| < \eta, \quad \text{and} \quad \|\Psi(Ax') - x'\| < \eta, \tag{4}$$

and

$$\|Ax - Ax'\| < \eta. \tag{5}$$

We then have the following:

(ii) **(False negatives)**. *There is a $z \in \mathbb{C}^N$ with $\|z\| \geq \|x - x'\|$, an $e \in \mathbb{C}^m$ with $\|e\| \leq \eta$ such that*

$$\|\Psi(A(x + z) + e) - x\| \leq \eta \tag{6}$$

Overperformance yields instabilities

Theorem 2 (Universal Instability Theorem)

Let $A \in \mathbb{C}^{m\times N}$, where $m < N$, and let $\Psi : \mathbb{C}^m \to \mathbb{C}^N$ be a continuous map. Suppose there are $x, x' \in \mathbb{C}^N$ and $\eta > 0$ such that

$$\|\Psi(Ax) - x\| < \eta, \quad \text{and} \quad \|\Psi(Ax') - x'\| < \eta, \tag{7}$$

and

$$\|Ax - Ax'\| < \eta. \tag{8}$$

We then have the following:

(ii) **(False positives)**. *There is a $z \in \mathbb{C}^N$ with $\|z\| \geq \|x - x'\|$, an $e \in \mathbb{C}^m$ with $\|e\| \leq \eta$ such that*

$$\|\Psi(Ax + e) - (x + z)\| \leq \eta \tag{9}$$

Paradox I:

If a reconstruction method becomes 'too accurate', it will become unstable.

However, if a reconstruction method becomes 'too stable', it can not be very accurate.

Kernel awareness is needed to protect against overperformance

Kernel awareness: The cardinal sin of recovering two elements x and x', whose difference $x - x'$ lies close to the kernel of A.

Many popular defence techniques provide no guarantees against overperformance:

- Enforcing consistency
- Training with random sampling patterns
- Adding random noise
- Adversarial training/augmenting the training set

For details see:
Gottschling, N. M., Antun, V., Adcock, B., & Hansen, A. C. (2020). *The troublesome kernel: why deep learning for inverse problems is typically unstable*. arXiv preprint arXiv:2001.01258. `https://arxiv.org/abs/2001.01258`

Kernel awareness is built into compressive sensing theory

Theorem 3

Suppose the matrix $A \in \mathbb{C}^{m \times N}$ satisfies the robust null space property (rNSP) or order s, with constants $0 < \rho < 1$ and $\gamma > 0$. Then for all s-sparse vectors $x, z \in \mathbb{C}^N$,

$$\|z - x\|_{l^2} \leq \frac{C_2}{2} \|A(z - x)\|_{l^2}$$

where

$$C_2 = \frac{(3\rho + 5)\gamma}{1 - \rho}. \tag{10}$$

Recap: Classification problems

Problem: Approximate function $f\colon \mathcal{M} \subset \mathbb{R}^d \to \{0,1\}$

Approach: Sample the graph $\{(x, f(x)) : x \in \mathcal{M}\}$ and use this information to compute an approximation $\tilde{f}\colon \mathcal{M} \to \{0,1\}$

Question in imaging:

- Will a map $\tilde{f}\colon \mathbb{C}^m \to \mathbb{C}^N$ satisfying $\|\tilde{f}(Ax^i) - x_i\| \leq \delta$ for $i = 1, \ldots, r$, be optimal?
- Does there exist a map $\tilde{f}\colon \mathbb{C}^m \to \mathbb{C}^N$, statisfying $\|\tilde{f}(Ax^i) - x_i\| \leq \delta$ for $i = 1, \ldots, r$?

Inverse problems in imaging

Convenient to describe the inverse problem in terms of a triple $\{A, \mathcal{M}_1, \mathcal{M}_2\}$, given by

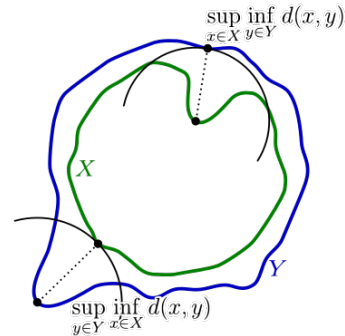
a *sampling map* $A \in \mathbb{C}^{m \times N}$, where $m < N$,
a *domain* $\mathcal{M}_1 \subset \mathbb{C}^N$, where $(\mathcal{M}_1, d_1)$ is a metric space,
the *range* $\mathcal{M}_2 = A(\mathcal{M}_1) \subset \mathbb{C}^m$, where $(\mathcal{M}_2, d_2)$ is also a metric space.

The inverse problem is now as follows:

Given measurements $y = Ax$ of $x \in \mathcal{M}_1$, recover x.

Notation

- A multivalued mapping is written with double arrows as
$$\varphi : \mathcal{M}_2 \rightrightarrows \mathbb{C}^N$$
- We use the Hausdorff metric on the set of bounded subsets of $\mathcal{M}_1$,
$$d_1^H(Z, X) = \max\{ \sup_{x \in X} \inf_{z \in Z} d_1(z, x), \sup_{z \in Z} \inf_{x \in X} d_1(z, x) \},$$
With slight abuse of notation we will denote a singleton $\{x\} \subset \mathcal{M}_1$ by x.



What do we try to learn? The optimal map

Definition 4 (Optimal map)

Let d_1 be a metric on $\mathbb{C}^N$, $\mathcal{M}_1 \subset \mathbb{C}^N$, $A \in \mathbb{C}^{m\times N}$ and $\mathcal{M}_2 = A(\mathcal{M}_1)$. The optimality constant of $\{A, \mathcal{M}_1\}$ is

$$c_{\mathrm{opt}}(A, \mathcal{M}_1) = \inf_{\varphi:\mathcal{M}_2 \rightrightarrows \mathbb{C}^N} \sup_{x\in\mathcal{M}_1} d_1^H(\varphi(Ax), x). \tag{11}$$

A map $\varphi\colon \mathcal{M}_2 \rightrightarrows \mathbb{C}^N$ is optimal if it attains the infimum in (11)

Paradox II: We do not know the optimal map

Theorem 2

Let the metrics d_1 and d_2 be on $\mathbb{C}^N$ and $\mathbb{C}^m$ be induced by norms. Let $A \in \mathbb{C}^{m\times N}$ with $\operatorname{rank}(A) \geq 1$, where $m < N$, $K \in \{2, \dots, \infty\}$, $\delta \leq 1/5$ and $\mathcal{B} \subset \mathbb{C}^N$ be the closed unit ball (with respect to d_1). Then the following holds:

(i) *(**Training may not yield optimal maps**). There exist uncountably many $\mathcal{M}_1 \subset \mathcal{B}$, such that for each $\mathcal{M}_1$ there exist uncountably many sets $\mathcal{T} \subset \mathcal{M}_2 \times \mathcal{M}_1$ with $|\mathcal{T}| = K$, where $\mathcal{M}_2 = A(\mathcal{M}_1)$, satisfying the following. Any map $\Psi : \mathcal{M}_2 \to \mathcal{M}_1$ (potentially multivalued $\Psi : \mathcal{M}_2 \rightrightarrows \mathcal{M}_1$) satisfying*

$$d_1^H(\Psi(y), x) \leq \delta, \qquad \forall (y, x) \in \mathcal{T}, \tag{12}$$

is not an optimal map. If K is finite, one can choose $|\mathcal{M}_1| = K + 1$.

Paradox II: We do not know the optimal map

Theorem 3

Let the metrics d_1 and d_2 be on $\mathbb{C}^N$ and $\mathbb{C}^m$ be induced by norms. Let $A \in \mathbb{C}^{m\times N}$ with $\operatorname{rank}(A) \geq 1$, where $m < N$, $K \in \{2, \ldots, \infty\}$, $\delta \leq 1/5$ and $\mathcal{B} \subset \mathbb{C}^N$ be the closed unit ball (with respect to d_1). Then the following holds:

(ii) *(**The map sought by training may not exist**). There exist uncountably many domains $\mathcal{M}_1 \subset \mathcal{B}$ with $|\mathcal{M}_1| = K$ such that, with $\mathcal{M}_2 = A(\mathcal{M}_1)$, there does not exist a map $\Psi : \mathcal{M}_2 \to \mathcal{M}_1$ (nor a multivalued map $\Psi : \mathcal{M}_2 \rightrightarrows \mathcal{M}_1$) for which*

$$d_1^H(\Psi(y), x) \leq \delta, \qquad \forall (y, x) \in \mathcal{M}_2 \times \mathcal{M}_1.$$

Stable and accurate reconstruction is only possible under certain conditions

Can we compute neural networks that solve (P_j)?

$$\min_{x \in \mathbb{C}^N} \|x\|_{l^1} \quad \text{subject to} \quad \|Ax - y\|_{l^2} \leq \eta \qquad (P_1)$$

$$\min_{x \in \mathbb{C}^N} \lambda \|x\|_{l^1} + \|Ax - y\|_{l^2}^2 \qquad (P_2)$$

$$\min_{x \in \mathbb{C}^N} \lambda \|x\|_{l^1} + \|Ax - y\|_{l^2} \qquad (P_3)$$

Interested in the **minimising** vectors (denoted Ξ).

Why the problems (P_j)?

- Avoid bizarre, unnatural & pathological mappings: (P_j) well-understood & well-used!
- Simpler solution map than inverse problem $\Rightarrow$ stronger impossibility results.
- Sparse regularisation is often used as a benchmark method.
- DL has also been used to speed up sparse regularisation and tackle (P_j).
  (see section on LISTA and unrolling later)

Recall - 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.

The following theorems showcase (ii) and (iii).

The set-up

$$A \in \mathbb{C}^{m\times N} \text{ (modality)}, \quad \mathcal{S} = \{y_k\}_{k=1}^R \subset \mathbb{C}^m \text{ (samples)}, \quad R < \infty$$

Question: Given a collection Ω of $(A, \mathcal{S})$, does there exist a neural network approximating Ξ (solution map of (P_j)), and can it be trained by an algorithm?

In practice, the matrix A is not known exactly or cannot be stored to infinite precision.

Assume access to: $\{y_{k,n}\}_{k=1}^R$ and A_n (rational approximations, e.g. floats) such that

$$\|y_{k,n} - y_k\| \leq 2^{-n}, \quad \|A_n - A\| \leq 2^{-n}, \quad \forall n \in \mathbb{N}.$$

And $\{x_{k,n}\}_{k=1}^R$ such that $\inf_{x^*\in\Xi(A_n,y_{k,n})} \|x_{k,n} - x^*\| \leq 2^{-n}, \quad \forall n \in \mathbb{N}.$

Training set associated with $(A, \mathcal{S}) \in \Omega$ is

$$\iota_{A,\mathcal{S}} \coloneqq \{(y_{k,n}, A_n, x_{k,n}) \mid k = 1, \ldots, R, \text{ and } n \in \mathbb{N}\}.$$

Good news - a neural network exists

Theorem (Neural networks exist for Ξ)

For (P_j) and any family Ω of $(A, \mathcal{S})$, there exists a mapping

$$\mathcal{K}\colon \iota_{A,\mathcal{S}} \to \varphi_{A,\mathcal{S}} \text{ (a neural network)}$$

such that $\varphi_{A,\mathcal{S}}(y)$ solves (P_j) for each $y \in \mathcal{S}$. In other words, $\mathcal{K}$ maps the training data to NNs that solve the optimisation problem (P_j) for each $(A, \mathcal{S}) \in \Omega$.

Proof.

Easy - apply universal approximation/interpolation theorems. □

The set-up

$$A \in \mathbb{C}^{m\times N} \text{ (modality)}, \quad \mathcal{S} = \{y_k\}_{k=1}^R \subset \mathbb{C}^m \text{ (samples)}, \quad R < \infty$$

Question: Given a collection Ω of $(A, \mathcal{S})$, does there exist a neural network approximating Ξ (solution map of (P_j)), and can it be trained by an algorithm?

In practice, the matrix A is not known exactly or cannot be stored to infinite precision.

Assume access to: $\{y_{k,n}\}_{k=1}^R$ and A_n (rational approximations, e.g. floats) such that

$$\|y_{k,n} - y_k\| \leq 2^{-n}, \quad \|A_n - A\| \leq 2^{-n}, \quad \forall n \in \mathbb{N}.$$

And $\{x_{k,n}\}_{k=1}^R$ such that $\inf_{x^*\in\Xi(A_n,y_{k,n})} \|x_{k,n} - x^*\| \leq 2^{-n}, \quad \forall n \in \mathbb{N}.$

Training set associated with $(A, \mathcal{S}) \in \Omega$ is

$$\iota_{A,\mathcal{S}} \coloneqq \{(y_{k,n}, A_n, x_{k,n}) \mid k = 1, \ldots, R, \text{ and } n \in \mathbb{N}\}.$$

Bad news - can't necessarily approximate such a neural network

Theorem 4

For (P_j), $N \geq 2$ and $m < N$. Let $K > 2$ be a positive integer, $L \in \mathbb{N}$. Then there exists a **well-conditioned** *class (condition numbers ≤ 1) Ω of elements $(A, \mathcal{S})$ s.t. (Ω* **fixed** *in what follows):*

(i) *There* **does not exist any algorithm** *that, given a training set $\iota_{A,\mathcal{S}}$, produces a neural network $\phi_{A,\mathcal{S}}$ with*

$$\min_{y \in \mathcal{S}} \inf_{x^* \in \Xi(A,y)} \|\phi_{A,\mathcal{S}}(y) - x^*\|_{l^2} \leq 10^{-K}, \quad \forall (A, \mathcal{S}) \in \Omega. \tag{13}$$

Furthermore, for any $p > 1/2$, no probabilistic algorithm *can produce a neural network $\phi_{A,\mathcal{S}}$ such that (13) holds with probability at least p.*

(ii) *There* **exists an algorithm** *that produces a neural network $\phi_{A,\mathcal{S}}$ such that*

$$\max_{y \in \mathcal{S}} \inf_{x^* \in \Xi(A,y)} \|\phi_{A,\mathcal{S}}(y) - x^*\|_{l^2} \leq 10^{-(K-1)}, \quad \forall (A, \mathcal{S}) \in \Omega.$$

However, for any such algorithm (even probabilistic), $M \in \mathbb{N}$ and $p \in \left[0, \frac{N-m}{N+1-m}\right)$, there exists a training set $\iota_{A,\mathcal{S}}$ such that for all $y \in \mathcal{S}$,

$$\mathbb{P}\Big(\inf_{x^* \in \Xi(A,y)} \|\phi_{A,\mathcal{S}}(y) - x^*\|_{l^2} > 10^{1-K} \text{ or size of training data needed} > M \Big) > p.$$

(iii) *There* **exists an algorithm** *using only L training data from each $\iota_{A,\mathcal{S}}$ that produces a neural network $\phi_{A,\mathcal{S}}(y)$ such that*

$$\max_{y \in \mathcal{S}} \inf_{x^* \in \Xi(A,y)} \|\phi_{A,\mathcal{S}}(y) - x^*\|_{l^2} \leq 10^{-(K-2)}, \quad \forall (A, \mathcal{S}) \in \Omega.$$

Understanding the theorem statement I: In words...

Nice classes Ω where one can prove NNs with great approximation qualities exist. But:

- No algorithm, even randomised can train (or compute) such a NN accurate to K digits with probability greater than $1/2$.
- There exists a deterministic algorithm that computes a NN with $K-1$ correct digits, but any such (even randomised) algorithm needs arbitrarily many training data.
- There exists a deterministic algorithm that computes a NN with $K-2$ correct digits using no more than L training samples.

Existence vs computation (universal approximation/interpolation theorems **not** enough).

Conclusion: Theorems on existence of neural networks may have little to do with the neural networks produced in practice.

Understanding the theorem statement II: well-conditioned

- **Classical matrix condition number:** $\mathrm{Cond}(AA^*) = \|AA^*\|\|(AA^*)^{-1}\|$.
- **Of mapping:**

$$\lim_{\text{perturbation size} \downarrow 0} \frac{\text{distance between outputs}}{\text{distance between inputs}}$$

- **Feasibility:** For (P_1),

$$\frac{\text{size of problem}}{\text{distance to infeasibility}}$$

Useful book: Bürgisser P., Cucker F., 2013. *Condition : The geometry of numerical algorithms.*

Understanding the theorem statement III: what's an algorithm?

- For first statement: Any model (Turing machine, analog computers, etc.).
- For second and third statement: Turing machines (digital computers).

Result **independent of neural network architecture** - a universal barrier.

Use the **Solvability Complexity Index** (SCI) - tools that classify problems measuring their intrinsic difficulty and proving optimality of algorithms.

The SCI hierarchy

Key question: What is possible in scientific computation?

- Combine techniques from numerical analysis, functional analysis and approximation theory $\Rightarrow$ new algorithms.
- Classify problems in a computational hierarchy measuring their intrinsic difficulty and the optimality of algorithms $\Rightarrow$ prove that algorithms realise the boundaries of what computers can achieve.

- Colbrook, M.J., Roman, B. and Hansen, A.C., 2019. *How to compute spectra with error control.* Physical review letters, 122(25), p.250201.
- Colbrook, M.J., Horning, A. and Townsend, A. *Computing spectral measures of self-adjoint operators.* SIAM Review, to appear.
- Hansen, A., 2011. *On the solvability complexity index, the n-pseudospectrum and approximations of spectra of operators.* Journal of the American Mathematical Society, 24(1).
- Ben-Artzi, J., Colbrook, M.J., Hansen, A.C., Nevanlinna, O. and Seidel, M., 2020. *Computing Spectra - On the Solvability Complexity Index Hierarchy and Towers of Algorithms.* arXiv preprint arXiv:1508.03280.

Numerical example: fails in MATLAB

Centred and standardised (columns of the matrix A below are normalised) Lasso problem

$$\min_{x \in \mathbb{R}^N} \frac{1}{m} \|A_\delta D_\delta x - y\|_2^2 + \lambda \|x\|_1.$$

Take $m = 3, N = 2$, $\lambda = 1/10$, and

$$A_\delta = \begin{pmatrix} \frac{1}{\sqrt{2}} - \delta & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} - \delta & -\frac{1}{\sqrt{2}} \\ 2\delta & 0 \end{pmatrix} \in \mathbb{R}^{3 \times 2}, \quad y = \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 0 \end{pmatrix}^T \in \mathbb{R}^3,$$

where D_δ is the unique diagonal matrix s.t. columns of $A_\delta D_\delta$ each have norm $\sqrt{m}$.

Use MATLAB's `lasso` solver.

Numerical example: fails in MATLAB

Live demo...

Numerical example: fails in MATLAB

Default settings				'RelTol' $= \epsilon_{\text{mach}}$			'RelTol' $= \epsilon_{\text{mach}}$ 'MaxIter' $= \epsilon_{\text{mach}}^{-1}$		
δ	Error	RunTime	Warn	Error	RunTime	Warn	Error	RunTime	Warn
2^{-1}	$1 \cdot 10^{-16}$	$< 0.01s$	0	$1 \cdot 10^{-16}$	$< 0.01s$	0	$1 \cdot 10^{-16}$	$< 0.01s$	0
2^{-7}	0.68	$< 0.01s$	0	$2 \cdot 10^{-16}$	$0.02s$	0	$2 \cdot 10^{-16}$	$0.02s$	0
2^{-15}	1.17	$< 0.01s$	0	1.17	$0.33s$	1	$1 \cdot 10^{-11}$	$1381.5s$	0
2^{-20}	1.17	$< 0.01s$	0	1.17	$0.33s$	1	no output	$> 12h$	0
2^{-24}	1.17	$< 0.01s$	0	1.17	$0.34s$	1	no output	$> 12h$	0
2^{-26}	1.17	$< 0.01s$	0	1.17	$0.34s$	1	no output	$> 12h$	0
2^{-28}	1.17	$< 0.01s$	0	1.17	$< 0.01s$	0	1.17	$< 0.01s$	0
2^{-30}	1.17	$< 0.01s$	0	1.17	$< 0.01s$	0	1.17	$< 0.01s$	0

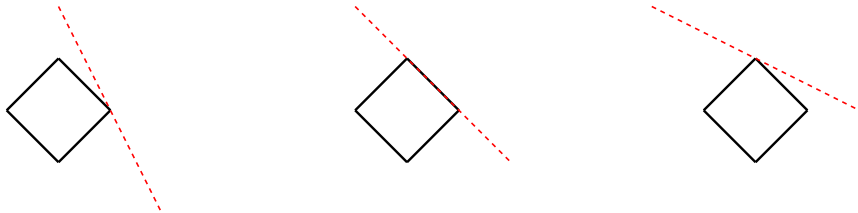
Most of the time, no warning is issued despite nonsensical outputs.

Numerical example: fails with training methods

$\text{dist}(\Psi_{A_n}(y_n), \Xi_3(A,y))$	$\text{dist}(\Phi_{A_n}(y_n), \Xi_3(A,y))$	$\|A_n - A\| \leq 2^{-n}$ $\|y_n - y\|_{l^2} \leq 2^{-n}$	10^{-K}	Ω_K
0.2999690	0.2597827	$n = 10$	10^{-1}	$K = 1$
0.3000000	0.2598050	$n = 20$	10^{-1}	$K = 1$
0.3000000	0.2598052	$n = 30$	10^{-1}	$K = 1$
0.0030000	0.0025980	$n = 10$	10^{-3}	$K = 3$
0.0030000	0.0025980	$n = 20$	10^{-3}	$K = 3$
0.0030000	0.0025980	$n = 30$	10^{-3}	$K = 3$
0.0000030	0.0000015	$n = 10$	10^{-6}	$K = 6$
0.0000030	0.0000015	$n = 20$	10^{-6}	$K = 6$
0.0000030	0.0000015	$n = 30$	10^{-6}	$K = 6$

Table: (**Impossibility of computing the existing neural network to arbitary accuracy**). A constructed from discrete cosine transform, $R = 8000$, $N = 20$, $m = 19$, solutions are 6-sparse. We demonstrate the impossibility statement (i) on FIRENETs Φ_{A_n}, and trained LISTA networks Ψ_{A_n}. The table shows the shortest l^2 distance between the output from the networks, and the true minimizer of the problem (P_3), with $w_l = 1$ and $\lambda = 1$, for different values of n and K.

The basic mechanism



Similar phase transitions can be built for (P_j) in arbitrary dimensions.

Algorithm unrolling:
Iterative algorithms for $(P_j) \Rightarrow$ *deep neural networks*

Monga, V., Li, Y. and Eldar, Y.C., 2019. *Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing.* arXiv preprint arXiv:1912.10557.

Iterative algorithms

Recall (P_2):

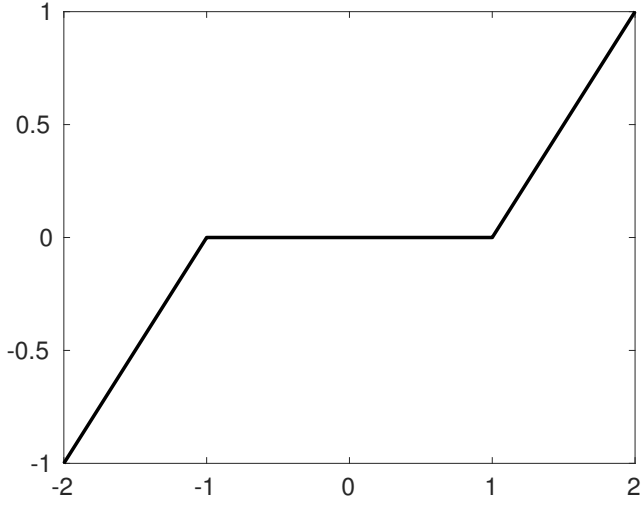
$$\text{argmin}_{x \in \mathbb{C}^N} F_2(x, y) := \lambda \|x\|_{l^1} + \|Ax - y\|_{l^2}^2.$$

Assume all vectors are real. Popular iterative method is the Iterative Shrinkage and Thresholding Algorithm (ISTA):

$$x^{(n+1)} = H_\theta \left(x^{(n)} - \frac{1}{L} A^*(Ax^{(n)} - y) \right).$$

Daubechies, I., Defrise, M. and De Mol, C., 2004. *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint.* Communications on Pure and Applied Mathematics, 57(11), pp.1413-1457.

Iterative algorithms



$$H_1(x) = (|x| - 1)_+ \operatorname{sign}(x)$$

Learned iterative shrinkage and thresholding algorithm (LISTA)

Idea: This looks like a neural network. Let's try learning!

Learn the linear maps.

Learn θ for

$$H_\theta(x) = (|x| - \theta)_+ \operatorname{sign}(x)$$

NB: We can apply the stability test through back-propagation.

Learning Fast Approximations of Sparse Coding

Karol Gregor and Yann LeCun {KGREGOR,YANN}@CS.NYU.EDU
Courant Institute, New York University, 715 Broadway, New York, NY 10003, USA

Abstract

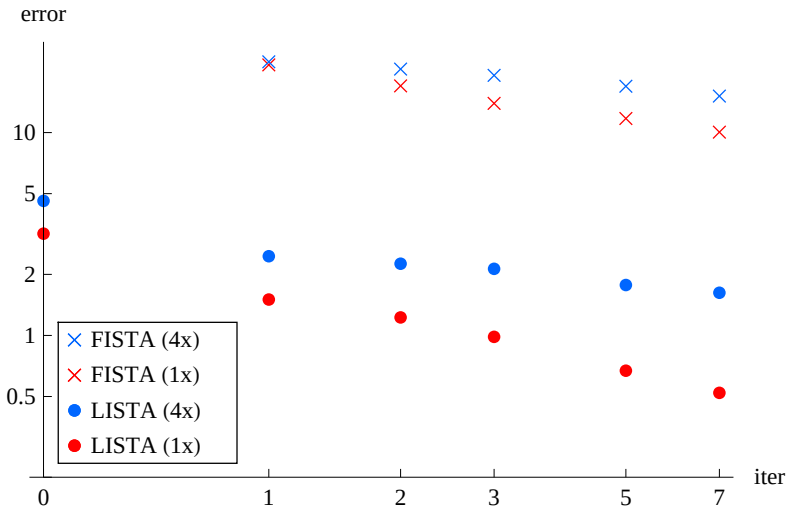
In Sparse Coding (SC), input vectors are reconstructed using a sparse linear combination of basis vectors. SC has become a popular method for extracting features from data. For a given input, SC minimizes a quadratic reconstruction error with an L_1 penalty term on the code. The process is often too slow for applications such as real-time pattern recognition. We proposed two versions of a very fast algorithm that produces approximate estimates of the sparse code that can be used to compute good visual features, or to initialize exact iterative algorithms. The main idea is to train a non-linear, feed-forward predictor with a specific architecture and a fixed depth to produce the best possible approximation of the sparse code. A version of the method, which can be seen as a trainable version of Li and Osher's coordinate descent method, is shown to produce approximate solutions with 10 times less computation than Li and Osher's for the same approximation error. Unlike previous proposals for sparse code predictors, the system allows a kind of approximate "explaining away" to take place during inference. The resulting predictor is differentiable and can be included into globally-trained recognition systems.

have been proposed to learn the dictionary. There have been applications of sparse coding in many fields including visual neuroscience (Olshausen & Field, 1996; Hoyer, 2004; Lee et al., 2007) and image restoration (Elad & Aharon, 2006; Ranzato et al., 2007b; Mairal et al., 2008). Recently, these methods have been the focus of a considerable amount of research for extracting visual features for object recognition (Ranzato et al., 2007a; Kavukcuoglu et al., 2008; Lee et al., 2009; Yang et al., 2009; Jarrett et al., 2009; Yu et al., 2009). A major problem with sparse coding for applications such as object recognition is that the inference algorithm is somewhat expensive, prohibiting real-time applications. Given an input image the inference algorithm must compute a sparse vector for each and every patch in the image (or for all local collections of low-level features, if sparse coding is used as a second stage of transformation (Yang et al., 2009)). Consequently, a large amount of research has been devoted to seeking efficient optimization algorithms for sparse coding (Daubechies et al., 2004; Lee et al., 2006; Wu & Lange, 2008; Li & Osher, 2009; Mairal et al., 2009; Beck & Teboulle, 2009; Hale et al., 2008; Vonesch & Unser, 2007).

The main contribution of this paper is a highly efficient learning-based method that computes good approximations of optimal sparse codes in a fixed amount of time. Assuming that the basis vectors of a sparse coder have been trained and are being kept fixed, the main idea of the method is to *train a parameterized non-linear "encoder" function to predict the op-*

LISTA

Results for dictionary for 10×10 sparse image patches.



Unrolling

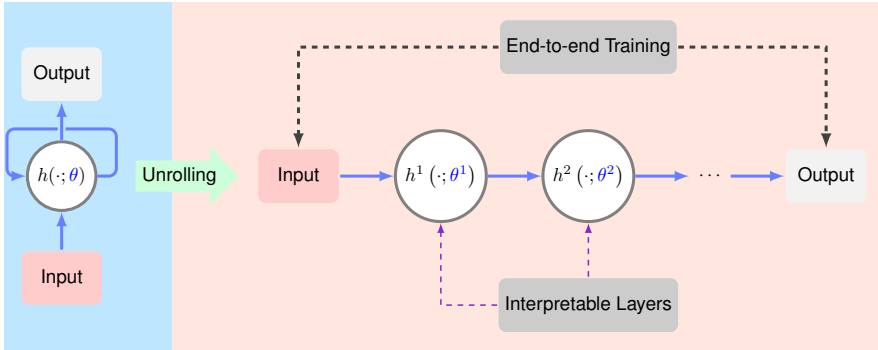


Fig. 1. A high-level overview of algorithm unrolling: given an iterative algorithm (left), a corresponding deep network (right) can be generated by cascading its iterations h. The iteration step h (left) is executed a number of times, resulting in the network layers $h^1, h^2, \ldots$ (right). Each iteration h depends on algorithm parameters θ, which are transferred into network parameters $\theta^1, \theta^2, \ldots$. Instead of determining these parameters through cross-validation or analytical derivations, we learn $\theta^1, \theta^2, \ldots$ from training datasets through end-to-end training. In this way, the resulting network could achieve better performance than the original iterative algorithm. In addition, the network layers naturally inherit interpretability from the iteration procedure. The learnable parameters are colored in blue.

Figure: Figure 1 of [Monga, Li, Eldar 2019.].

NB: In imaging, typical to use convolutional neural networks (CNNs).

Reference	Year	Application domain	Topics	Underlying Iterative Algorithms
Hershey *et al.* [30]	2014	Speech Processing	Signal channel source separation	Non-negative matrix factorization
Wang *et al.* [26]	2015	Computational imaging	Image super-resolution	Coupled sparse coding with iterative shrinkage and thresholding
Zheng *et al.* [31]	2015	Vision and Recognition	Semantic image segmentation	Conditional random field with mean-field iteration
Schuler *et al.* [32]	2016	Computational imaging	Blind image deblurring	Alternating minimization
Chen *et al.* [16]	2017	Computational imaging	Image denoising, JPEG deblocking	Nonlinear diffusion
Jin *et al.* [27]	2017	Medical Imaging	Sparse-view X-ray computed tomography	Iterative shrinkage and thresholding
Liu *et al.* [33]	2018	Vision and Recognition	Semantic image segmentation	Conditional random field with mean-field iteration
Solomon *et al.* [34]	2018	Medical imaging	Clutter suppression	Generalized ISTA for robust principal component analysis
Ding *et al.* [35]	2018	Computational imaging	Rain removal	Alternating direction method of multipliers
Wang *et al.* [36]	2018	Speech processing	Source separation	Multiple input spectrogram inversion
Adler *et al.* [37]	2018	Medical Imaging	Computational tomography	Proximal dual hybrid gradient
Wu *et al.* [38]	2018	Medical Imaging	Lung nodule detection	Proximal dual hybrid gradient
Yang *et al.* [14]	2019	Medical imaging	Medical resonance imaging, compressive imaging	Alternating direction method of multipliers
Hosseini *et al.* [39]	2019	Medical imaging	Medical resonance imaging	Proximal gradient descent
Li *et al.* [40]	2019	Computational imaging	Blind image deblurring	Half quadratic splitting
Zhang *et al.* [41]	2019	Smart power grids	Power system state estimation and forecasting	Double-loop prox-linear iterations
Zhang *et al.* [42]	2019	Computational imaging	Blind image denoising, JPEG deblocking	Moving endpoint control problem
Lohit *et al.* [43]	2019	Remote sensing	Multi-spectral image fusion	Projected gradient descent
Yoffe *et al.* [44]	2020	Medical Imaging	Super resolution microscopy	Sparsity-based super-resolution microscopy from correlation information [45]

Figure: Table 1 of [Monga, Li, Eldar 2019.].

Why unrolling?

- Fast NN approximations for sparse regularisation.
  E.g. Modern computational platforms optimised towards the relevant operations and also seek to reduce the number of iterations/layers.
- Sometimes can **carry over theory** and prove convergence results and generalisation properties. (Example - **FIRENETs** to follow)
- Can we combine best of both hand crafted priors and learning?
- Can we understand what the NN is doing? E.g. Does this help us spot limitations?
- Can be very easy to train - less parameters, use iterative method as a warm start,...

However...

- Training a fixed number of layers can incur **instability issues** mentioned above.
- **Generalisation can be worse** than classical iterative algorithms.
- **Learning can prevent convergence analysis** of iterative methods carrying over.

E.g. Current theoretical guarantees for LISTA involve parameters computed as solutions of intractably large optimisation problems and that are not used in practice. Moreover, not clear whether the needed assumptions on A hold in practice.

Current challenges: Accuracy stability trade-off and theoretical guarantees.

Tune in next time for...

We have now laid down all the groundwork for:

- Overcoming barriers: Structured sampling and achieving kernel awareness
- **FIRENETs** - neural networks based on unrolled and restarted primal-dual algorithms that are **stable** and have **exponential convergence**!
- Applications in imaging.
- Numerical examples.

DAY I	DAY II	Day III
Gravity of AI	Inverse Problems	Achieving Kernel Awareness
Image Classification	Instabilities & Kernel Awareness	FIRENETs
Need for Foundations	Intriguing Barriers	Imaging Applications
AI for Image Reconstruction	Algorithm Unrolling	Numerical Examples