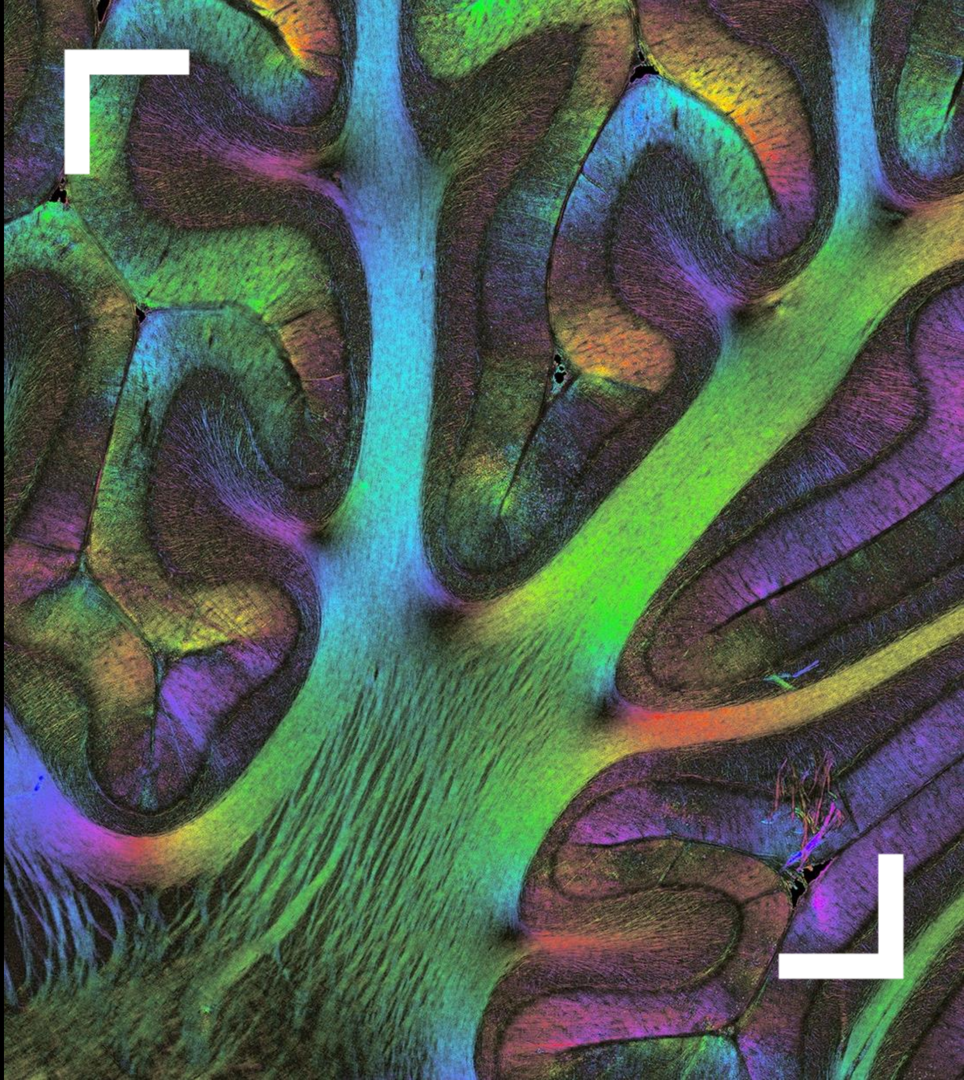
Uncertainty-awareness and energy-efficency in Deep Learning

Martin Burger
Helmholtz Imaging, FS-CI, DESY

Deep Learning Summit
Hamburg, 08.12.2023

HELMHOLTZ IMAGING

DESY.



Deep Learning – A Math Perspective

Why Deep Learning ?

Deep learning is ubiquituos, some people even think it is AI*

Why are we using deep learning, deep neural networks ?

What are the theoretical foundations ?

*Remember the I is for intelligence

Deep Learning Architectures: High Dimensional Function Approximation

Universal approximation property

Famous results due to Cybenko 1989 / Hornik, Stinchcombe / White 1989:

Every continuous function can be approximated arbitrarily well with deep (even with shallow) networks*

Even better: dimension-independence

Barron 1992 / 1995 shows that functions in arbitrary dimension can be approximated with rate $n^{-1/2}$, where n is the number of parameters**

*The proof is actually a quite simple application of the Stone-Weierstrass Theorem

**Ok, this was cheated, there was a hidden assumption that implies the admissible class depends on dimension

Deep Learning Architectures: High Dimensional Function Approximation

Universal approximation property

Famous results due to Cybenko 1989 / Hornik, Stinchcombe / White 1989:

Every continuous function can be approximated arbitrarily well with deep (even with shallow) networks*

Even better: dimension-independence

Barron 1992 / 1995 shows that functions in arbitrary dimension can be approximated with rate $n^{-1/2}$, where n is the number of parameters**

*The proof is actually a quite simple application of the Stone-Weierstrass Theorem

**Ok, this was cheated, there was a hidden assumption that implies the admissible class depends on dimension

Deep Learning Architectures: High Dimensional Function Approximation

What nobody tells you

Universal approximation by many systems

Even dimension-independent bound generic for nonlinear approximation

So why do we really use deep neural networks ?

Deep Learning on GPUs

Why do we use deep neural networks ?

Because they fit to NVIDIA GPU architectures

Because Google and Facebook implemented them* and had the amount of data to

Should we be concerned ?**

*TensorFlow / Pytorch

**If Google would optimize everything for diffusion equations, would we simulate waves with diffusion equations ?

Deep Learning

What else to be concerned

Uncertainties: deep learning performs statistical computations without error bars

Deep learning can be fooled easily

Deep learning often based on brute-force computing, horrible carbon and water footprint

Extracting Training Data from ChatGPT

AUTHORS

Milad Nasr[*1], Nicholas Carlini[*1], Jon Hayase[1,2], Matthew Jagielski[1], A. Feder Cooper[3], Daphne Ippolito[1,4], Christopher A. Choquette-Choo[1], Eric Wallace[5], Florian Tramèr[6], Katherine Lee[+1,3]

[1]Google DeepMind, [2] University of Washington, [3]Cornell, [4]CMU, [5]UC Berkeley, [6]ETH Zurich. * Joint first author, [+]Senior author.

PUBLISHED

November 28, 2023

READ:

[arxiv]

Uncertainties in Deep Learning

Supervised learning

Standard formulation: empirical risk minimization

$$\min_{\theta} \frac{1}{N} \sum \mathbf{loss}(f_\theta(x_i), y_i)$$

Train parameters to obtain a network *f* that approximates output *y* given input *x*

Uncertainties in Deep Learning

Training Data Uncertainty

$$\min_\theta \frac{1}{N} \sum \mathsf{loss}(f_\theta(x_i), y_i)$$

Standard question: generalization
(from finite number of training data to full population)

Standard results: statistical estimates on generalization error / population loss

$$\mathbb{E}_{(x,y)\sim\mu}(\mathbf{loss}(f_\theta(x), y))$$

Under assumption that training data are i.i.d. sampled

Note: even exact knowledge of expected loss does not tell you much about single data point

Uncertainties in Deep Learning

Training Data Uncertainty

$$\min_{\theta} \frac{1}{N} \sum \mathsf{loss}(f_\theta(x_i), y_i)$$

$$\mathbb{E}_{(x,y)\sim\mu}(\mathbf{loss}(f_\theta(x), y))$$

Further issues: distribution shifts

Training data not sampled from μ

Typical example: simulated data, e.g. in inverse problems
Model errors, wrong noise statistics, missing ground truth data

Even if sampled from μ, not i.i.d. or not from whole population

Learning in Image Reconstruction

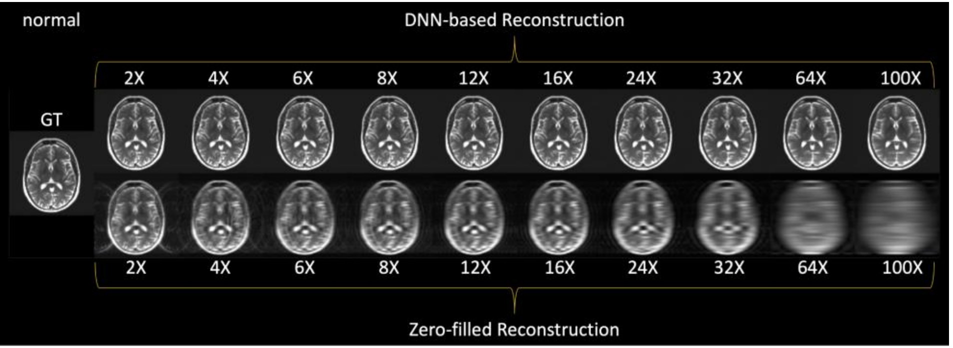
Undersampled MRI

Undersampling in MRI is attractive since it does not

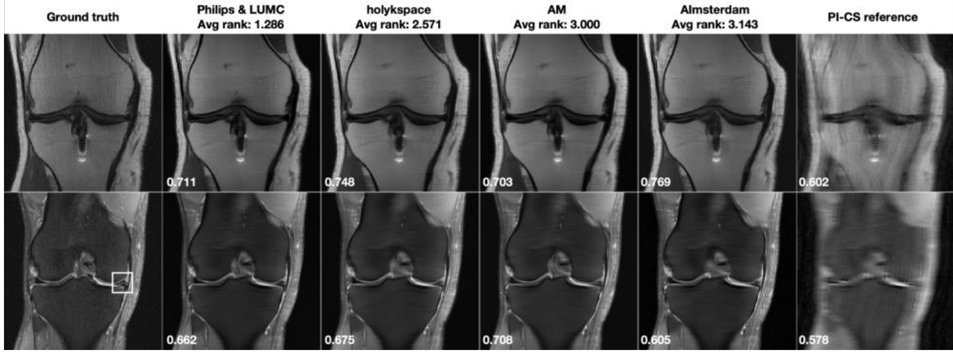
Lower complexity, since forward operator just Fourier transform, low noise

Isometry property of Fourier transform leads to low Lipschitz constant of inverse

Data pairs from existing fully sampled measurements and reconstructions



Radmanesh Radiology AI 2022



Knoll MRM 2019 / 2020 (fast MRI challenge)

Learning in Image Reconstruction

Undersampled MRI

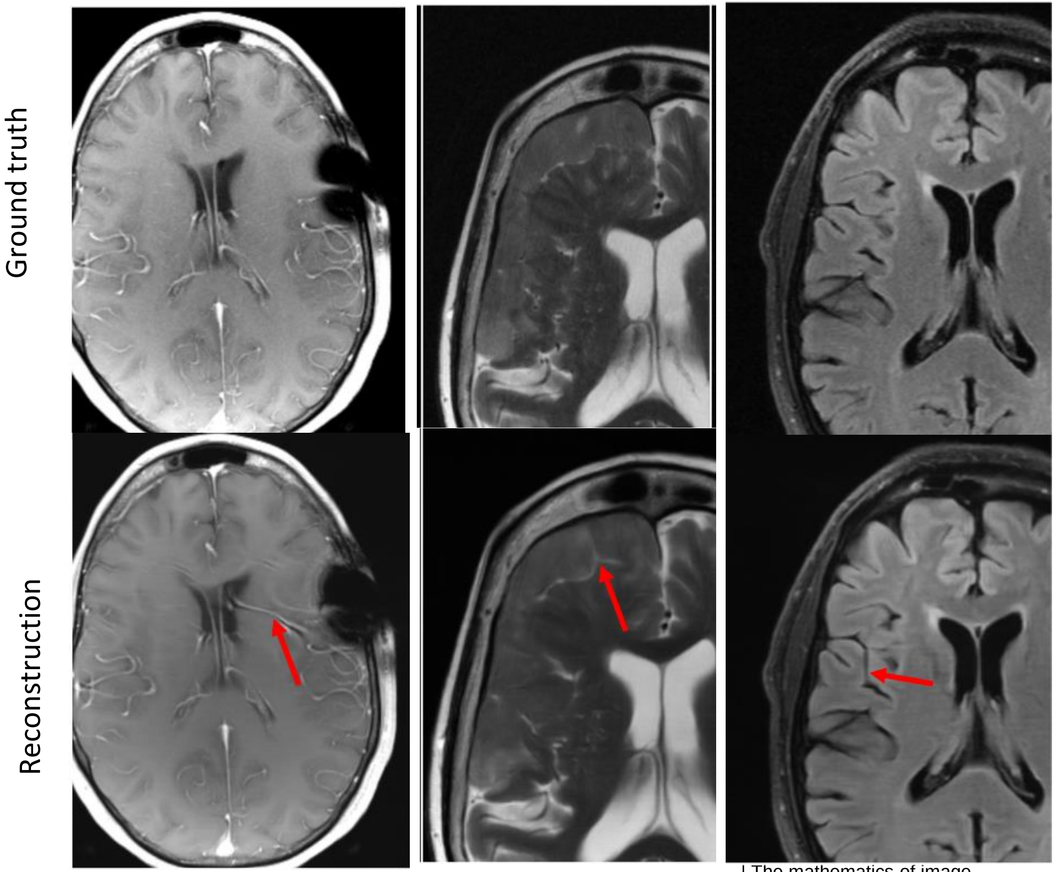
Majority of results convincing

But possible hallucinations
on few data sets

Not recognizable by
experienced radiologists

(courtesy Florian Knoll, Erlangen)

Muckley TMI 2021



Uncertainties in Deep Learning

Single Data Uncertainty

Even for a well trained model, there may be inputs *x* (our outputs *y*)

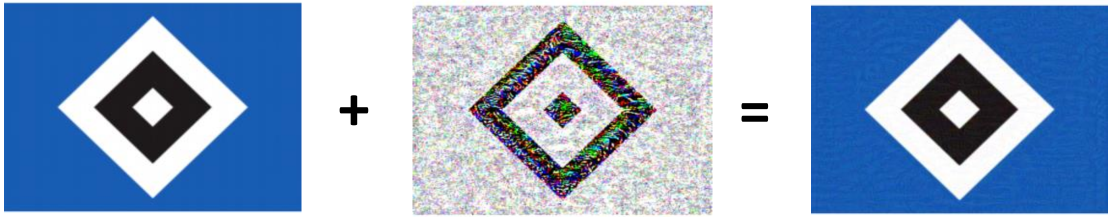
Basic smoothness assumption

$$f(x) \approx f(\tilde{x}) \text{ for all } \tilde{x} \text{ in the vicinity } B_{\|\cdot\| \leq \epsilon}(x)$$

may be violated

Adversarial Attacks

Even the worst can happen



HSV **δ = 0.02** **St. Pauli**

Adversarial Attacks

Carefully manipulate the input data

Ideal attack around a given point

$$\arg \max_{x:\|x-x_0\|\leq\epsilon} \underbrace{l(f_\theta(x), y_{x_0})}_{:=L(x,y)}$$

Difficult to compute, approximate to first order

$$l(f_\theta(x), y_{x_0}) \approx l(f_\theta(x_0), y_{x_0}) + \langle x - x_0, \nabla_x l(f_\theta(x_0), y_{x_0})\rangle$$

Thus, for small perturbations we can maximize

$$\max_{x:\|x-x_0\|\leq\epsilon} \langle x - x_0, \nabla_x l(f_\theta(x), y_{x_0})\rangle = \max_{x:\|x-x_0\|\leq 1} \left\langle \frac{x - x_0}{\epsilon}, \nabla_x l(f_\theta(x_0), y_{x_0}) \right\rangle$$

Adversarial Attacks

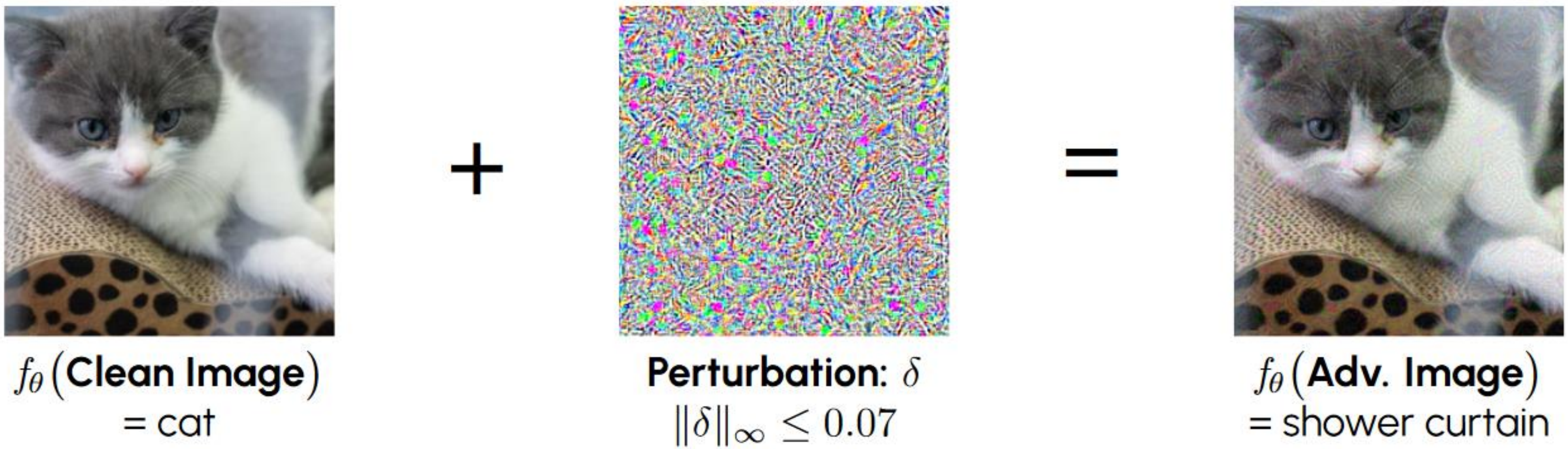
Different ways to compute attacks

Fast gradient methods

	FGM (Fast gradient method)	FGSM (Fast gradient sign method)
Maximization Problem	$\arg\max_{x:\|x-x_0\|_2\leq\epsilon} L(x,y)$	$\arg\max_{x:\|x-x_0\|_\infty\leq\epsilon} L(x,y)$
Scheme	$x = x_0 + \epsilon\frac{\nabla L(x,y)}{\|\nabla L(x,y)\|_2}$	$x = x_0 + \epsilon\,\mathrm{sign}(\nabla L(x,y))$
ODE	$\partial_t x(t) = \frac{\nabla L(x,y)}{\|\nabla L(x,y)\|_2}$	$\partial_t x(t) \in \mathrm{sign}(\nabla L(x,y))$

Adversarial Attacks

Fooling AI by noise



Robust training

Adversarial defense

Incorporate possible changes already in the training loss

$$\frac{1}{N}\sum_{i=1}^{N}\mathbf{loss}_{\epsilon}(f_{\theta}(x_i),y_i)=\frac{1}{N}\sum_{i=1}^{N}\min_{\xi,\|\xi-x_i\|\leq\epsilon}\mathbf{loss}(f_{\theta}(\xi),y_i)$$

Alternatives: stochastic formulations, expectation over local perturbations instead of minimization

Robust training

Total variation regularization

$$\min_{\theta} \int \max_{w:\|w-x\|\leq\epsilon} \textsf{loss}(f_\theta(w), y)\, d\mu$$

$$= \min_{\theta} \int \textsf{loss}(f_\theta(x), y)\, d\mu + \epsilon \int \frac{\max_{w:\|w-x\|\leq\epsilon} \textsf{loss}(f_\theta(w), y) - \textsf{loss}(f_\theta(x), y)}{\epsilon}\, d\mu$$

$$\approx \min_{\theta} \int \textsf{loss}(f_\theta(x), y)\, d\mu + \epsilon \int |\partial_x \textsf{loss}(f_\theta(x), y)|\, d\mu$$

Robust training

Distributional adversaries

Generalize from changes of inputs to full distribution

$$\min_{\theta} \max_{\mu : G(\mu, \mu_0) \leq \epsilon} \mathbb{E}_{(x,y)\sim\mu} L(x, y)$$

G suitable distance on distributions, e.g. Wasserstein metric

For certain Wasserstein metrics equivalence to pointwise formulation

Uncertainties in Deep Learning

Generative models

Diffusion models, normalizing flows, GANs …

Basic structure: map an empirical data distribution into a distribution that can be sampled (neural network parametrizes the map / transport)

$$(x_i)_{i=1,\ldots,N} \sim \mu \quad \rightarrow \quad \rho \approx \mu$$

Similar issues: generalization, distribution shifts

Further issue: reconstruction of training data (privacy etc)

Uncertainties in Deep Learning

Scientific data / questions

Importance of adversarial attacks / robustness demonstrated in many data sets

Unclear: use in scientific cases

Can measurement noise / errors act like adversaries ?
Which robustness to enforce ?

Open to your Input !!

Efficiency in Deep Learning

Options

Avoid overuse of deep learning (remember old knowledge / marry with model-based approaches)

New methods that can deal with less training data

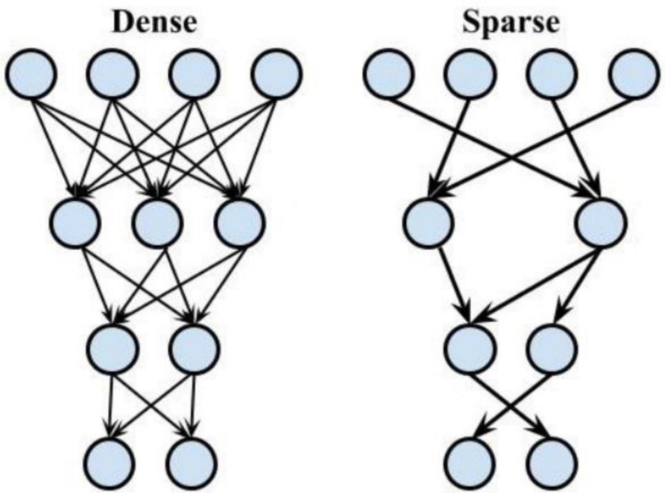
More efficient / sparse network structures

Novel computing architectures (quantum / neural …)

Sparse Neural Networks

Lottery ticket hypothesis*

Each dense neural network contains an equally performing sparse subnetwork



How to find sparse neural networks ?

- dense-to-sparse training (pruning)
- sparse-to-sparse training
- sparse training with a possibly hidden dense structure

*Frankle et al 2018

Inverse Scale Space / Bregman Iterations

Idea from Image Reconstruction and Analysis

Use variational regularization to define scale

Perform an iteration / flow that iteratively adds finer and finer scales

Image reconstruction: scale related to geometric objects

Regularization = total variation

Interpretation as length of edge sets

Scale: boundary length vs. volume

MB-Gilboa-Osher-Xu-Yin 2005, Benning-MB 2018

Inverse Scale Space / Bregman Iterations

Idea from Image Reconstruction and Analysis

Use variational regularization to define scale

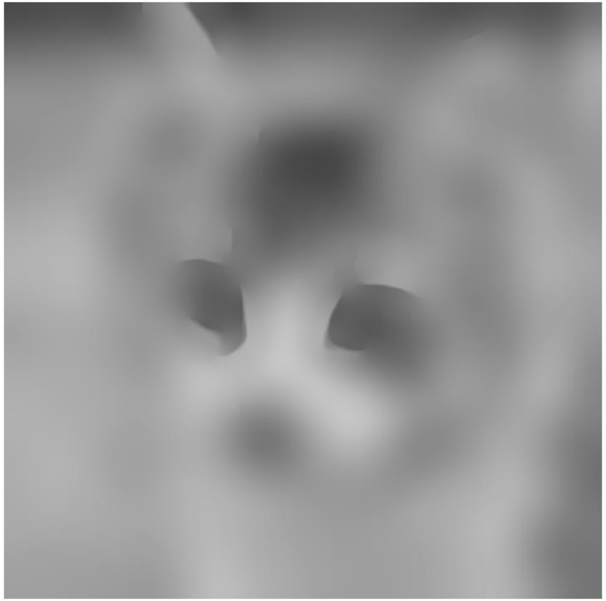
Perform an iteration / flow that iteratively adds finer and finer scales

Image reconstruction: scale related to geometric objects

Regularization = total variation

Interpretation as length of edge sets

Scale: boundary length vs. volume



Inverse Scale Space / Bregman Iterations

Idea from Image Reconstruction and Analysis

Use variational regularization to define scale

Perform an iteration / flow that iteratively adds finer and finer scales

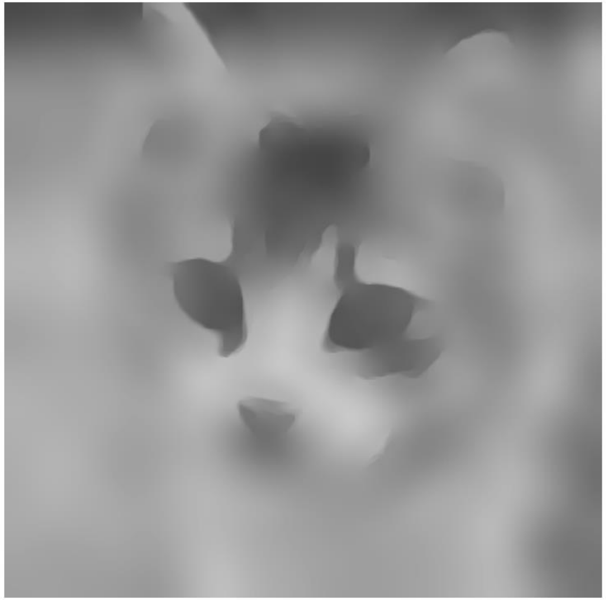


Image reconstruction: scale related to geometric objects

Regularization = total variation

Interpretation as length of edge sets

Scale: boundary length vs. volume

Inverse Scale Space / Bregman Iterations

Idea from Image Reconstruction and Analysis

Use variational regularization to define scale

Perform an iteration / flow that iteratively adds finer and finer scales

Image reconstruction: scale related to geometric objects

Regularization = total variation

Interpretation as length of edge sets

Scale: boundary length vs. volume

Inverse Scale Space / Bregman Iterations

Idea from Image Reconstruction and Analysis

Use variational regularization to define scale

Perform an iteration / flow that iteratively adds finer and finer scales

Image reconstruction: scale related to geometric objects

Regularization = total variation

Interpretation as length of edge sets

Scale: boundary length vs. volume

Inverse Scale Space / Bregman Iterations

Idea from Image Reconstruction and Analysis

Use variational regularization to define scale

Perform an iteration / flow that iteratively adds finer and finer scales

Image reconstruction: scale related to geometric objects

Regularization = total variation

Interpretation as length of edge sets

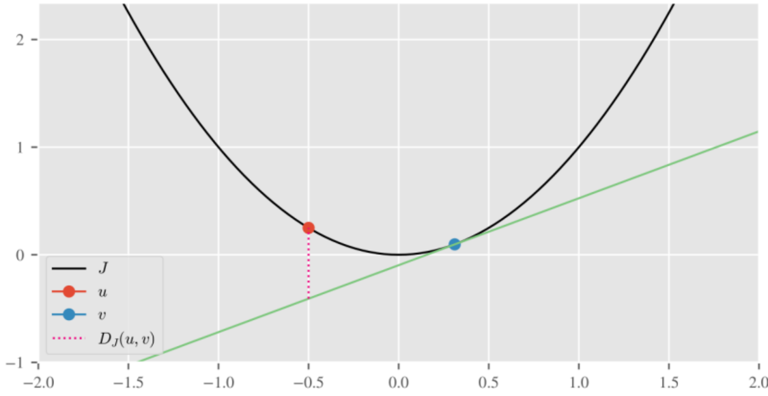
Scale: boundary length vs. volume

Enforcing Sparsity: 1- norm

Basic iteration scheme for loss (gradient descent when Euclidean distance is replaced by Bregman distance)

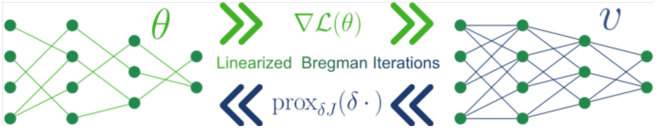
$$D_J^p(u,v) := J(u) - J(v) - \langle p, u-v\rangle, \quad p \in \partial J(v).$$

$$\begin{aligned} v^{k+1} &= v^k - \tau^k \nabla \mathcal{L}(\theta^k), \\ \theta^{k+1} &= \mathrm{prox}_{\delta J}(\delta v^k), \end{aligned}$$



$$\mathrm{prox}_{\delta J}(v) = \arg\min_w \left(\frac{1}{2}\|w-v\|^2 + \delta J(w) \right)$$

Choose $J(\theta) = \|\theta\|_1$ as 1-norm (or group 1-norm).
Choose $\mathcal{L}$ as empirical risk on training data.



LinBreg Algorithm

We replace the empirical risk $\mathcal{L}$ on a training set $\mathcal{T}$ by

$$L(\theta; B) := \frac{1}{|B|} \sum_{(x,y)\in B} \ell(f_\theta(x), y), \quad B \subset \mathcal{T}.$$

Algorithm 1: LinBreg.

default: $\delta = 1$

$\theta \leftarrow$ sparse, $\quad v \leftarrow \partial J(\theta) + \frac{1}{\delta}\theta$ `// Initialize`

for epoch $e = 1$ **to** E **do**

  for minibatch $B \subset \mathcal{T}$ **do**

    $g \leftarrow \nabla L(\theta; B)$ `// Backpropagation`

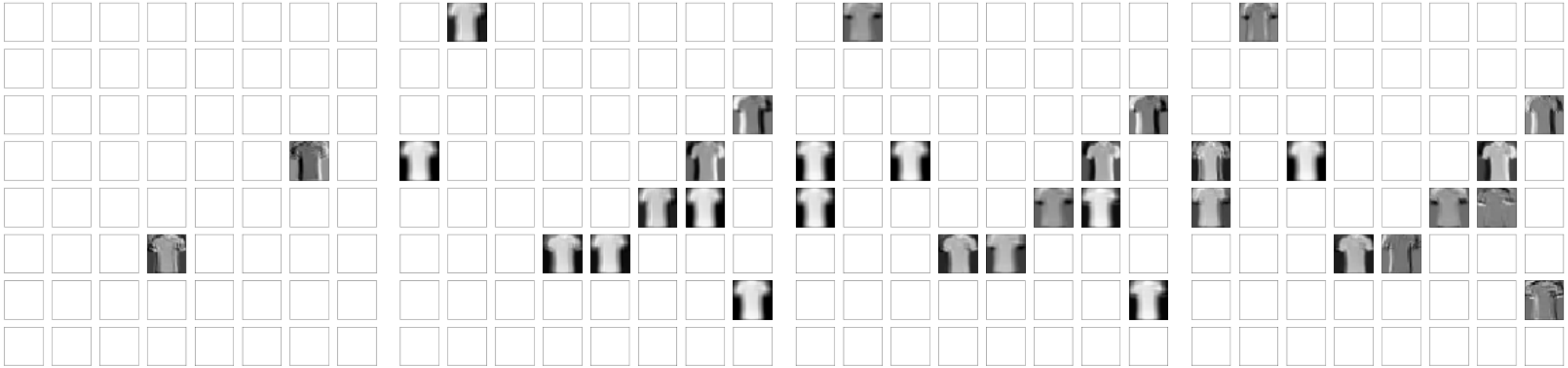
    $v \leftarrow v - \tau g$ `// Gradient step`

    $\theta \leftarrow \operatorname{prox}_{\delta J}(\delta v)$ `// Regularization`

Sparse Convnet

We can also incorporate convolutional kernels $K^l_{i,j} \in \mathbb{R}^{k,k}$, for which we employ a group sparsity regularizer, i.e.,

$$J(\theta) := \sum_{l,i,j} \left\| K^l_{i,j} \right\|_2 .$$



(a) Iteration 0 (b) Iteration 5 (c) Iteration 20 (d) Iteration 100

Neural Architecture search

Use group sparsity for different groups

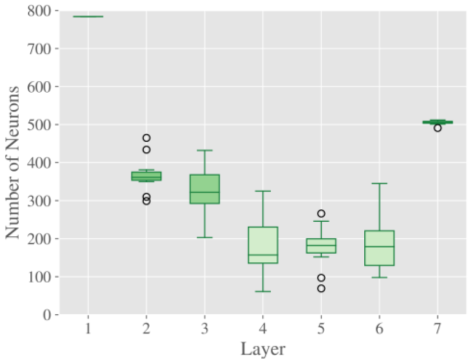
$$J(u) = \sum_i \|u_{G_i}\|_2 + \frac{\delta^2}{2}\|u\|_2^2$$

- few neurons: group weights corresponding to one neuron
- few skip connections
- few layers (only for residual network)

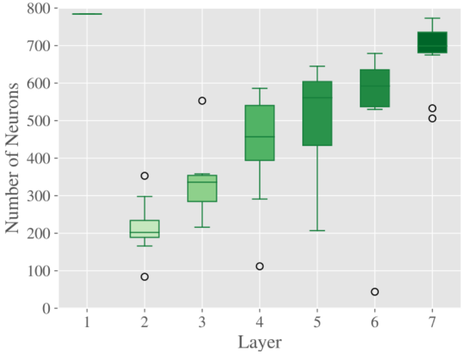
Sparsity for Different Tasks

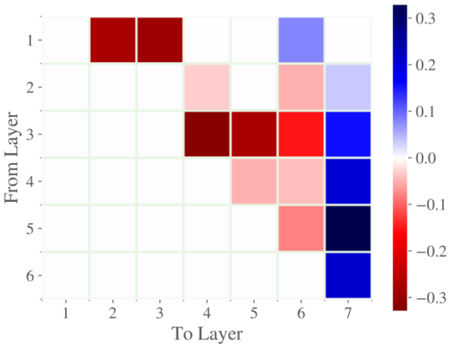
Denoising

Without skip



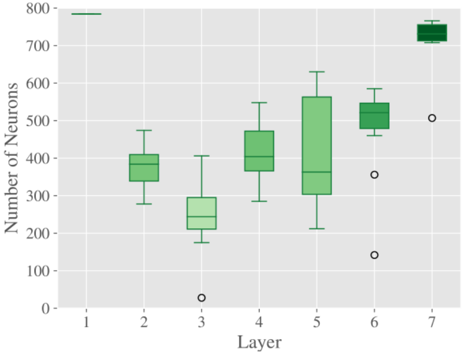
With skip connections

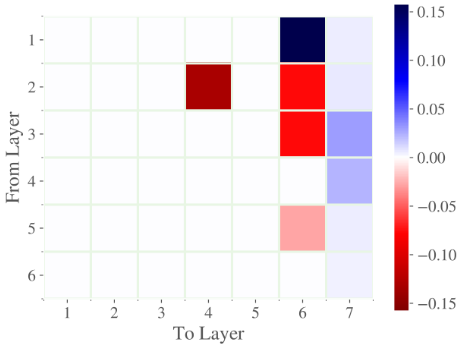




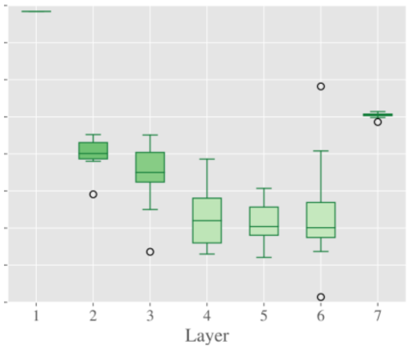
Deblurring

With skip connections





Without skip

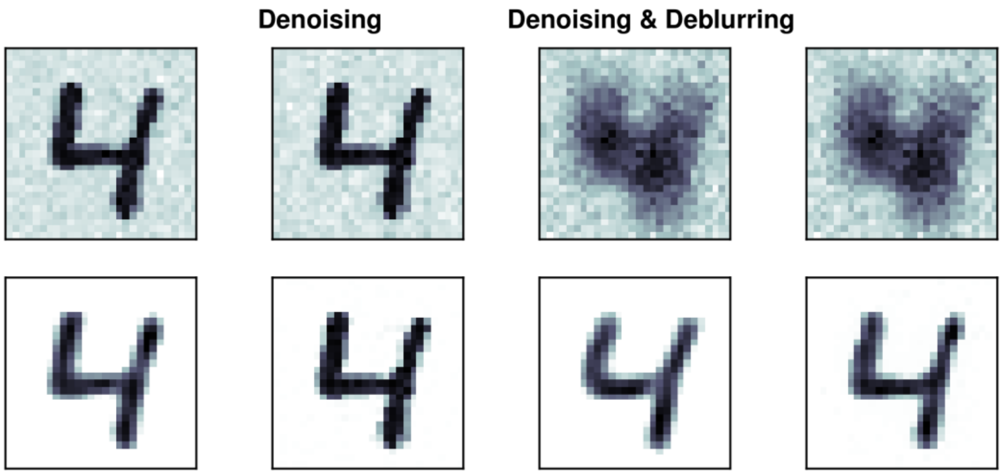


Outlook: Lossless Data Compression

Neural architecture search for image reconstruction

Train an autoencoder structure with low complexity in the middle layer (encoder)

Save encoded data and parameters of decoder network



Outlook: Robustness and Sparsity

Do we gain robustness for sparse networks ?

Do we loose robustness for sparse networks ? (larger Lipschitz constant ?)

Notkestraße 85
22607 Hamburg

+49 40 8998-4198
info@helmholtz-imaging.de

www.helmholtz-imaging.de

© 2021, Helmholtz Imaging. Unauthorized use and reproduction, as well as any transfer to third parties without consultation is not permitted.