Do stable networks with recovery guarantees exist?

Can compressed sensing shed light on neural networks?

Matthew Colbrook
DAMTP, University of Cambridge

Setup

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

$$y = Ax + e,$$

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

Task: reconstruct x from the noisy measurements y.

Setup

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

$$y = Ax + e,$$

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

Task: reconstruct x from the noisy measurements y.

Without additional assumptions, such as sparsity of x, this problem is highly ill-posed.

Setup

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

$$y = Ax + e,$$

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

Task: reconstruct x from the noisy measurements y.

Without additional assumptions, such as sparsity of x, this problem is highly ill-posed.

Might try to solve via a solution of

$$\min_{z \in \mathbb{C}^N} \|z\|_1 \quad \text{s.t.} \quad \|Az - y\|_2 \leq \nu,$$

or

$$\min_{z \in \mathbb{C}^N} \|z\|_1 + \|Az - y\|_2^2.$$

Setup

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

$$y = Ax + e,$$

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

Task: reconstruct x from the noisy measurements y.

Without additional assumptions, such as sparsity of x, this problem is highly ill-posed.

Might try to solve via a solution of

$$\min_{z \in \mathbb{C}^N} \|z\|_1 \quad \text{s.t.} \quad \|Az - y\|_2 \leq \nu,$$

or

$$\min_{z \in \mathbb{C}^N} \|z\|_1 + \|Az - y\|_2^2.$$

Or neural networks...

A growing problem

Most “state-of-the-art” neural networks are unstable. Now well-known for image classification:

- Universal small perturbations [Moosavi-Dezfooli et al., 2017]
- Across different networks [Szegedy et al., 2013]
- Unrecognisable images confidently classified [Nguyen et al., 2015]

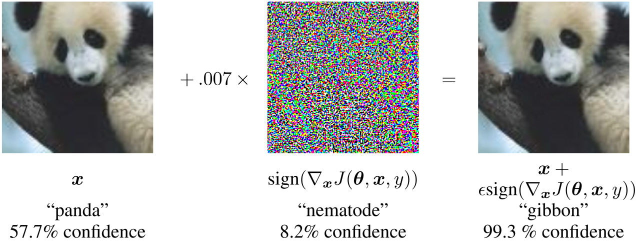

$+ .007 \times$
$=$
$\boldsymbol{x}$
“panda”
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
“nematode”
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
“gibbon”
99.3 % confidence

Figure 1: A demonstration of fast adversarial example generation applied to GoogLeNet (Szegedy et al., 2014a) on ImageNet. By adding an imperceptibly small vector whose elements are equal to the sign of the elements of the gradient of the cost function with respect to the input, we can change GoogLeNet's classification of the image. Here our ϵ of .007 corresponds to the magnitude of the smallest bit of an 8 bit image encoding after GoogLeNet's conversion to real numbers.

Figure: Source: *Explaining and harnessing adversarial examples* [Goodfellow et al., 2014].

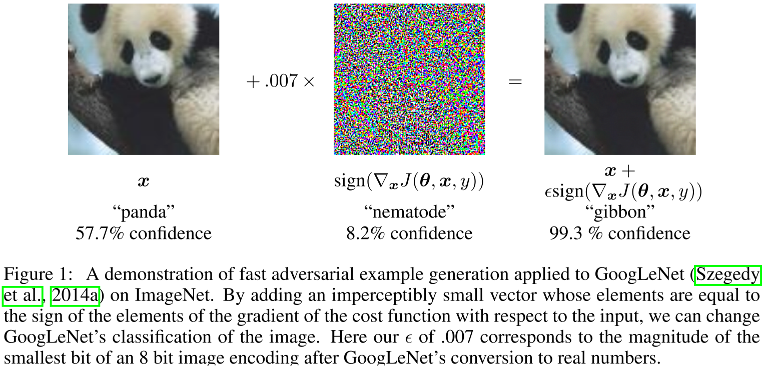

$+ .007 \times$
$=$
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Figure 1: A demonstration of fast adversarial example generation applied to GoogLeNet (Szegedy et al., 2014a) on ImageNet. By adding an imperceptibly small vector whose elements are equal to the sign of the elements of the gradient of the cost function with respect to the input, we can change GoogLeNet's classification of the image. Here our ϵ of .007 corresponds to the magnitude of the smallest bit of an 8 bit image encoding after GoogLeNet's conversion to real numbers.

Figure: Source: *Explaining and harnessing adversarial examples* [Goodfellow et al., 2014].

BUT can also happen with image denoising/reconstruction...

A stability test (see [Antun et al., 2019])

Consider a neural network (or in general a map) $\phi : \mathbb{C}^m \to \mathbb{C}^N$ which aims to reconstruct the image $\phi(y) \approx x$ from the (noisy) measurements $y = Ax + e$.

A stability test (see [Antun et al., 2019])

Consider a neural network (or in general a map) $\phi : \mathbb{C}^m \to \mathbb{C}^N$ which aims to reconstruct the image $\phi(y) \approx x$ from the (noisy) measurements $y = Ax + e$.

Algorithm seeks a vector $r \in \mathbb{R}^N$ such that

$$\|\phi(y + Ar) - \phi(y)\|_2 \text{ is large, while } \|r\|_2 \text{ is small.}$$

A stability test (see [Antun et al., 2019])

Consider a neural network (or in general a map) $\phi : \mathbb{C}^m \to \mathbb{C}^N$ which aims to reconstruct the image $\phi(y) \approx x$ from the (noisy) measurements $y = Ax + e$.

Algorithm seeks a vector $r \in \mathbb{R}^N$ such that

$$\|\phi(y + Ar) - \phi(y)\|_2 \text{ is large, while } \|r\|_2 \text{ is small.}$$

Consider the optimisation problem

$$r^*(y) \in \operatorname*{argmax}_r \frac{1}{2}\|\phi(y + Ar) - x\|_2^2 - \frac{\lambda}{2}\|r\|_2^2.$$

A stability test (see [Antun et al., 2019])

Consider a neural network (or in general a map) $\phi : \mathbb{C}^m \to \mathbb{C}^N$ which aims to reconstruct the image $\phi(y) \approx x$ from the (noisy) measurements $y = Ax + e$.

Algorithm seeks a vector $r \in \mathbb{R}^N$ such that

$$\|\phi(y + Ar) - \phi(y)\|_2 \text{ is large, while } \|r\|_2 \text{ is small.}$$

Consider the optimisation problem

$$r^*(y) \in \operatorname*{argmax}_r \frac{1}{2}\|\phi(y + Ar) - x\|_2^2 - \frac{\lambda}{2}\|r\|_2^2.$$

Test aims to locate local maxima by using a gradient ascent with momentum on

$$Q_y^\phi(r) = \frac{1}{2}\|\phi(y + Ar) - x\|_2^2 - \frac{\lambda}{2}\|r\|_2^2$$

Example

Simple example for the AUTOMAP network, reported in *Nature* as a “state-of-the-art” network:

“Furthermore, AUTOMAP reconstructions exhibit superior noise immunity compared to those from conventional methods, as quantified by image signal-to-noise ratio and root-mean-squared error (RMSE) metrics.”

Example

Simple example for the AUTOMAP network, reported in *Nature* as a “state-of-the-art” network:

“Furthermore, AUTOMAP reconstructions exhibit superior noise immunity compared to those from conventional methods, as quantified by image signal-to-noise ratio and root-mean-squared error (RMSE) metrics.”

Do we believe this?

Example

Simple example for the AUTOMAP network, reported in *Nature* as a “state-of-the-art” network:

Not so state-of-the-art in terms of stability...

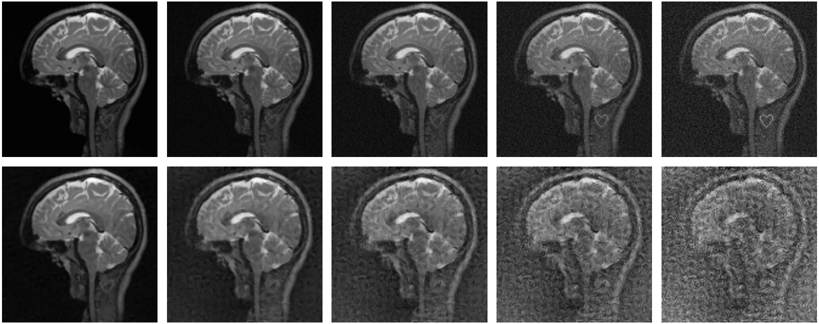


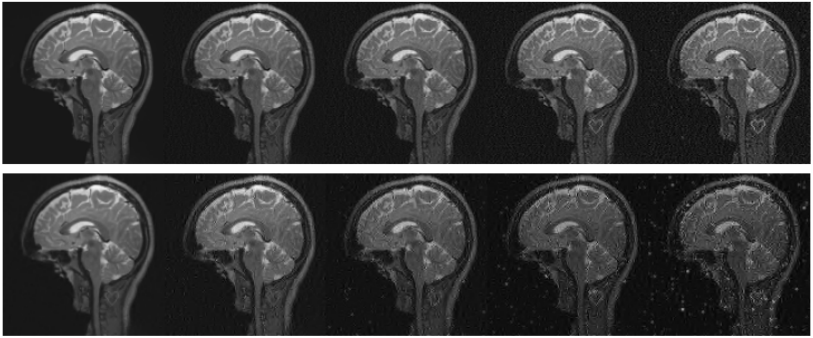
Figure: Stability test for AUTOMAP taken from [Antun et al., 2019], and where A is a subsampled Fourier transform. Top row: original image with perturbations. Bottom row: reconstructions using AUTOMAP.

Question: Can stability be hard-wired into the networks or, more generally, methods of reconstruction at all?

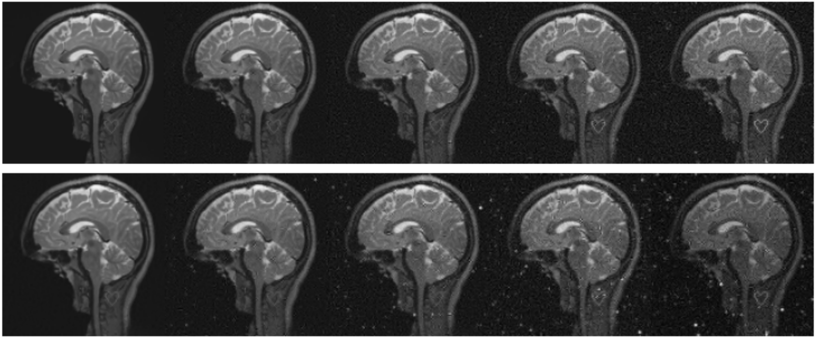
Question: Can stability be hard-wired into the networks or, more generally, methods of reconstruction at all?

This is subtle, preliminary results suggest that this can't be done by simply unfolding FISTA on LASSO or Chambolle-Pock on basis pursuit...

Solving LASSO with FISTA



Solving basis pursuit with Chambolle-Pock



Neural networks are FANTASTIC approximators!

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

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

given by

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

Neural networks are FANTASTIC approximators!

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

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

given by

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

Theorem ([Pinkus, 1999])

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

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

Also true for single hidden layer networks.

Neural networks are FANTASTIC approximators!

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

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

given by

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

Theorem ([Pinkus, 1999])

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

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

Also true for single hidden layer networks.
But: need a <u>constructive</u> 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. Also models a type of numerical stability.

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. Also models a type of numerical stability.

Training set must be

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

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. Also models a type of numerical stability.

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 $\mathcal{T}$?

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. Also models a type of numerical stability.

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 $\mathcal{T}$?

Again, maybe we expect to be able to do this by unfolding standard (iterative) optimisation algorithms? Like ISTA, FISTA, NESTA,...

Theorem (Impossible in general)

Let $K > 2, L \in \mathbb{N}$ and d be any norm on $\mathbb{C}^N$ where $N \geq 2$. 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 with $\mathcal{T}$ as input that produces a neural network Ψ that approximates Φ on $(A, \mathcal{M}) \in \Omega$ to K correct digits in the norm d.*

(ii) *There exists an algorithm with $\mathcal{T}$ as input that produces a neural network Ψ that approximates Φ on $(A, \mathcal{M}) \in \Omega$ to $K - 1$ correct digits in the norm d. However, any algorithm producing such a network will need arbitrary many samples of elements from $\mathcal{T}$.*

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

It is NOT enough to just “unfold” your favourite algorithm. Unsurprisingly the theorem tells us extra assumptions need to be made on the problem at hand.

It is NOT enough to just “unfold” your favourite algorithm. Unsurprisingly the theorem tells us extra assumptions need to be made on the problem at hand.

Theorem also holds for other popular optimisation problems such as LASSO.

It is NOT enough to just “unfold” your favourite algorithm. Unsurprisingly the theorem tells us extra assumptions need to be made on the problem at hand.

Theorem also holds for other popular optimisation problems such as LASSO.

NB: Compressed sensing type results always assume we can compute the minimiser. Convergence guarantees on iterative algorithms tend to be given in terms of the objective function instead.

It is NOT enough to just “unfold” your favourite algorithm. Unsurprisingly the theorem tells us extra assumptions need to be made on the problem at hand.

Theorem also holds for other popular optimisation problems such as LASSO.

NB: Compressed sensing type results always assume we can compute the minimiser. Convergence guarantees on iterative algorithms tend to be given in terms of the objective function instead.

Questions: Can we solve this type of problem in a stable and constructive manner? What assumptions do we need?

A result saying when we can...

Definition (Sparsity in levels)

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

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

We denote the set of $(\mathbf{s}, \mathbf{M})$*-sparse vectors by* $\Sigma_{\mathbf{s},\mathbf{M}}$*.*

$$\|x\|_{l^1_w} = \sum_{i=1}^{N} w_i \left| x_i \right|,$$

$$\sigma_{\mathbf{s},\mathbf{M}}(x)_{l^1_w} = \inf\{\|x - z\|_{l^1_w} : z \in \Sigma_{\mathbf{s},\mathbf{M}}\}.$$

Why is this reasonable?

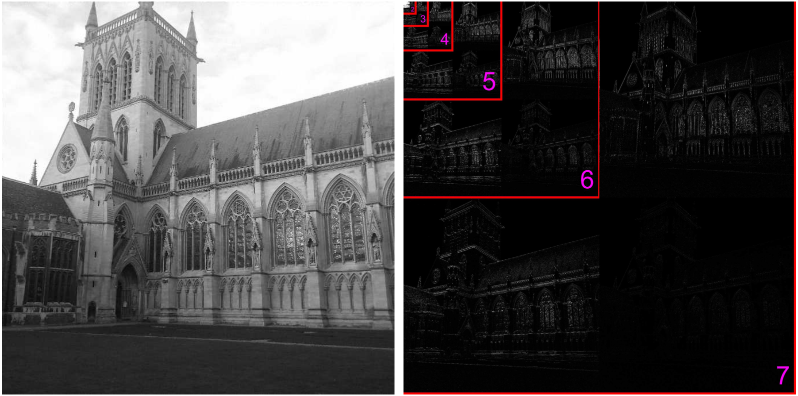


Figure: An image and its wavelet coefficients, where a brighter colour corresponds to a larger value.

In practice, if W denotes wavelet transform, expect $\sigma_{\mathbf{s},\mathbf{M}}(Wx)_{l^1_w}$ to be small if we use wavelet levels.

Definition (Robust nullspace property in levels)

We say that A has the weighted robust nullspace property in levels (w-RNSPL) of order $(\mathbf{s}, \mathbf{M})$ if there exists $\delta \in (0,1)$ and $\tau > 0$ such that for any $(\mathbf{s}, \mathbf{M})$-sparse set S and $x \in \mathbb{C}^N$,

$$\|x_S\|_{l^2} \leq \frac{\delta}{\sqrt{\sum_i s_i}} \|x_{S^c}\|_{l^1_w} + \tau \|Ax\|_{l^2}.$$

Some final quantities...

Assume that if $M_{j-1}+1 \leq i \leq M_j$ then $w_i = w_{(j)}$ (i.e. constant in each level).

$$\lambda(\mathbf{w},\mathbf{s}) = \frac{\sum_{j=1}^{r} s_j w_{(j)}^2}{\min_{j=1,\ldots,r} s_j w_{(j)}^2}, \quad \eta(\mathbf{w},\mathbf{s}) = \sum_{j=1}^{r} s_j w_{(j)}^2.$$

Theorem (Stable Methods Exist)

Suppose that A has the w-RNSPL of order $(\mathbf{s}, \mathbf{M})$. Then, there exists a (computable[1]) iterative algorithm ϕ_n^A (that can be unfolded as a neural network with $3n$ layers) such that the following uniform recovery guarantee holds. For any $x \in \mathbb{C}^N$ with $\|x\|_{l^2} \lesssim 1$ and any $y \in \mathbb{C}^m$,

$$\|\phi_n^A(y) - x\|_{l^2} \lesssim \frac{\lambda(\mathbf{w}, \mathbf{s})^{\frac{1}{4}}}{\sqrt{\eta(\mathbf{w}, \mathbf{s})}} \sigma_{\mathbf{s},\mathbf{M}}(x)_{l^1_w} + \frac{\lambda(\mathbf{w}, \mathbf{s})^{\frac{1}{4}} \|A\|}{n} + \lambda(\mathbf{w}, \mathbf{s})^{\frac{1}{4}} \|Ax - y\|_{l^2}.$$

For large n, well behaved (effectively small local Lipschitz constant) near manifold of sparse vectors.

[1] Discussion of computability beyond scope of talk, but means I can use the setup $\mathcal{T}$, and everything can be made to work on rationals $\mathbb{Q}$ etc.

Example of when this occurs

Sparsifying transform: Haar wavelets with matrix W. Take sparsity in levels to correspond to wavelet levels.

Example of when this occurs

Sparsifying transform: Haar wavelets with matrix W. Take sparsity in levels to correspond to wavelet levels.

Definition (Multilevel random sampling)

Let $l \in \mathbb{N}, \mathbf{N} = (N_1, \ldots, N_l) \in \mathbb{N}^l$ *with* $1 \leq N_1 < \ldots < N_l$, $\mathbf{m} = (m_1, \ldots, m_l) \in \mathbb{N}^l$, *with* $m_k \leq N_k - N_{k-1}$, $k = 1, \ldots, l$, *and suppose that*

$$\Omega_k \subset \{N_{k-1} + 1, \ldots, N_k\}, \ |\Omega_k| = m_k, \quad k = 1, \ldots, l,$$

are chosen uniformly at random, where $N_0 = 0$. *We refer to the set* $\Omega = \Omega_{\mathbf{N},\mathbf{m}} = \Omega_1 \cup \ldots \cup \Omega_l$ *as an* $(\mathbf{N}, \mathbf{m})$*- multilevel sampling scheme.*

Case 1: Fourier measurements

U corresponds to the d-dimensional discrete Fourier transform.

Case 1: Fourier measurements

U corresponds to the d-dimensional discrete Fourier transform.

We divide the different frequencies into dyadic bands B_k, where $B_1 = \{0, 1\}$ and for $k = 2, ..., r$

$$B_k = \{-2^{k-1} + 1, ..., -2^{k-2}\} \cup \{2^{k-2} + 1, ..., 2^{k-1}\}.$$

Case 1: Fourier measurements

U corresponds to the d-dimensional discrete Fourier transform.

We divide the different frequencies into dyadic bands B_k, where $B_1 = \{0, 1\}$ and for $k = 2, ..., r$

$$B_k = \{-2^{k-1} + 1, ..., -2^{k-2}\} \cup \{2^{k-2} + 1, ..., 2^{k-1}\}.$$

In d dimensions set

$$B_{\mathbf{k}}^{(d)} = B_{k_1} \times ... \times B_{k_d}, \quad \mathbf{k} = (k_1, ..., k_d) \in \mathbb{N}^d.$$

Multilevel random sampling with $(m_{\mathbf{k}=(k_1,...,k_d)})_{k_1,...,k_d=1}^{r}$, $|m_{\mathbf{k}}| \leq \left|B_{\mathbf{k}}^{(d)}\right|$.

Case 1: Fourier measurements

Take measurement A to be subsampled U, change basis to AW^* so results stated in terms of $\sigma_{\mathbf{s},\mathbf{M}}(Wx)_{l^1_w}$ which we expect to be small.

Case 1: Fourier measurements

Take measurement A to be subsampled U, change basis to AW^* so results stated in terms of $\sigma_{\mathbf{s},\mathbf{M}}(Wx)_{l^1_w}$ which we expect to be small.

One horrible looking formula...

$$\mathcal{M}_{\mathcal{F}}(\mathbf{s},\mathbf{k}) = \sum_{l=1}^{\|\mathbf{k}\|_\infty} s_l \prod_{i=1}^{d} 2^{-|k_i-l|} + \sum_{l=\|\mathbf{k}\|_\infty+1}^{r} s_l 2^{-2(l-\|\mathbf{k}\|_\infty)} \prod_{i=1}^{d} 2^{-|k_i-l|}.$$

Case 1: Fourier measurements

Take measurement A to be subsampled U, change basis to AW^* so results stated in terms of $\sigma_{\mathbf{s},\mathbf{M}}(Wx)_{l^1_w}$ which we expect to be small.

One horrible looking formula...

$$\mathcal{M}_{\mathcal{F}}(\mathbf{s},\mathbf{k}) = \sum_{l=1}^{\|\mathbf{k}\|_\infty} s_l \prod_{i=1}^{d} 2^{-|k_i-l|} + \sum_{l=\|\mathbf{k}\|_\infty+1}^{r} s_l 2^{-2(l-\|\mathbf{k}\|_\infty)} \prod_{i=1}^{d} 2^{-|k_i-l|}.$$

Let $\epsilon_{\mathbb{P}} \in (0,1)$, $r, d \in \mathbb{N}$, $N = 2^{r\cdot d}$ and suppose

$$\begin{aligned} m_{\mathbf{k}} &\gtrsim \mathcal{M}_{\mathcal{F}}(\mathbf{s},\mathbf{k}) \cdot L, \\ L &= d \cdot r^2 \cdot \log(m) \cdot \log^2(s\lambda(\mathbf{w},\mathbf{s})) + \log(\epsilon_{\mathbb{P}}^{-1}). \end{aligned}$$

Then conditions of theorem met with probability at least $1-\epsilon_{\mathbb{P}}$.

How to interpret?

- Up to log-factors, equivalent to oracle estimator (as $n \to \infty$).
- Number of samples required in each annular region

$$\sum_{\|\mathbf{k}\|=k} m_{\mathbf{k}} \gtrsim \left(s_k + \sum_{l=1}^{k-1} s_l 2^{-(k-l)} + \sum_{l=k+1}^{r} s_l 2^{-3(l-k)} \right) \cdot L.$$

is (up to logarithmic factors) proportional to s_k + exponentially decaying terms.

Case 2: Binary measurements

U corresponds to Walsh-Hadamard transform with tensor product basis.

Case 2: Binary measurements

U corresponds to Walsh-Hadamard transform with tensor product basis.

$$\mathcal{M}_{\mathcal{B}}(\mathbf{s}, \mathbf{k}) = s_{\|\mathbf{k}\|_\infty} \prod_{i=1}^{d} 2^{-|k_i - \|\mathbf{k}\|_\infty|}$$

Case 2: Binary measurements

U corresponds to Walsh-Hadamard transform with tensor product basis.

$$\mathcal{M}_{\mathcal{B}}(\mathbf{s}, \mathbf{k}) = s_{\|\mathbf{k}\|_\infty} \prod_{i=1}^{d} 2^{-|k_i - \|\mathbf{k}\|_\infty|}$$

Theorem then the same but now

$$\sum_{\|\mathbf{k}\|=k} m_{\mathbf{k}} \gtrsim 2^d d s_k L,$$

and there are no terms from the sparsity levels $s_l, l \neq k$.

Numerical Example

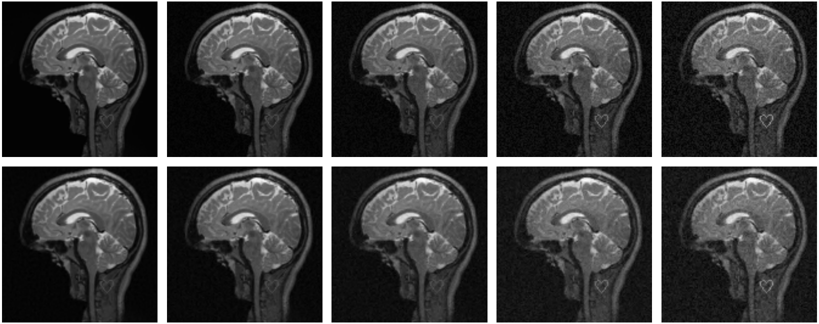


Figure: Stability test for new networks. Top row: original image with perturbations. Bottom row: reconstructions.

Conclusions

- Given the last fifty years of inverse problems, stability should **not be overlooked**.
- There is likely a rich classification theory, stating limits on the performance of stable methods - trade-off.
- One such example was presented with explicitly constructed stable neural networks.
- Further study: can we use these ideas in trained models?

Antun, V., Renna, F., Poon, C., Adcock, B., and Hansen, A. C. (2019).
On instabilities of deep learning in image reconstruction - Does AI come at a cost?
Submitted.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014).
Explaining and harnessing adversarial examples.
arXiv preprint arXiv:1412.6572.

Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., and Frossard, P. (2017).
Universal adversarial perturbations.
In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1765–1773.

Nguyen, A., Yosinski, J., and Clune, J. (2015).
Deep neural networks are easily fooled: High confidence predictions for unrecognizable images.
In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 427–436.

Pinkus, A. (1999).
Approximation theory of the mlp model in neural networks.
Acta numerica, 8:143–195.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013).
Intriguing properties of neural networks.
arXiv preprint arXiv:1312.6199.