Mathematisches Forschungsinstitut Oberwolfach

Report No. 11/2018

DOI: 10.4171/OWR/2018/11

Mini-Workshop: Deep Learning and Inverse Problems

Organised by
Simon Arridge, London
Maarten Valentijn de Hoop, Houston
Peter Maaß, Bremen
Carola-Bibiane Schönlieb, Cambridge UK

4 March – 10 March 2018

Abstract. Machine learning and in particular deep learning offer several data-driven methods to amend the typical shortcomings of purely analytical approaches. The mathematical research on these combined models is presently exploding on the experimental side but still lacking on the theoretical point of view. This workshop addresses the challenge of developing a solid mathematical theory for analyzing deep neural networks for inverse problems.

Mathematics Subject Classification (2010): 65J22, 65Y20, 94C99.

Introduction by the Organisers

The mini-workshop *Deep Learning and Inverse Problems*, organized by Simon Arridge (London), Maarten Valentijn de Hoop (Houston), Peter Maaß (Bremen) and Carola-Bibiane Schönlieb (Cambridge) was well attended with 15 participants and aimed at bringing together experts from different scientific directions to contribute in overall 11 talks mathematically proven results in the theory of deep neural networks for inverse problems.

The classical approach to inverse problems starts with an analytical description $F : X \to Y$ of the forward operator in some function spaces X and Y. The main target in inverse problems is to reconstruct an unknown x^* from given noisy data $y^\delta \sim F(x^*)$, where the generalized inverse F^{-1} is unbounded. However, these purely analytic models are typically just an approximation to the real application and their extension are often restricted due to the high degree of complexity or an only partial understanding of the underlying physical processes. Furthermore,

the inputspace of many applications will be just a subspace of the whole function space X and obey an unknown stochastic distribution.
The huge field of machine learning provides several data-driven approaches to tackle these problems by using training datasets to either construct a problem adapted forward operator and use an established inversion method or to solve the inverse problem directly. In particular deep learning approaches using neural networks with multiple internal layers have become popular over the last decade. However, no consistent mathematical theory on deep neural networks for inverse problems has been developed yet besides the stunning experimental results, which have been published so far for many different types of applications to inverse problems.
One theme which was addressed by several talks during the workshop was the interpretation of the different layers of a neural network as discretization of continuous systems like ODEs, PDEs and integro-differential equations (see the abstracts of Eldad Haber, Lars Ruthotto, Carola Schönlieb and Thomas Pock). These approaches allow to address in particular stability of neural networks and it allows to develop novel network designs based on classical discretization schemes for inverse problems.
Optimization of neural networks by using a functional analytical network for e.g. optimizing activation functions (Michael Unser) or regularization schemes (Martin Benning) was a second common point of discussion. The classical interpretation of Tikhonov regularization for inverse problems can thus be mirrored by the design of the neural network.
Another source of inspiration for tackling deep neural networks is harmonic analysis as a concept for detecting invariants in large data sets and for interpretation on neural network behaviour in general (see the abstracts of Maarten de Hoop and Gitta Kutyniok).
On the application side, the focus was on medical imaging, where remarkable results were presented for CT-reconstructions with few measurements (Ozan Öktem) and for the linear problem of photoacoustic tomography (Andreas Hauptmann and Simon Arridge).

Acknowledgement: The MFO and the workshop organizers would like to thank the National Science Foundation for supporting the participation of junior researchers in the workshop by the grant DMS-1641185, “US Junior Oberwolfach Fellows”.

Mini-Workshop: Deep Learning and Inverse Problems

Table of Contents

Abstracts

Deep Networks meet ODEs

Eldad Haber
(joint work with L. Ruthotto, E. Holtham, L. Meng, B. Chang)

In this work, we explore deep learning (see e.g. [2, 1]) from the point of view of dynamical systems. A main goal is to use the interpretation of deep neural networks as a parameter estimation problem of a nonlinear dynamical system to analyze the stability of the forward propagation for simplified Residual Network (ResNet) architectures [5].
For given training data $\mathbf{Y}_0 = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_s]^\intercal \in \mathbb{R}^{s\times n}$ and $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_s]^\intercal \in \mathbb{R}^{s\times m}$, we consider the forward propagation of the ResNet, whose N layers are given by

$$\mathbf{Y}_{j+1} = \mathbf{Y}_j + h\sigma(\mathbf{Y}_j K_j + b_j) \qquad \text{for} \qquad j = 0, \ldots, N-1, \tag{1}$$

with a nonlinear activation function $\sigma : \mathbb{R}^{s\times n} \to \mathbb{R}^{s\times n}$, weights $\mathbf{K}_0, \ldots, \mathbf{K}_{N-1} \in \mathbb{R}^{n\times n}$, biases $b_0, b_1, \ldots, b_{N-1} \in \mathbb{R}$ and a parameter $h > 0$ to allow a continuous interpretation of the model. The description of the ResNet in (1) can be regarded as an explicit Euler discretization of the nonlinear Ordinary Differential Equation (ODE)

$$\dot{\mathbf{y}}(t) = \sigma(\mathbf{K}^\intercal(t)\mathbf{y}(t) + b(t)), \quad \text{with} \quad \mathbf{y}(0) = \mathbf{y}_0, \tag{2}$$

for a time interval $t \in [0, T]$ with final time $T > 0$ (see also [4, 3]). Therefore, the stability of the forward propagation can be analyzed via the classical stability theory of ODEs. Accordingly, the ODE is stable if

$$\max_{i=1,2,\ldots,n} \Re(\lambda_i(\mathbf{J}(t))) \leq 0$$

for all $t \in [0, T]$, where $\Re$ denotes the real part and $\lambda_i(\mathbf{J}(t))$ is the i-th eigenvalue of the Jacobian $\mathbf{J}(t) \in \mathbb{R}^{n\times n}$ of the right-hand side in Equation (2). Based on these observations, we derive stability criteria for such ResNet models and develop new network architectures, which ensure stability as well as a well-posed learning problem.

References

[1] J. Friedmann, T. Hastie and R. Tibshirani. *The elements of statistical learning*, Springer series in statistics Springer, (2001).
[2] I. Goodfellow, Y. Bengio and A. Courville. *Deep Learning*, MIT Press, (2016).
[3] E. Haber and L. Ruthotto. *Stable architectures for deep neural networks*, Inverse Problems, **34**(1), (2018).
[4] E. Haber, L. Ruthotto and E. Holtham. *Learning across scales - a multiscale method for convolution neural networks*, *arXiv preprint*, arXiv:1703.02009, (2017).
[5] K. He, X. Zhang, S. Ren and J. Sun. *Deep residual learning for image recognition*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 259–268 (2016).

Expressibility of Sparsely Connected Deep Neural Networks

Gitta Kutyniok
(joint work with Helmut Bölcskei, Philipp Grohs, Philipp Petersen)

The last years have seen a renaissance of neural networks, which can mainly be attributed to the massive amounts of available training data and the significantly increased computational power allowing the training of *deep* neural networks. The outstanding success of deep neural networks in real-world applications can be witnessed in applications such as game intelligence, image analysis, and speech recognition. Despite this success story, most of the related research is empirically driven and a mathematical foundation is almost completely missing.
One central task of a neural network is to approximate a function, which for instance encodes a classification task. Given a set of sample values, the network is then trained by, for instance, stochastic gradient descent. Thus, one key question concerns the general ability of deep neural networks to approximate functions. Certainly, the more neurons, edges, and layers are available, the more the approximation power – also coined *expressiblity* – of the network should grow. Hence it seems conceivable to consider this question for a restricted number of edges or layers, say. Such *sparse* deep neural networks – often referred to as networks with *sparse connectivity* – are also interesting from the point of view of computational efficiency and memory requirements.
To analyze the question of the expressive power of sparsely connected deep neural networks, we start with a mathematically precise definition of a neural network.

Definition 1. Let d be the dimension of the input layer, L the total number of layers, $N_1, \dots, N_L$ the dimensions of the $L-1$ hidden layers, and hence $N := \sum_{j=1}^{L} N_j$ the total number of neurons. Then $\Phi : \mathbb{R}^d \to \mathbb{R}^{N_L}$ given by

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

is called a *neural network*, which is composed of non-linear functions $\rho : \mathbb{R} \to \mathbb{R}$ – often referred to as *activation functions* or *rectifiers* – acting component-wise and affine linear maps $W_\ell : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell}$, $1 \leq \ell \leq L$. We denote the class of such neural networks by $\mathcal{NN}_{L,M,d,\rho}$

Let us next review how approximation theory measures the approximation rate depending on certain restricting conditions on the approximation. One common scenario is to consider a class of functions $\mathcal{C}$ within $L^2(\mathbb{R}^d)$ and a complete system $(\varphi_i)_{i \in I} \subseteq L^2(\mathbb{R}^d)$, with the restricting condition being the number of elements of the system allowed to compose an approximation. Taking this viewpoint, one then studies the *error of best M-term approximation* of some $f \in \mathcal{C}$ given by

$$\|f - f_M\|_{L^2(\mathbb{R}^d)} := \inf_{I_M \subset I, \# I_M = M, (c_i)_{i \in I_M}} \|f - \sum_{i \in I_M} c_i \varphi_i\|_{L^2(\mathbb{R}^d)},$$

where f_M is referred to as *best M-term approximation.* The largest $\gamma > 0$ such that

$$M^\gamma \sup_{f\in\mathcal{C}} \|f - f_M\|_{L^2(\mathbb{R}^d)} \to 0 \qquad \text{as } M \to \infty$$

determines the *optimal (sparse) approximation rate* of $\mathcal{C}$ by the system $(\varphi_i)_{i\in I}$. From a conceptual viewpoint, we derive a relation between the approximation accuracy and the complexity of the approximating system in terms of sparsity.
We now transfer this concept to the approximation with neural networks, replacing the complexity of the approximating system in terms of sparsity by the complexity of the approximating neural network in terms of sparse connectivity. For this, we first aim to derive a lower bound on the sparsity of the connectivity. This requires the introduction of a complexity measure for a class of functions $\mathcal{C}$ within $L^2(\mathbb{R}^d)$. We choose the following notion from rate-distortion theory, which describes the dependence of the minimally possible code length of $\mathcal{C}$ on the required approximation quality.

Definition 2. Let $d \in \mathbb{N}$ and, for each $\ell \in \mathbb{N}$, let $\mathfrak{E}^\ell := \{E : L^2(\mathbb{R}^d) \to \{0,1\}^\ell\}$ denote the set of *binary encoders with length* ℓ and $\mathfrak{D}^\ell := \{D : \{0,1\}^\ell \to L^2(\mathbb{R}^d)\}$ the set of *binary decoders of length* ℓ. For arbitrary $\epsilon > 0$ and $\mathcal{C} \subset L^2(\mathbb{R}^d)$, the *minimax code length* $L(\epsilon, \mathcal{C})$ is then given by

$$L(\epsilon,\mathcal{C}) := \min\{\ell \in \mathbb{N} : \exists (E,D) \in \mathfrak{E}^\ell \times \mathfrak{D}^\ell : \sup_{f\in\mathcal{C}} \|D(E(f)) - f\|_{L^2(\mathbb{R}^d)} \le \epsilon\},$$

and the *optimal exponent* $\gamma^*(\mathcal{C})$ is defined by

$$\gamma^*(\mathcal{C}) := \inf\{\gamma \in \mathbb{R} : L(\epsilon,\mathcal{C}) = O(\epsilon^{-\gamma})\}.$$

This now allows us to state our first main result from [1]. Notice how the lower bound on the number of edges for a given approximation accuracy is dependent on the optimal exponent.

Theorem 1. *Let $d \in \mathbb{N}$, $\rho : \mathbb{R} \to \mathbb{R}$, $c > 0$, and let $\mathcal{C} \subset L^2(\mathbb{R}^d)$. Further, let*

$$\mathbf{Learn} : (0,1) \times \mathcal{C} \to \mathcal{NN}_{\infty,\infty,d,\rho}$$

satisfy that, for each $f \in \mathcal{C}$ and $0 < \epsilon < 1$, all weights of $\mathbf{Learn}(\epsilon, f)$ can be encoded with $< -c\log_2(\epsilon)$ bits and $\sup_{f\in\mathcal{C}} \|f - \mathbf{Learn}(\epsilon, f)\|_{L^2(\mathbb{R}^d)} \le \epsilon$. Then, for all $\gamma < \gamma^*(\mathcal{C})$,

$$\epsilon^\gamma \sup_{f\in\mathcal{C}} \mathcal{M}(\mathbf{Learn}(\epsilon, f)) \to \infty \qquad \text{as } \epsilon \to 0,$$

where $\mathcal{M}(\mathbf{Learn}(\epsilon, f))$ denotes the number of non-zero weights in $\mathbf{Learn}(\epsilon, f)$.

This now raises the question, whether this is indeed a sharp result in the sense of whether – for a given class of functions $\mathcal{C}$ – there exists a neural network such that

$$\sup_{f\in\mathcal{C}} \mathcal{M}(\mathbf{Learn}(\epsilon, f)) = O(\epsilon^{-\gamma^*(\mathcal{C})}) \qquad \text{as } \epsilon \to 0.$$

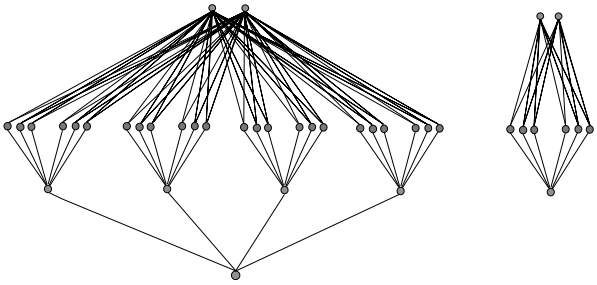


FIGURE 1. Left: A neural network mimicking an M-term approximation composed of elements of $(\varphi_i)_{i\in I}$. Right: A neural network mimicking some φ_i.

This would lead to *optimally memory efficient* deep neural networks.
Let now $\mathcal{C}$ be a class of functions in $L^2(\mathbb{R}^d)$ and $(\varphi_i)_{i\in I} \subset L^2(\mathbb{R}^d)$. We first assume that $(\varphi_i)_{i\in I}$ satisfies that, for each $i \in I$, there exists

(1) a neural network Φ_i with at most $C > 0$ edges such that $\varphi_i = \Phi_i$.

By the construction illustrated in Figure 1, we can in fact construct a network Φ with $O(M)$ edges, which mimics an M-term approximation as

$$\Phi = \sum_{i\in I_M} c_i\varphi_i, \quad \text{if } |I_M| = M.$$

Next, assume that, in addition, we find some $\tilde{C} > 0$ such that, for all $f \in \mathcal{C} \subset L^2(\mathbb{R}^d)$, there exists $I_M \subset I$ with

$$\|f - \sum_{i\in I_M} c_i\varphi_i\| \leq \tilde{C}M^{-1/\gamma^*(\mathcal{C})}. \tag{2}$$

This leads to the following result from [1], stated here in a significantly simplified form. Notice though that already this result indicates how to derive neural networks which are memory optimal.

Theorem 2. *Let $\mathcal{C} \subset L^2(\mathbb{R}^d)$ and $(\varphi_i)_{i\in I} \subset L^2(\mathbb{R}^d)$ satisfying* (1) *and* (2)*. Then every $f \in \mathcal{C}$ can be approximated up to an error of ϵ by a neural network with only $O(\epsilon^{-\gamma^*(\mathcal{C})})$ edges.*

Proof. By (1), there exists a network Φ with $O(M)$ edges with $\Phi = \sum_{i\in I_M} c_i\varphi_i$. Now set $\epsilon = \tilde{C}M^{-1/\gamma^*(\mathcal{C})}$ and solve (2) for the number of edges M. This yields $M = O(\epsilon^{-\gamma^*(\mathcal{C})})$. □

Thus, for those networks there exists some constant $C > 0$ satisfying

$$\sup_{f\in\mathcal{C}} \mathcal{M}(\mathbf{Learn}(\epsilon, f)) \leq C\epsilon^{-\gamma^*(\mathcal{C})} \qquad \text{for all } \epsilon > 0,$$

thereby showing that the bound in Theorem 1 is sharp.

Since in applied harmonic analysis, representation systems such as wavelets, ridgelets, or shearlets – more generally α-shearlets [2], which includes those as

special cases – are build as affine systems, they are particularly amenable to the requirement (1) and in fact do satisfy it. In addition, it is known that α-shearlets provide (almost) optimally sparse approximation properties for α-cartoon-like functions [2, 3] – which could even be regarded as classification functions –, thereby fulfilling (2).
For more details we refer to [1]. Thus, roughly speaking, it is fair to say that deep neural networks have as much approximation power as classical systems from the area of applied harmonic analysis.
Numerical experiments using samples of simple classification functions as input to a network of a topology as given in Figure 1 show that already the standard backpropagation algorithm generates deep neural networks (almost) obeying those optimal approximation rates. Most intriguingly, one can even witness that this network then automatically learns those representation systems from applied harmonic analysis which are provably known to provide optimally sparse approximations. For the numerical experiments, please see [1].

References

[1] H. Bölcskei, P. Grohs, G. Kutyniok and P. Petersen. *Optimal Approximation with Sparsely Connected Deep Neural Networks*, *arXiv preprint*, arXiv:1705.01714.
[2] P. Grohs, S. Keiper, G. Kutyniok and M. Scäfer. α-*Molecules*, Appl. Comput. Harmon. Anal. **42**, 297–336 (2016).
[3] F. Voigtlaender and A. Pein. *Analysis vs. synthesis sparsity for α-shearlets*, *arXiv preprint*, arXiv:1702.03559.

PDE-Based Image Classification using Deep Neural Networks

Lars Ruthotto
(joint work with Eldad Haber)

In this talk, we presented the connection between Convolutional Neural Networks (CNNs), which are a common machine learning method applied widely to speech, image, and video data, and Partial Differential Equations (PDEs), which are invaluable in modeling physical phenomena but are also used broadly in many branches of applied mathematics. Our interpretation shows that training neural networks is similar to the inverse problem of estimating parameters of a nonlinear, time-dependent system of PDEs.

The abstract goal of machine learning is to find a function $f : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^m$ that can be described by its parameter $\theta \in \mathbb{R}^p$ such that $f(\cdot, \theta)$ accurately predicts the result of an observed phenomenon (e.g., the class of an image, a spoken word etc.). In supervised learning, a set of input features $\mathbf{y}_1, \ldots, \mathbf{y}_s \in \mathbb{R}^n$ and output labels $\mathbf{c}_1, \ldots, \mathbf{c}_s \in \mathbb{R}^m$ is available and used to build the model $f(\cdot, \theta)$. The output labels are vectors that represent the probability of a particular example, to belong to each class. For brevity, we will denote the training data by $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_s] \in \mathbb{R}^{n \times s}$ and $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_s] \in \mathbb{R}^{m \times s}$.

Motivated by supervised image classification, we consider deep residual networks (ResNets), introduced in [4]. We derived a continuous interpretation of the

filtering provided by ResNets in [1], and similar observations were made in [5]. ResNets can be seen as a forward Euler discretization (with fixed step size of $\delta_t = 1$) of the initial value problem on $t \in [O, T]$

$$\partial_t \mathbf{Y}(\theta, t) = \mathbf{F}(\theta(t), \mathbf{Y}(t)), \text{ for } t \in (0, T] \tag{1}$$

$$\mathbf{Y}(\theta, 0) = \mathbf{Y}_0, \tag{2}$$

where $\mathbf{Y}_0$ are the input features and $\mathbf{F}$ is an operator comprising of affine linear transformations and pointwise nonlinearities. The *learning problem* then consists of finding θ and weights of a linear classifier by solving

$$\min_{\theta, \mathbf{W}, \mu} \frac{1}{2} S(\mathbf{W}\mathbf{Y}(\theta, T) + \mathbf{B}\mu, \mathbf{C}) + \alpha R(\theta, \mathbf{W}, \mu), \tag{3}$$

where S is a loss function (e.g., least-squares misfit or a cross entropy), $\mathbf{B}\mu$ is a bias vector, and R is a regularizer (e.g., a Tikhonov regularizer also referred to as weight decay).

When dealing with image data, the features in $\mathbf{Y}$ have an exploitable structure. We highlighted the relation between deep residual CNNs and nonlinear systems of PDEs. In convolutional neural networks, the weights θ parameterize a typically band-limited convolution operator that is used to filter the data. Due to their compact support, the convolution operator can be written as a linear combination of partial differential operators whose contributions are controlled by θ. In our case of image classification, the input data, $\mathbf{Y}$ can be seen as a discretization of a continuous process, $Y(x)$ and hence we obtain a fully continuous setting for training deep residual CNNs.

The key idea presented in this talk is to design the function $\mathbf{F}$ in (1) so that the resulting CNN can be related to, e.g., parabolic or hyperbolic PDEs. This structure can then be used computationally. Note that for classical residual neural networks the type of the underlying PDE depends on the choice of θ and can even vary in time. We present three different dynamics that are inspired by parabolic, hyperbolic and Hamiltonian systems. The parabolic networks allow one to devise multiscale training algorithms, whereas the latter two types allow for memory-free implementation. Despite their very different characteristics, we found in extensive numerical examples that all CNNs can be trained and that in many cases committing to a dynamic helps the CNN generalize better.

The talk led to many fruitful discussions that identified a few future research questions, including:

- The continuous framework results in a new understanding of the concepts of layers as time discretization points. This motivates adaptive schemes for choosing layers, e.g., shallow-to-deep training. While we have good results for small-scale test problems with deterministic optimization methods [1] an interesting and open question is how to incorporate such a scheme in stochastic optimization methods.
- Another question concerns the design of regularization functionals for the learning problem. Currently, smoothness in time is enforced and found

to help both in ensuring stability and in generalization. However, more explicit prior knowledge and constraints will be considered in the future.
- In the context of inverse problems an important question is the stability of the resulting CNN. The conjecture is that the PDE-motivated networks - after suitable discretization - help enforce stability. However, more thorough analysis and detailed numerical experiment need to be performed.

References

[1] E. Haber and L. Ruthotto. *Stable architectures for deep neural networks*, Inverse Problems abs/1705.03341, 1–21 (2017).
[2] E. Haber, L. Ruthotto, E. Holtham and S. H. Jun. *Learning across scales - Multiscale methods for convolution neural networks*, AAAI Conference, 1–8 (2018).
[3] E. Haber, L. Ruthotto and E. Holtham. *Reversible architectures for arbitrarily deep residual neural networks*, AAAI Conference, 1–8 (2018).
[4] K. He, X. Zhang, S. Ren and J. Sun. *Deep residual learning for image recognition*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778 (2016).
[5] E. Weinan. *A Proposal on Machine Learning via Dynamical Systems*, Communications in Mathematics and Statistics, **5**(1), 1–11 (2017).

Recent Approaches for Using Machine Learning in Image Reconstruction

Ozan Öktem

(joint work with Jonas Adler, Carola-Bibiane Schönlieb, Sebastian Lunz)

The starting point in using techniques from machine learning to solve an inverse problem is to rephrase the latter as a statistical estimation problem, so a reconstruction method becomes a (non-randomised) decision rule. Statistical decision theory can then be used for selecting an appropriate method.

Statistical Inverse Problems

Let X and Y denote separable Banach spaces where $(Y, \mathfrak{S}_Y)$ and $(X, \mathfrak{S}_X)$ are measurable spaces. Next, define the data model as $\mathcal{M}\colon X \to \mathscr{P}_Y$ where $\mathscr{P}_Y$ is the set of probability measures on Y. Following [4], a (statistical) inverse problem is the task of estimating $x^* \in X$ from data $y \in Y$, which is a single observation generated by Y-valued random variable $\mathsf{y} \sim \mathcal{M}(x^*)$ with a known data model $\mathcal{M}\colon X \to \mathscr{P}_Y$.

It is often common to introduce a X-valued random variable x that generates x^*. In such case, the data model can be thought of as the data likelihood, i.e., $\mathcal{M}(x) = \mathcal{L}(\mathsf{y} \mid \mathsf{x} = x)\,\mathrm{d}y$. Furthermore, if $\mathsf{x} \sim \mathbb{P}_{x^*}$ with a probability distribution fully specified by $x^* \in X$, then $(\mathsf{x}, \mathsf{y}) \sim \mathbb{P}_{x^*} \otimes \mathcal{M}(x^*)$. A special case is when the Y-valued random variable y is given by a known forward operator $\mathcal{A}\colon X \to Y$ as

$$\mathsf{y} = \mathcal{A}(x^*) + \mathsf{e} \quad \text{with } \mathsf{e} \sim \mathbb{P}_{\text{noise}}. \tag{1}$$

This corresponds to having a data model $\mathcal{M}(x) := \delta_{\mathcal{A}(x)} * \mathbb{P}_{\text{noise}} = \mathbb{P}_{\text{noise}}\big(\,\cdot - \mathcal{A}(x)\big)$ whenever e does not depend on $\mathcal{A}(x^*)$.

Reconstruction

A reconstruction method is formally a measurable X-valued mapping on Y, i.e., it can be viewed as non-randomised decision rule. More precisely, the tuple $\big((Y, \mathfrak{S}_Y), \{\mathcal{M}(x)\}_{x\in X}\big)$ defines a statistical model, which in turn defines a statistical decision problem by selecting $(X, \mathfrak{S}_X)$ as decision space and considering a loss function given by a functional $\ell_X \colon X \times X \to \mathbb{R}$.
A common framework is Bayesian regularisation, which starts out by assuming that $(\mathsf{x}, \mathsf{y}) \sim \pi_0 \otimes \mathcal{M}(x^*)$ where both π_0 and $x \mapsto \mathcal{M}(x)$ are known and $x^* \in X$ is unknown. It therefore becomes natural to estimate x^* by exploring the posterior distribution, e.g., to maximise it (maximum a posteriori estimator). Another is to minimise Bayes risk, i.e., to consider the reconstruction method $\mathcal{A}^\dagger \colon Y \to X$ that minimises the π_0-averaged expected loss:

$$\mathcal{R}_{\pi_0}(\mathcal{A}^\dagger) := \mathbb{E}_\mu\Big[\ell_X\big(\mathsf{x}, \mathcal{A}^\dagger(\mathsf{y})\big)\Big] \quad \text{where } \mu = \pi_0 \otimes \mathcal{M}(x^*).$$

In the Gaussian setting, there is a well developed theory, e.g., one can describe how the posterior distribution concentrates its mass near $x^* \in X$ as $\mathsf{e} \to 0$. Furthermore, for certain class of forward operators it is also possible to characterise the microscopic fluctuations of the posterior around x^*. The latter involves considering the inverse of the associated normal operator (Fischer information operator), which describes how the ill-posedness of the inverse problem influences the uncertainty, see [8] for more about this line of research.

Challenges

Bayesian non-parametric theory [5] provides a large class of priors, and variants used in imaging primarily encode regularity properties [6, 3]. Despite this, available handcrafted priors π_0 are incomplete in the sense that they only captures a fraction of the a priori information that is available about x^*. As an example, a natural a priori information in medical imaging is that the object being imaged is a human being. It is very difficult, if not impossible, to explicitly construct a prior that captures this information.
Furthermore, exploring the posterior is highly non-trivial due to computational issues and MCMC techniques, albeit efficient, are insufficient for imaging applications. One can perhaps compute the maximum a posteriori estimator (MAP) estimator, but any estimator requiring integration over X, such as the posterior mean (conditional mean) or minimising Bayes risk, is computationally unfeasible. The same also holds for estimators relevant for uncertainty quantification.
As we shall see next, *both* these challenges are addressed by *learned iterative methods* that use a deep neural network to define an optimal reconstruction method, i.e., one that minimises Bayes risk $\mathcal{A}^\dagger \mapsto \mathcal{R}_{\pi_0}(\mathcal{A}^\dagger)$.

Learned Iterative Methods

Machine learning, and deep neural networks in particular, have demonstrated a remarkable capacity in capturing intricate relations from example data [7]. Instead of using handcrafted problem specific models, one uses generic models that are adapted through learning against example data. It is therefore temping to investigate whether one can learn a prior by these techniques.

First, instead of providing a handcrafted π_0, we have (supervised) training data $(x_i, y_i) \in X \times Y$ generated by $(\mathsf{x}, \mathsf{y}) \sim \mu = \pi_0 \otimes \mathcal{M}(x^*)$. Next, finding an optimal reconstruction method requires searching over all non-randomised decision rules, which is computationally unfeasible. Instead, we restrict our attention to those given by a (deep) neural network architecture since these have large capacity (can approximate any Borel measurable mapping arbitrarily well [9] and there are computationally feasible implementations. To summarise, we have a parametrised family of reconstruction methods $\mathcal{A}_\theta^\dagger \colon Y \to X$ and the optimal one is given by

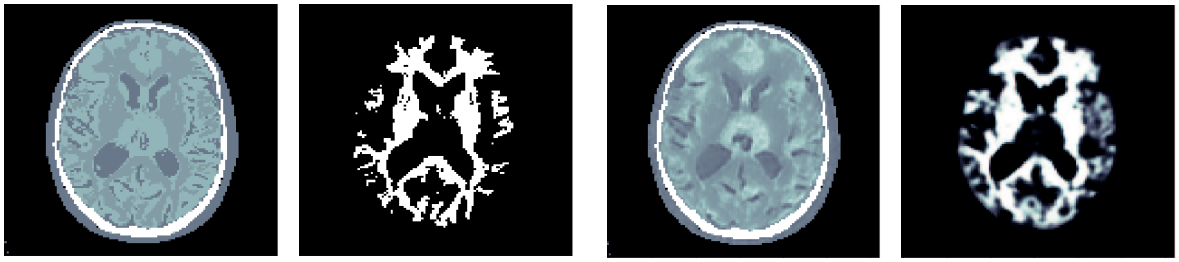
$$\theta^* \in \operatorname*{arg\,min}_{\theta \in \mathbb{R}^N} \mathbb{E}_\mu \Big[\ell_X \big(\mathsf{x}, \mathcal{A}^\dagger(\mathsf{y}) \big) \Big] \approx \operatorname*{arg\,min}_{\theta \in \mathbb{R}^N} \Big[\frac{1}{m} \sum_i \ell_X \big(x_i, \mathcal{A}_\theta^\dagger(y_i) \big) \Big]. \tag{2}$$

The above is a fully data driven approach for reconstruction, i.e., the (unknown) measure $\mu = \pi_0 \otimes \mathcal{M}(x^*)$ is replaced by its empirical counterpart given by the training data. Hence, the optimal reconstruction method $\mathcal{A}_{\theta^*}^\dagger$ is derived without utilising knowledge about how data is generated. This is an serious problem for imaging applications where the large number of unknowns require large training datasets that are not available. In many inverse problem, knowledge about how data is generated is contained in the data model $x \mapsto \mathcal{M}(x)$ that is known. The learned iterative schemes [1, 2] construct a deep convolutional neural network architecture that accounts for the data model, or more precisely the data likelihood. The idea is to unroll a fixed point iterative scheme relevant for solving the inverse problem and use a deep convolutional residual network to learn the iterative update. The resulting reconstruction method is computationally feasible and outperforms state-of-the-art by a significant margin as shown in [2].

Task Based Reconstruction

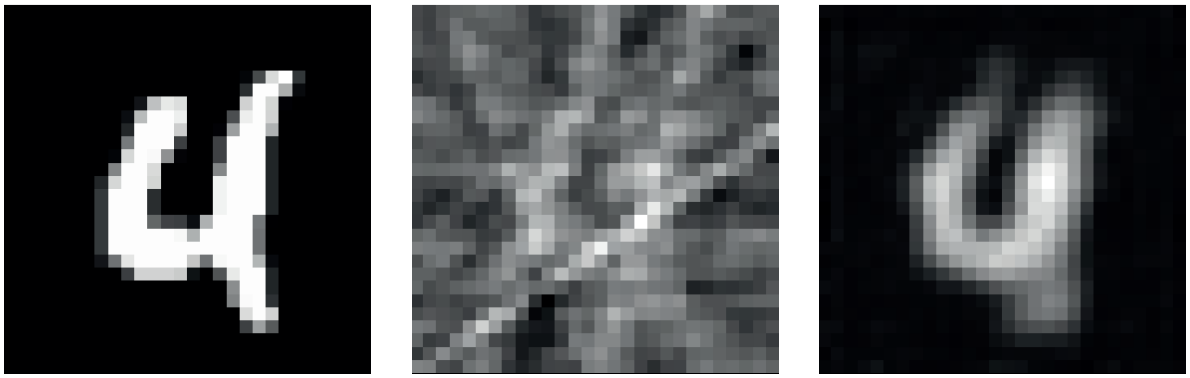
A learned iterative methods for reconstruction can be combined in a generic manner with any task that can be adequately adressed using a neural network, thereby resulting in an end-to-end reconstruction method that is adapted to the task.

More precisely, let $\big((X, \mathfrak{S}_X), \{\mathbb{P}_a\}_{a \in \triangle} \big)$ be a statistical model for the reconstruction space and consider any task that can be formalised as a non-randomised decision rule $\mathcal{T} \colon X \to D$, where $(D, \mathfrak{S}_D)$ is a task adapted decision space. We also introduce the task related loss function $L_D(\Theta, a) := \ell_D\big(\tau(\Theta), a\big)$ for given $\tau \colon \triangle \to D$ (feature extraction map) and $\ell_D \colon D \times D \to \mathbb{R}$. If $(\mathsf{x}, \mathsf{a}) \sim \eta$ denotes a $(X \times D)$-valued random variable, then an 'optimal' task minimises the expected loss $\mathcal{T} \mapsto \mathbb{E}_\eta \big[\ell_D \big(\mathcal{T}(\mathsf{x}), \mathsf{a} \big) \big]$.



True image & segmentation ($k = 2$). Joint reconstruction and segmentation.

FIGURE 1. Joint tomographic reconstruction and segmentation. The reconstructed segmentation is a grey-scale image with values in $[0, 1]$ that gives probability that point is part of the segmented structure.



True image of a '4'. Sequential scheme: Image classified as '8' with 99.99% confidence. Joint scheme: Image classified as '4' with 99.70% confidence.

FIGURE 2. Joint tomographic reconstruction and classification of MNIST images. Tomographic data is from 5 directions and highly noisy (Poisson noise). Classification from noisy tomographic data using the sequential scheme yields an overall accuracy of 93.35% whereas the joint scheme has an accuracy of 96.60%. Classifying against clean images yields 97.5% accuracy. Clearly, a joint scheme outperforms a sequential one.

Just as with reconstruction, it is unfeasible to search over all decision rules so we restrict our attention to those given by a (deep) neural network architecture. Moreover, the joint law η is approximated by its empirical counterpart derived from (supervised) training data $(x_i, a_i) \in X \times D$ generated by $(\mathsf{x}, \mathsf{a}) \sim \eta$. To summarise, we have a parametrised family of mappings $\mathcal{T}_\Theta \colon X \to D$, each a candidate for modelling the task, and the optimal one is given by

$$\Theta^* \in \operatorname*{arg\,min}_{\Theta \in \mathbb{R}^M} \mathbb{E}_\eta \big[\ell_D\big(\mathcal{T}_\Theta(\mathsf{x}), \mathsf{a}\big) \big] \approx \operatorname*{arg\,min}_{\Theta \in \mathbb{R}^M} \Big[\frac{1}{m} \sum_i \ell_D\big(\mathcal{T}_\Theta(x_i), a_i\big) \Big]. \tag{3}$$

A joint task based reconstruction operator is now defined as $\mathcal{T}_\Theta \circ \mathcal{A}_\theta^\dagger \colon Y \to D$ and the issue at hand is how to select an 'optimal' parameter $(\theta, \Theta) \in \mathbb{R}^N \times \mathbb{R}^M$. We consider a choice that minimises the following *joint expected loss (risk)* against triplets of training data $(x_i, y_i, a_i) \in X \times Y \times D$ generated by $(\mathsf{x}, \mathsf{y}, \mathsf{a}) \sim \sigma$:

$$(\theta^*, \Theta^*) \in \operatorname*{arg\,min}_{(\theta,\Theta)\in\mathbb{R}^N\times\mathbb{R}^M} \mathbb{E}_{\widehat{\sigma}}\Big[C_1 \ell_X\big(\mathcal{A}_\theta^\dagger(\mathsf{y}), \mathsf{x}\big) + C_2 \ell_D\big(\mathcal{T}_\Theta \circ \mathcal{A}_\theta^\dagger(\mathsf{y}), \mathsf{a}\big)\Big] \tag{4}$$

The joint task based reconstruction operator is given by $\mathcal{T}_{\Theta^*} \circ \mathcal{A}_{\theta^*}^\dagger \colon Y \to D$ where (θ^*, Θ^*) solves (4). Note that $\widehat{\sigma}$ above is the empirical measure given by the training data (x_i, y_i, a_i) and it replaces the unknown measure σ. Furthermore, the parametrisation (neural network architecture) of $\mathcal{A}_\theta^\dagger$ incporsporates the knowledge of how data in Y is generated (data likelihood). We also note that it is possible to pre-train by solving (2) and (3) using separate training data sets.

We conclude with showing two examples involving grey-scale images on a domain $\Omega \subset \mathbb{R}^d$, so $X = L^2(\Omega, \mathbb{R})$. The first is joint reconstruction and segmentation (figure 1). The task of segmenting an image into k components can be represented by a mapping defined on X that associates a point in the image to a probability distribution over $\mathbb{Z}_k$ (k labels). This can be formulated as a statistical estimation problem with non-randomised decision rules where $\triangle$ is $\mathbb{Z}_k$-valued measurable maps on Ω, and $\mathbb{P}_\theta$ is some given set of probability measures on X. The decision space D is the set of measurable mappings from Ω to the class of probability measures on $\mathbb{Z}_k$ and the loss function is $L_D(\theta, a) := \ell_D\big(\tau(\theta), a\big)$ where

$$\ell_D(a, z) := \int_\Omega \Big[-\sum_{i\in\mathbb{Z}_k} a(t)(i) \log\big[z(t)(i)\big]\Big] \,\mathrm{d}t \quad \text{for } a, z \colon \Omega \to \mathscr{P}_{\mathbb{Z}_k},$$
$$\tau(\theta)(t) := \delta_{\theta(t)} \quad \text{for } \theta \colon \Omega \to \mathbb{Z}_k \text{ and } t \in \Omega.$$

Note that ℓ_D simply integrates the point-wise Shannon entropy of the (spatially independent) probability measures $a(t)$ and $z(t)$ for $t \in \Omega$. Such a task is e.g. well modelled by an 'off the shelf' U-net convolutional neural net [10].

The second is joint reconstruction and classification (figure 2) where an image is classified into one of k labels. Here D is the space of probability distributions over $\mathbb{Z}_k$ and ℓ_D is the cross entropy. Such a task is e.g. well modelled by an 'off the shelf' convolutional neural net classifier using ReLU activation with 3 convolutional layers, each followed by 2×2 max pooling for segmentation.

References

[1] J. Adler and O. Öktem. *Solving ill-posed inverse problems using iterative deep neural networks*, Inverse Problems, **33**(12), 124007, (2017).

[2] J. Adler and O. Öktem. *Learned primal-dual reconstruction*, IEEE Transactions on Medical Imaging, (2018).

[3] M. Benning and M. Burger. *Modern regularization methods for inverse problems*, Acta Numerica, **27**, 1–111, (2018).

[4] S. N. Evans and P. B. Stark. *Inverse problems as statistics*, Inverse Problems, **18**(4), R1–R55, (2002).

[5] S. Ghosal and A. W. van der Vaart. *Fundamentals of Nonparametric Bayesian Inference*, Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, (2017).
[6] J. P. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*, Applied Mathematical Sciences, **160**, (2005).
[7] Y. LeCun, Y. Bengio and G. Hinton. *Deep learning*, Nature, **521**(7553), 436–444, (2015).
[8] R. Nickl. *On Bayesian inference for some statistical inverse problems with partial differential equations*, Bernoulli News, **24**(2), 5–9, (2017).
[9] A. Pinkus. *Approximation theory of the MLP model in neural networks*, Acta Numerica, 143–195, (1999).
[10] O. Ronneberger, P. Fischer and T. Brox. *U-Net: Convolutional networks for biomedical image segmentation*, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015: 18th International Conference, Proceedings, Part III, Lecture Notes in Computer Science, **9351**, 234–241, (2015).

New Representer Theorems: From Compressed Sensing to Deep Learning

Michael Unser

Regularization is a classical technique for dealing with ill-posed inverse problems; it has been used successfully for biomedical image reconstruction and machine learning.

In this talk, we present a unifying continuous-domain formulation that addresses the problem of recovering a function f from a finite number of linear functionals corrupted by measurement noise. We show that depending on the type of regularization—Tikhonov vs. generalized total variation (gTV)—we obtain very different types of solutions/representer theorems.

While the solutions can be interpreted as splines in both cases, the main distinction is that the spline knots are fixed and as many as there are data points in the former setting (classical theory of RKHS) [5, 2], while they are adaptive and few in the case of gTV [4].

Finally, we consider the problem of the joint optimization of the weights and activation functions in a deep neural network subject to a second-order total variation penalty. The remarkable outcome is that the optimal configuration is achieved with a deep-spline network that can be realized using standard ReLU units [3]. The latter result is compatible with the state-of-the-art in deep learning, but it also suggests some new computational/ optimization challenges.

References

[1] Y. LeCun, Y. Bengio and G. Hinton. *Deep learning*, Nature, **521**, 436–444 (2015).
[2] B. Schölkopf and A.J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT press, (2002).
[3] M. Unser. *A representer theorem for deep networks*, *arXiv preprint*, arXiv:1802.09210, (2018).
[4] M. Unser, J. Fageot and J. P. Ward. *Splines are universal solutions of linear inverse problems with generalized-TV regularization*, SIAM Review, **59**(4), 769–793 (2017).
[5] G. Wahba. *Spline Models for Observational Data*, Society for Industrial and Applied Mathematics, Philadelphia, PA, (1990).

Deep Learning for some Tomographic Problems

Simon Arridge and Andreas Hauptmann

(joint work with Jonas Adler, Paul Beard, Marta Betcke, Ben Cox, Sarah Hamilton, Nam Huynh, Felix Lucka, Vivek Muthurangu, Sebastien Ourselin, and Jennifer Steeden)

Mathematically, the task of reconstructing a tomographic image from measurement data is formulated as an inverse problem: given the unknown (image) of interest $f_{\text{true}} \in X$, the measured data $g \in Y$, and a forward operator $\mathcal{A} : X \to Y$, then the forward problem is modelled by the simple equation

$$g = \mathcal{A}(f_{\text{true}}) + \delta g,$$

where $\delta g \in Y$ denotes some noise in the observation. The inverse problem aims to recover f_{true} from the measurement of g. This is typically an ill-posed task which is conventionally approached through the design of an operator $\mathcal{A}^\dagger_{\mathcal{R}}$ based on knowledge of the forward and adjoint mappings and an explicit regularisation operator $\mathcal{R}$. However, in a learning based approach, the idea is to find a mapping $\mathcal{F}^\dagger_\theta$ parametrized by θ that is simple to design and faster to apply.

In this work we combine the conventional and learning based frameworks, and differentiate between two fundamentally different approaches:

(1) Model enforced: Direct reconstruction followed by learning based post-processing. In this approach image reconstruction is carried out using a simple/fast inversion step, and post-processing is used to remove artefacts and noise. In this case we are given a reconstruction operator $\mathcal{A}^\dagger : Y \to X$, then our *inverse mapping* is given by $\mathcal{F}^\dagger_\theta = \mathcal{G}_\theta \circ \mathcal{A}^\dagger$ where $\mathcal{G}_\theta : X \to X$ is typically a sophisticated convolutional neural network (CNN).

(2) Model based learning and reconstruction: In this approach the forward and adjoint operators of the imaging problem are used directly in the inverse algorithm. Here we learn an iterative update $f_{k+1} = G_\theta(\nabla d(g, \mathcal{F}(f_k)), f_k)$, where $d(g, \mathcal{F}(f_k))$ denotes the data-fit and $G_\theta : X \times X \to X$ is typically a simple CNN.

In the following we present various results of ongoing work in our research group, with a specific focus on application to experimental and clinical data.

Spatio-temporal De-aliasing for Magnetic Resonance Imaging

In magnetic resonance imaging (MRI) one obtains the measurement g as the Fourier transform of f. Ideally a stable reconstruction can be obtained by inverse Fourier transformation of fully sampled k-space data, but in cardiac imaging a full k-space sampling can only be obtained during cardiac gated breath-hold and especially sick and very young patients find breath-holding difficult. Thus, real-time sequences with highly undersampled data are required to achieve sufficient acceleration factors (14× in our application).

As shown in [5] a convolutional neural network is especially suited for inverse problems, such as MRI, where the normal operator $\mathcal{A}^*\mathcal{A}$ is of convolutional type.

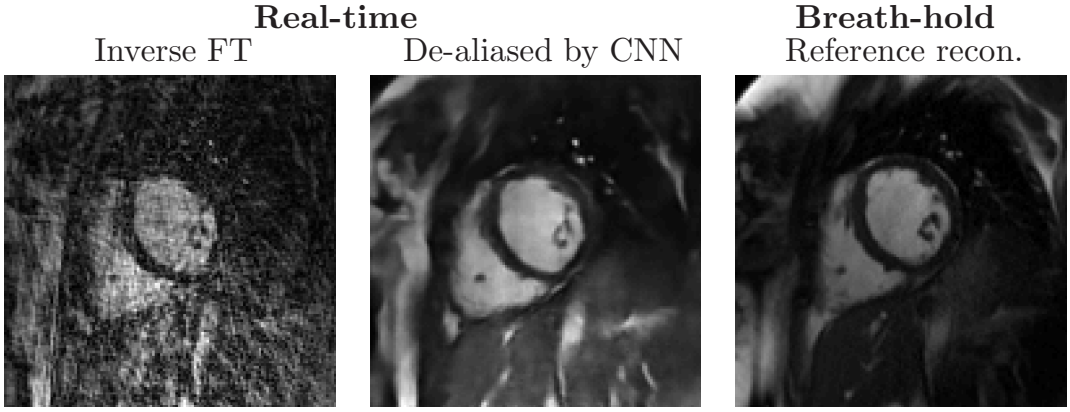


FIGURE 1. Reconstructions of actual real-time measurements in comparison to breath-hold reference of the same patient (right)

In our study [4], we extend this approach to a 3D (2D plus time) setting and investigate both reconstruction quality, and clinical relevance of the reconstruction.

The specific CNN was trained using synthetic training data created from previously acquired breath hold cine images from 250 patients and then used to reconstruct actual real-time, tiny Golden Angle (tGA) radially sampled free breathing data acquired in 10 new patients. Clinical relevance was determined by calculating ventricular volumes from the reconstructed data. Results show that clinical measures of reconstructions from real-time data are not statistically significantly different from gold-standard, cardiac gated, breath-hold techniques.

Post-processing of Direct Reconstructions in Electrical Impedance Tomography

In [2] we present a similar approach but applied to the nonlinear inverse problem of electrical impedance tomography (EIT), modeled as an inverse boundary value problem governed by a generalised Laplace equation

$$\left\{ \begin{array}{rcl} \nabla \cdot \sigma \nabla u & = & 0 \text{ in } \Omega, \\ \sigma \partial_\nu u & = & \varphi \text{ on } \partial\Omega, \end{array} \right. \tag{1}$$

where u models the electrical potential inside the domain $\Omega \subset \mathbb{R}^2$ for a given conductivity σ, with the Neumann boundary condition describing the applied mean-free current φ. The measurement data consists of pairs of current and voltage measurements and is modeled by the current-to-voltage (Neumann-to-Dirichlet) map Λ_σ defined by $\Lambda_\sigma \varphi := u|_{\partial\Omega}$.

A reconstruction of the conductivity σ, can be obtained by a direct inversion algorithm known as the D-bar method [6]. First the measured data is transformed to a non-linear Fourier transform of the conductivity, the so-called scattering transform, which can then be inverted by solving a D-bar equation. Regularisation is applied by low-pass filtering (the stable region) of the scattering transform, resulting in blurred reconstructions. Therefore we investigated training a post-processing network to sharpen these low-pass filtered reconstructions. The trained

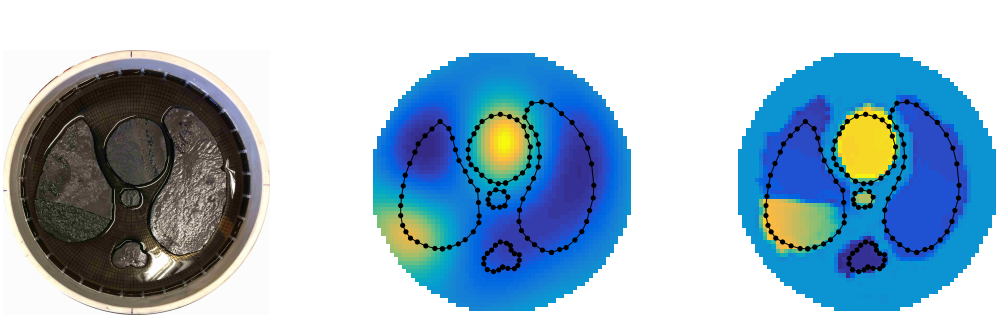


FIGURE 2. Reconstructions of EIT measurements from the ACT4 system at RPI, simulating a pleural effusion (higher conductivity) in the left lung.

network is then applied to measurement data from the ACT4 system located at the Rensselaer Polytechnic Institute.

LEARNED ITERATIVE RECONSTRUCTION FOR PHOTOACOUSTIC LIMITED-VIEW MEASUREMENTS

As seen in the two previous studies direct reconstruction and post-processing performs very well, but is ultimately limited by the information contained in the initial reconstruction. This limitation will have a considerable influence in limited-view geometries and hence motivated by [1] we investigate in [3] a possibility to learn an iterative reconstruction algorithm for realistic 3D high resolution limited-view photoacoustic tomography.

In this application we consider only the linear part that is typically modeled by the following initial value problem for the wave equation

$$(\partial_{tt} - c_0^2\Delta)p(r,t) = 0, \quad p(r,t=0) = x, \quad \partial_t p(r,t=0) = 0.$$

The measurement is then modeled as a linear operator $\mathcal{M}$ acting on the pressure field $p(r,t)$ restricted to the boundary of the computational domain Ω and a finite time window: $y = \mathcal{M}\, p_{|\partial\Omega\times(0,T)}$. The simulation of the forward operator and its adjoint is computationally demanding and hence learning of the iterative reconstruction algorithm needs to follow a greedy approach, i.e. each iterate is learned separately and to the best possible state given the result of the previous iterate.

The network is trained on a set of segmented lung vessels from human CT scans and with some modifications applied to in-vivo measurements of a human hand. Results show that the iterative approach does outperform simple post-processing at the cost of longer computations times, but significantly faster than classical iterative schemes.

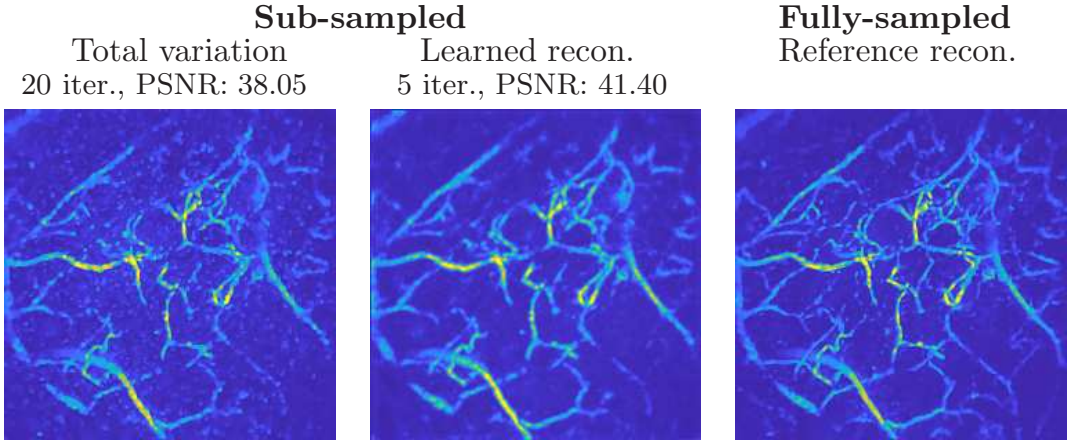


FIGURE 3. Example for real measurement data of a human palm. The images shown are top-down maximum intensity projections.

REFERENCES

[1] J. Adler and O. Öktem. *Solving ill-posed inverse problems using iterative deep neural networks*, Inverse Problems **33** (12), 124007 (2017).
[2] S. J. Hamilton and A. Hauptmann. *Deep D-bar: Real time Electrical Impedance Tomography Imaging with Deep Neural Networks*, *arXiv preprint*, arXiv:1711.03180, (2017)
[3] A. Hauptmann, F. Lucka, M. Betcke, N. Huynh, B. Cox, P. Beard, S. Ourselin and S. Arridge. *Model based learning for accelerated, limited-view 3D photoacoustic tomography*, Accepted for publication in IEEE Transactions on Medical Imaging (2018).
[4] A. Hauptmann, S. Arridge, F. Lucka, V. Muthurangu and J. A. Steeden. *Real-time Cardiovascular MR with Spatio-temporal De-aliasing using Deep Learning - Proof of Concept in Congenital Heart Disease*, *arXiv preprint*, arXiv:1803.05192, (2018)
[5] 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 **26** (9), 4509–4522 (2017).
[6] K. Knudsen, M. Lassas, J. L. Mueller and S. Siltanen. *Regularized D-bar method for the inverse conductivity problem*, Inverse Problems and Imaging **35** (4), 599 (2009).

From Variational Models to Variational Networks

THOMAS POCK
(joint work with K. Kunisch, Y. Chen, K. Hammernik, E. Kobler, F. Knoll)

Variational models are one of the most successful and flexible mathematical frameworks for solving inverse problems in imaging. The basic idea is to represent the solution u^* of the inverse problem as a minimizer of a regularized least-squares problem of the form

$$u^* = \arg\min_u \mathcal{R}(u) + \frac{1}{2}\|Au - b\|^2,$$

where A is the linear forward operator of the inverse problem, b is the measurement data, and $\mathcal{R}$ is a regularization functional that should favor physically meaningful solutions. Over the years, different regularization functionals have been proposed.

Arguably, the most successful one is the total variation (TV) semi-norm [6]

$$\mathrm{TV}(u) = \int_\Omega |Du|.$$

Its most important property is that it allows for sharp discontinuities in the solution. From a computational point of view, the TV is very convenient, since it is convex and hence allows to compute a global minimizer. One caveat of the total variation is that it favors piecewise constant solutions. This leads to the so-called staircasing artifact, which introduces blocky structures in the solution. In order to overcome this problem, different higher-order smoothness variant of the TV have been proposed, such as the total generalized variation (TGV) [1]. The TGV of second order is defined as

$$\mathrm{TGV}^2(u) = \inf_v \int_\Omega |Du - v| + \alpha \int_\Omega |\mathcal{E}v|,$$

where v is an auxiliary vector field and $\mathcal{E}v$ denotes the symmetrized gradient of v. The parameter $\alpha > 0$ can be used to assign a different importance to the higher-order smoothness term.

Although TV and TGV are very successful models, they are still too simple to capture the complexity of natural images, e.g. thin structures, texture. Therefore an interesting questions is whether one could not learn a better regularization functional from images. An interesting model, which still can be seen as a generalization of the TV is given by the so-called fields of experts (FoE) prior model [5]. It is given by

$$\mathrm{FoE}(u) = \sum_{i=1}^n \int_\Omega \phi_i((k_i * u)(x)\mathrm{d}x,$$

where k_i are small filter kernels and ϕ_i are non-linear potential functions. In [3], we used bilevel optimization to learn the optimal kernels and potential functions for the task of image denoising. The idea is to find parameters $\vartheta = (k_i, \phi_i)_{i=1}^n$ that minimize the quadratic difference between the minimizer of the variational model and the corresponding groundtruth solution g. Formally, the bilevel optimization problem is given by

$$\begin{aligned} &\min_\vartheta \frac{1}{2}\|u^*(\vartheta) - g\|^2 \\ \text{s.t.} \quad &u^*(\vartheta) = \arg\min_u \mathrm{FoE}(u, \vartheta) + \frac{1}{2}\|Au - b\|^2. \end{aligned}$$

In order to compute gradients of the upper level function with respect to the parameters ϑ, one needs to rely on the implicit function theorem. This requires in each gradient computation to solve the variational model with high accuracy and also to solve a linear system equation involving the Hessian matrix of the variational model. This is computationally very expensive and very sensitive with respect to errors in the computation of the minimizer of the variational model.

An alternative way to learn the parameters is to replace the exact minimizer in the bilevel formulation by the T-th iterate of a solver for the variational model, e.g.

the T-th iterate of a gradient descent algorithm. With this, the bilevel optimization problem becomes

$$\begin{aligned} &\min_{\vartheta} \frac{1}{2}\|u^T(\vartheta) - g\|^2 \\ \text{s.t.} \quad &u^{t+1} = u^t - h^t \left(\nabla \mathrm{FoE}(u^t, \vartheta) + A^*(Au^t - b)\right), \quad t = 0, ..., T-1, \end{aligned}$$

where h^t is a suitable step size and u^0 is a suitable initial solution. The striking advantage of this formulation is that we can always compute the exact gradient of the upper level function with respect to the parameter vector ϑ, based on the chain rule, which in the neural networks community is known as the back-propagation algorithm. Indeed, representing the iterative algorithm as a graph, we see that it becomes very similar to recently proposed convolutional neural networks (CNNs). Another advantage of this formulation is that inaccuracies of the solution in the lower-level problem, e.g. when performing only a small number of iterations, do not introduce errors to the learning problem. In fact the parameters will be learned such that we get the best performance out of a fixed computational budget prescribed by the number of iterations T.

In [2], we generalized the above scheme, by allowing the parameter vector ϑ to vary in each iteration of the iterative algorithm making the scheme even more similar to CNNs. Hence, we gave this scheme the name *variational networks*. It turns out that the additional flexibility leads to a significant performance increase on a number of inverse problems including image restoration, superresolution, JPEG deblocking and MRI reconstruction [4].

Unfortunately, it turns out that it is hard to make theoretical statements about variational networks. For example it would be helpful to show that variational networks decrease in each iteration some (Bregman) distance to the ground truth solution. We hope that we can give some answers to such problems in our future research.

References

[1] K. Bredies, K. Kunisch and T. Pock. *Total generalized variation*, SIAM Journal on Imaging Sciences, **3**(3), 492–526 (2010).

[2] Y. Chen and T. Pock. *Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **39**(6), 1256–1272 (2016).

[3] Y. Chen, R. Ranftl and T. Pock. *Insights into analysis operator learning: From patch-based sparse models to higher order MRFs*, IEEE Transactions on Image Processing, **23**(3), 1060–1072 (2014).

[4] K. Hammernik, T. Klatzer, E. Kobler, M. Recht, D. Sodickson, T. Pock and F. Knoll. *Learning a variational network for reconstruction of accelerated mri data*, Magnetic resonance in Medicine, (2017).

[5] S. Roth and M. J. Black. *Fields of experts*, Int. J. Comput. Vis., **82**(2), 205–229 (2009).

[6] L. I. Rudin, S. Osher and E. Fatemi. *Nonlinear total variation based noise removal algorithms*, Phys. D Nonlinear Phenom, **60**(1-4), 259–268 (1992).

Deep Learning Mitigating Ill-posedness in Inverse Problems

Maarten Valentijn de Hoop
(joint work with Joan Bruna, Ivan Dokmanić, Stéphane Mallat)

We present two complementary learning-based methods to recovery in ill-posed inverse problems.
The first approach is based on learning generalized moments in a feature space of the models from the data. We distinguish subspaces on which the inverse problem is Lipschitz continuous from a residual that we aim to "Gaussianize" in a feature space. For the feature transform, we introduce a deep scattering network, providing a scale-dependent sparsity specification. If we have a sufficiently large set of relevant models, we can introduce and train a generative network the "inverse" of which can replace the above mentioned transform.
The second method is based on learning a deep network generating a new acquisition representing a "preconditioning" of the data. Here, we introduce a strategy and example network derived from boundary control that expand the well-posedness of the inverse problem.

References

[1] M. T. McCann, K. H. Jin and M. Unser. *Convolutional Neural Networks for Inverse Problems in Imaging: A Review*, IEEE Signal Processing Magazine **34** (6), 85–95 (2017).
[2] J. Sirignano and K. Spiliopoulos. *DGM: A deep learning algorithm for solving partial differential equations*, *arXiv preprint*, arXiv:1708.07469.
[3] A. Bora, A. Jalal, E. Price and A. G. Dimakis. *Compressed Sensing using Generative Models*, *arXiv preprint*, arXiv:1703.03208.

Large Scale Machine Learning and Inverse Problems

Lorenzo Rosasco
(joint work with Ernesto De Vito, Andrea Caponnetto, Umberto De Giovannini, Francesca Odone)

Extracting an estimator from a finite set of input-output samples of an unknown probability distribution, which should be descriptive for new input data – this is the main goal of the theory of learning from examples (see [8, 4]).
Many works established the strong connections of this field with the theory of inverse problems (see [8, 5, 4]). However, these works mainly focus on the discrete setting and do not consider the probabilistic properties of the learning problem. The probabilistic aspect of learning theory and the intrinsically deterministic nature of the theory of inverse problems makes it non-trivial to draw straightforward connections between these two field of studies.
In this work, which is based on the findings in [2], we try to extend this analysis to the continuous case, which will allow us, among other things, to clearify the relation of the consistency approach in learning theory and the stability convergence property in inverse problems.

As learning problem, we consider the input space $X \subset \mathbb{R}^n$ and the output space $Y \subset [-M, M] \subset \mathbb{R}$ for $M \geq 0$. The input $x \in X$ and the output $y \in Y$ are related via the unknown probability distribution $\rho(x, y)$ on $X \times Y$. The main goal is now to extract a function $f_{\mathbf{z}} : X \to \mathbb{R}$ via a given training set $\mathbf{z} := (\mathbf{x}, \mathbf{y}) = ((x_1, y_1), \ldots, (x_\ell, y_\ell))$, which is drawn independently and identically distributed with respect to the probability distribution ρ. The function $f_{\mathbf{z}}$ should be able to describe the distribution ρ, such that $f_{\mathbf{z}}(x)$ is a good estimate of the output for given input data $x \in X$. The map $\mathbf{z} \mapsto f_{\mathbf{z}}$ is commonly referred to as the learning algorithm. The computation of $f_{\mathbf{z}}$ is done by minimizing the expected risk

$$I[f] := \int_{X \times Y} V(f(x), y) \, \mathrm{d}\rho(x, y),$$

where $V(f(x), y)$ is the loss function.
In our work, we restrict ourselves for simplicity reasons to the most common quadratic loss

$$V(f(x), y) = (f(x) - y)^2.$$

Usually, the minimization problem is reformulated as a regularized least squares problem of the form

$$\min_{f \in \mathcal{H}} \left\{ \frac{1}{\ell} \sum_{i=1}^{\ell} (f(x_i) - y_i)^2 + \lambda \Omega(f) \right\}, \tag{1}$$

where $\mathcal{H}$ is a fixed space of functions, called the hypothesis space, Ω a penalty term and $\lambda > 0$ a regularization parameter. In our work, we assume that $\mathcal{H}$ is a reproducing kernel Hilbert space on X with a continuous kernel K to ensure the necessary structure for our analysis (see also [1, 6]). The learning algorithm is called consistent, if the parameter λ can be chosen such that

$$\lim_{\ell \to +\infty} \mathbb{P} \left[I[f_{\mathbf{z}}^{\lambda(\ell, \mathbf{z})}] - \inf_{f \in \mathcal{H}} I[f] \geq \varepsilon \right] = 0 \tag{2}$$

for all $\varepsilon > 0$.
From the side of inverse problems, we consider two Hilbert spaces $\mathcal{H}$ and $\mathcal{K}$, a linear bounded operator $A : \mathcal{H} \to \mathcal{K}$ and the equation

$$Af = g,$$

where $g \in \mathcal{K}$ denotes the unknown exact data without noise. The problem is typically called ill-posed, if the solution does not exist, is not unique or does not depend continuously of the data. What is actually known from the data is just a noisy version of g, namely $g_\delta \in \mathcal{K}$ with $\|g - g_\delta\|_{\mathcal{K}} \leq \delta$. One of the most common ways to tackle this is by considering a variational regularization approach. Applying Tikhonov regularization to the inverse problem, we aim for a solution of the minimization task

$$\min_{f \in \mathcal{H}} \left\{ \|Af - g_\delta\|_{\mathcal{K}}^2 + \lambda \|f\|_{\mathcal{H}}^2 \right\} \tag{3}$$

for $\lambda > 0$, which admits the unique minimizer

$$f_\delta^\lambda = (A^*A + \lambda I)^{-1} A^* g_\delta$$

with A^* the adjoint operator of A. To be a valid regularization technique, the parameter λ has to be chosen depending on the noise level δ and the noisy data g_δ, such that the reconstruction $f_\delta^{\lambda(\delta, g_\delta)}$ converges to the solution $f^\dagger$ given by the Moore-Penrose inverse, if the noise goes to zero, i.e.

$$\lim_{\delta \to 0} \left\| f_\delta^{\lambda(\delta, g_\delta)} - f^\dagger \right\|_{\mathcal{H}} = 0. \tag{4}$$

The similarity between the least squares problem in (1) and the variational regularization approach in (3) as well as between the consistency property (2) of the learning problem and the convergence property (4) in ill-posed inverse problems is striking. However, it is not clear from the learning problem, how to derive a direct problem, i.e. an operator between the Hilbert spaces $\mathcal{H}$ and $\mathcal{K}$ and how the noise level δ in inverse problems is connected to statistical learning.
This work makes an in-depth analysis of these relations and will result, among other things, in a new probabilistic bound for the regularized least-squares algorithm.

References

[1] N. Aronszajn. *Theory of reproducing kernels*, Trans. Amer. Math. Soc., **68**, 337–404 (1950).
[2] E. De Vito, L. Rosasco, A. Caponnetto, U. De Giovannini and F. Odone. *Learning from Examples as an Inverse Problem*, Journal of Machine Learning Research, **6**, 883–904 (2005).
[3] H. W. Engl, M. Hanke and A. Neubauer. *Regularization of inverse problems*, Mathematics and its Applications, **375**, (1996).
[4] T. Evgeniou, M. Pontil and T. Poggio. *Regularization networks and support vector machines*, Adv. Comp. Math., **13**, 1–50 (2000).
[5] T. Poggio and F. Girosi. *A theory of networks for approximation and learning*, Foundation of Neural Networks, 91–106 (1992).
[6] L. Schwartz. *Sous-espaces hilbertiens d'espaces vectoriels topologiques et noyaux associés (noyaux reproduisants)*, J. Analyse Math., **13**, 115–256 (1964).
[7] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill Posed Problems*, W. H. Winston, Washington, D.C., (1977).
[8] V.-N. Vapnik. *Statistical learning theory*, Adaptive and Learning Systems for Signal Processing, Communications and Control. John Wiley & Sons Inc., New York (1998).

Deep Neural Bregman Architectures

Martin Benning
(joint work with Martin Burger)

We show that, under certain constraints, deep neural networks and iterative regularisation methods are equivalent. We further exploit this equivalence to derive a novel Fejér-monotonicity result for deep neural network architectures.

A deep neural network is a composition of simple, parametrised non-linear operators, i.e. mathematically a deep neural network can be described as

$$u^k = \begin{cases} G_{k-1}(\theta_{k-1}, u^{k-1}) & k \neq 1 \\ G_0(\theta_0, f^\delta) & k = 1 \end{cases}, \tag{1}$$

for $k = \{1, \dots, k^*\}$. Here f^δ denotes some input, u^{k^*} describes the output of the k^*-layer network, the individual u^k-s are the hidden layers, and $\{G_k\}_{k=0}^{k^*-1}$ represent the non-linear operators, parametrised with parameters $\{\Theta_k\}_{k=0}^{k^*-1}$. A famous example for (1) is the Rectified Linear Unit (ReLU [6]), which reads as

$$u^k = \begin{cases} \max(0, A_{k-1}u^{k-1} + b_{k-1}) & k \neq 1 \\ \max(0, A_0 f^\delta + b_0) & k = 1 \end{cases}. \tag{2}$$

Hence, the individual non-linear operators are all of the form $G_k(\Theta_k, u^k) = \max(0, A_k u^k + b_k)$, for parameters $\Theta_k := (A_k, b_k)$.

We now demonstrate that the ReLU (2) can be interpreted as an iterative regularisation method of the form of a modification of the linearised Bregman iteration [8, 7]. In its generic form, this modification reads as [1]

$$u^k = \arg\min_u \left\{ \langle K^* Q_{k-1}(Ku^{k-1} - f^\delta), u - u^{k-1} \rangle + D_J^{p^{k-1}}(u, u^{k-1}) \right\}, \tag{3a}$$

$$p^k = p^{k-1} - K^* Q_{k-1}(Ku^{k-1} - f^\delta), \tag{3b}$$

for $k = \{1, \dots, k^*\}$, initial values u^0 and $p^0 \in \partial J(u^0)$. Here K is a linear and bounded operator, $\{Q_k\}_{k=0}^{k^*-1}$ a family of symmetric, positive definite operators, J a proper, convex and lower semi-continuous functional, and $D_J^{p^{k-1}}(u^k, u^{k-1})$ the Bregman distance [2, 5] with respect to J, for arguments u^k and u^{k-1}, and a subgradient $p^{k-1} \in \partial J(u^{k-1})$. If we choose

$$J(u) := \frac{1}{2}\|u\|_2^2 + \chi_{\geq 0}(u),$$

where $\chi_{\geq 0}$ denotes the characteristic function over the non-negative orthant, we can rewrite (3a) to

$$u^k = \max(0, (I - K^* Q_{k-1} K)u^{k-1} + K^* Q_{k-1} f^\delta + q^{k-1}).$$

For $A_{k-1} := I - K^* Q_{k-1} K$ and $b_{k-1} := K^* Q_{k-1} f^\delta + q^{k-1}$ this coincides with (2), if we allow for a different input (in this case u^0). An important consequence of the connection between (2) and (3) is that we can derive a Fejér monotonicity result for (3) if we choose J and $\{Q_k\}_{k=0}^{k^*-1}$ such that the surrogate functional

$$J_k(u) := J(u) - \frac{1}{2}\langle Q_k(Ku - f^\delta), Ku - f^\delta \rangle$$

is convex for all $k \in \{0, \dots, k^* - 1\}$. If we assume that we want to iterate towards a function $u^\dagger$, and that for $u^\dagger$ we can choose the stopping index k^* such that

$$\begin{aligned} \langle Q_{k^*-1}(Ku^{k^*} - f^\delta), Ku^{k^*} - f^\delta \rangle &\leq \langle Q_{k^*-1}(Ku^\dagger - f^\delta), Ku^\dagger - f^\delta \rangle \\ &< \langle Q_{k-1}(Ku^k - f^\delta), Ku^k - f^\delta \rangle \end{aligned} \tag{4}$$

is satisfied for all $k < k^*$, then we can guarantee the following monotonicity result.

Lemma 1 ([1, Lemma 9.7])**.** *Suppose $u^\dagger$ satisfies the discrepancy estimates* (4)*. Then the iterates* (3) *satisfy*

$$D_{J_k}^{q^k}(u^\dagger, u^k) \leq D_{J_{k-1}}^{q^{k-1}}(u^\dagger, u^{k-1})$$

with $q^k \in \partial J(u^k)$, for all $k < k^*$.

As a consequence, given samples $\{f_j^\delta, u_j^\dagger\}_{j=1}^m$ of training (input and output) data, a sensible learning model to obtain optimal parameters $\hat{\Theta} = (\hat{Q}_0, \dots, \hat{Q}_{k^*-1})$ could be the following minimisation problem:

$$\hat{\Theta} \in \arg\min_{\Theta} \left\{ \sum_{j=1}^m D_{J_{k^*}}^{q^{k^*}}(u_j^\dagger, u_j^{k^*}(\Theta)) \quad \text{s.t.} \quad J_k \text{ is convex for all } k < k^* \right\}.$$

Open research questions are the numerical realisation and numerical examples of the proposed model. Also, the monotonic decrease result is merely a first step towards a better mathematical understanding of deep neural networks. Follow-up research can, for example, try to establish whether deep neural networks with suitable constraints are also convergent regularisation methods, or if the iterates converge to a minimiser of some objective energy.

References

[1] M. Benning and M. Burger. *Modern regularization methods for inverse problems*, *arXiv preprint*, arXiv:1801.09922, (2018).

[2] L. M. Bregman. *The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming*, USSR computational mathematics and mathematical physics, **7**(3), 200–217 (1967).

[3] C. Jian-Feng, S. Osher and Z. Shen. *Linearized Bregman iterations for compressed sensing*, Mathematics of Computation, **78**(267), 1515–1536 (2009).

[4] C. Jian-Feng, S. Osher and Z. Shen. *Linearized Bregman iterations for frame-based image deblurring*, SIAM Journal on Imaging Sciences, **2**(1), 226–252 (2009).

[5] Krzysztof C. Kiwiel. *Proximal minimization methods with generalized Bregman functions*, SIAM journal on control and optimization, **35**(4), 1142–1168 (1997).

[6] V. Nair and Geoffrey E. Hinton. *Rectified linear units improve restricted boltzmann machines*, Proceedings of the 27th international conference on machine learning (ICML-10), 807–814 (2010).

[7] W. Yin. *Analysis and generalizations of the linearized Bregman method*, SIAM Journal on Imaging Sciences, **3**(4), 856–877 (2010).

[8] W. Yin, S. Osher, D. Goldfarb and J. Darbon. *Bregman iterative algorithms for ell_1-minimization with applications to compressed sensing*, SIAM Journal on Imaging Sciences, **1**(1), 143–168 (2008).

Learning Regularisers for Imaging Inverse Problems: From Quotient Minimisation to Adversarial Neural Networks

Carola-Bibiane Schönlieb
(joint work with Martin Benning, Guy Gilboa, Joana Grah, Sebastian Lunz, Ozan Öktem)

We consider variational regularisation models of the form

$$\min_{u\in B} \{\mathcal{J}(u) + D(Tu, g)\}, \tag{1}$$

where g is the given datum, u an image computed from g, $T : B \to H$ a linear forward operator between an appropriately chosen solution space B and data space H, D a distance function in H and $\mathcal{J}$ a regularisation functional. We are interested in the idea of learning parametrised regularisers that can distinguish between desired and undesired examples, that is train $\mathcal{J}$ from elements in B that we want $\mathcal{J}$ to favour and from elements in B that we want $\mathcal{J}$ to discourage. In particular, we propose two strategies in this realm: learning sparsity-promoting regularisers by quotient minimisation [2, 3] and learning regularisers via deep generative adversarial networks (GANs) [5], in particular using so-called Wasserstein GANs [1]. Let us briefly review both approaches.

In [2, 3] we propose to learn a K-dimensional parametrisation h of a regularisation functional $\mathcal{J}$ by

$$\hat{h} \in \underset{\substack{\|h\|_2=1 \\ \text{mean}(h)=0}}{\arg\min} \frac{\frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{K} \mathcal{J}(u_i^+; h_k)}{\frac{1}{N}\sum_{j=1}^{N}\sum_{k=1}^{K} \mathcal{J}(u_j^-; h_k)}, \quad \mathcal{J}(u; h) = \|u * h\|_1$$

where we focus on sparsity-promoting $\mathcal{J}$s, h_k corresponds to a convolution kernel, and u_i^+ and u_i^- are desired and undesired input signals, respectively. We show how this approach can be used for learning optimal sparse filters for signals and images of a particular class, e.g. anisotropic filters for images with only vertical structures, to decompose an image into different sparsity patterns, and how learned filters can be deployed as a very simple image classification approach. For the numerical solution of the quotient minimisation model we propose an optimisation scheme that is based on numerical algorithms for non-linear eigenvalue problems [4] and for which convergence along subsequences can be proven.

The second approach for learning disparity-based regularisers from [5] is based on Wasserstein GANs [1]. Here, a regulariser is trained with the discriminative part of a Wasserstein GAN to distinguish between the distribution of 'undesirable' outcomes (modelled by, if it exists, the pseudo-inverse $T^\dagger g$) and 'desirable' outcomes (modelled by the groundtruth for u). More precisely, we train a regulariser Ψ_Θ, parametrised by a neural network Θ to minimise the Wasserstein loss function

$$\mathbb{E}_{X\sim\pi}\Psi_\Theta(X) - \mathbb{E}_{X\sim\rho}\Psi_\Theta(X) + \lambda \cdot \mathbb{E}\left(\|\nabla_x \Psi_\Theta(X)\| - 1\right)_+^2.$$

which is used to approximate

$$\sup_{f \in 1-Lip} \mathbb{E}_{X\sim\rho} f(X) - \mathbb{E}_{X\sim\pi} f(X),$$

where π and ρ are the distributions of desirable images (ground truth images) and of typical corrupted images ρ, respectively. For the resulting regularisation model (1) with $\mathcal{J} = \Psi_\Theta$ we provide, under appropriate assumptions on π and ρ, existence and stability results. The performance of the regulariser is discussed for applications in image denoising and computed tomography.

In conclusion, the discussed approaches demonstrate two different and promising ways for learning image regularisers by showing them both desirable and undesirable solutions. Note that both approaches render regularisers that can be integrated in a variational framework, making them amenable to the derivation of mathematical guarantees for the solution and for statistical interpretation, and they can be trained with both paired and unpaired training data (π, ρ) providing a more flexible supervised learning framework.

References

[1] M. Arjovsky, S. Chintala and L. Bottou. *Wasserstein generative adversarial networks*, In International Conference on Machine Learning, 214–223 (2017).

[2] M. Benning, G. Gilboa and C.-B. Schönlieb. *Learning parametrised regularisation functions via quotient minimisation*, PAMM **16** (1), 933–936 (2016).

[3] M. Benning, G. Gilboa, J. Grah and C.-B. Schönlieb. *Learning Filter Functions in Regularisers by Minimising Quotients*, Scale Space Var. Meth. Comp. Vis. (SSVM), 12 p. (2017).

[4] M. Hein and T. Bühler. *An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse PCA*, Advances in Neural Information Processing Systems (2010).

[5] S. Lunz, O. Öktem and C.-B. Schönlieb. *Learning image regularisers with adversarial neural networks*, preprint (2018).

Reporter: Pascal Fernsel

Participants

Prof. Dr. Simon R. Arridge
Department of Computer Science
University College London
Gower Street
London WC1E 6BT
UNITED KINGDOM

Dr. Martin Benning
Department of Applied Mathematics
and Theoretical Physics (DAMTP)
Centre for Mathematical Sciences
Wilberforce Road
Cambridge CB3 0WA
UNITED KINGDOM

Prof. Dr. Maarten V. de Hoop
Simons Chair in Computational and
Applied Mathematics and Earth Science
Rice University
Houston TX 77005
UNITED STATES

Pascal Fernsel
Zentrum für Technomathematik
Universität Bremen
Bibliothekstrasse 1
28359 Bremen
GERMANY

Prof. Dr. Eldad Haber
Department of Mathematics
University of British Columbia
121-1984 Mathematics Road
Vancouver BC V6T 1Z2
CANADA

Dr. Andreas Hauptmann
The Centre for Medical Image
Computing
The Front Engineering Building, Floor 3
Malet Place
London WC1E 7JE
UNITED KINGDOM

Prof. Dr. Gitta Kutyniok
Institut für Mathematik
Sekr. MA 5-4
Technische Universität Berlin
Straße des 17. Juni 136
10623 Berlin
GERMANY

Prof. Dr. Peter Maaß
Fachbereich 3 -
Mathematik und Informatik
Universität Bremen
28344 Bremen
GERMANY

Dr. Ozan Öktem
Department of Mathematics
KTH
10044 Stockholm
SWEDEN

Prof. Dr. Thomas G. Pock
Institute for Computer Graphics and
Vision
University of Technology Graz
Inffeldgasse 16b/II
8010 Graz
AUSTRIA

Prof. Dr. Lorenzo Rosasco
Massachusetts Institute of Technology
Office: 46-5155 C
43, Vassar Street
Cambridge, MA 02139
UNITED STATES

Prof. Dr. Lars Ruthotto
Department of Mathematics and
Computer Science
Emory University
400, Dowman Drive
Atlanta, GA 30322
UNITED STATES

Prof. Dr. Otmar Scherzer
Computational Science Center
Universität Wien
Oskar-Morgenstern-Platz 1
1090 Wien
AUSTRIA

Prof. Dr. Carola-Bibiane Schoenlieb
Department of Applied Mathematics and Theoretical Physics (DAMTP)
Centre for Mathematical Sciences
Wilberforce Road
Cambridge CB3 0WA
UNITED KINGDOM

Prof. Dr. Michael Unser
EPFL Lausanne
BM 4.134 (Bâtiment B)
Station 17
1015 Lausanne
SWITZERLAND